

ANÁLISIS DE MALWARE

CryptoLocker

2025



Rubens Ballester Lillo

BBK Bootcamp - Ciberseguridad

ÍNDICE

1. Resumen.	1
2. Introducción.	2
2.1. Objeto y objetivos del estudio.	2
2.2. Metodología.	3
2.3. Herramientas utilizadas.	4
3. Información de la muestra utilizada.	6
4. Análisis Estático.	8
4.1. PE Studio.	8
4.1.1. Indicadores.	8
4.1.2. Secciones.	10
4.1.3. Librerías.	11
4.1.4. Importaciones.	12
4.2. PortexAnalyzer.	12
4.3. Strings.	14
5. Análisis Dinámico.	16
5.1. Any.run.	16
5.1.1. Táctica: Persistencia.	16
5.1.2. Táctica: Evasión de Defensas.	16
5.1.3. Táctica: Descubrimiento.	17
5.1.4. Táctica: Impacto.	17
5.2. Wireshark.	21
5.1.1. Resumen del Tráfico de Red	21
5.1.2. Características Sospechosas Identificadas	22
5.1.3. Análisis de IPs Involucradas	22
5.1.4. Comunicación C&C.	22
6. Análisis de Código.	24
6.1. Ghidra.	24
6.2. ILSpy.	25
7. Conclusiones.	27
7.1. Resumen de los hallazgos más importantes de los análisis.	27
7.2. Evaluación de la peligrosidad y complejidad de CryptoLocker.	28
8. Bibliografía y referencias.	29

Análisis del Malware CryptoLocker.

1. Resumen.

CryptoLocker surgió a finales de 2013, identificándose rápidamente como una de las amenazas de ransomware más notorias y extendidas de su época. Este malware se especializaba en cifrar los archivos de los usuarios en sistemas Windows, volviéndolos inaccesibles, para luego exigir un pago, comúnmente en Bitcoin u otras criptomonedas, a cambio de la clave de descifrado. Su aparición marcó un punto de inflexión en la evolución del ransomware debido a su efectividad y al modelo de negocio criminal que perfeccionó.

Habitualmente, CryptoLocker se propagaba a través de correos electrónicos de phishing que contenían archivos adjuntos maliciosos o mediante la explotación de vulnerabilidades en software desactualizado. También se le asoció con la botnet Gameover Zeus, que facilitó su distribución a gran escala. Una vez en el sistema, buscaba una amplia gama de tipos de archivo para cifrarlos utilizando criptografía asimétrica (RSA-2048), lo que hacía prácticamente imposible la recuperación de los archivos sin la clave privada en posesión de los atacantes.

El impacto de CryptoLocker fue considerable a nivel mundial, afectando tanto a usuarios particulares como a empresas. Muchas víctimas perdieron datos valiosos de forma permanente, y las empresas se enfrentaron a interrupciones operativas significativas y costes económicos derivados del pago de rescates o de los esfuerzos de recuperación. Se estima que los creadores de CryptoLocker lograron extorsionar millones de dólares durante su periodo de mayor actividad.

En junio de 2014, una operación internacional coordinada por agencias de la ley, conocida como "Operation Tovar", consiguió dismantelar la infraestructura de la botnet Gameover Zeus, lo que interrumpió en gran parte las operaciones de CryptoLocker. Aunque la amenaza original fue neutralizada en gran medida, su "éxito" inspiró a numerosos imitadores y dio lugar a una nueva generación de ransomware que adoptó tácticas similares, convirtiéndose en un problema persistente en el panorama de la ciberseguridad. A pesar de su dismantelamiento, el legado de CryptoLocker reside en cómo demostró la viabilidad del modelo de ransomware como servicio y el profundo impacto que este tipo de malware puede tener.

2. Introducción.

2.1. Objeto y objetivos del estudio.

El objeto de este estudio es diseccionar y comprender en profundidad la funcionalidad, el origen y el impacto potencial del ransomware CryptoLocker. CryptoLocker fue una de las primeras familias de ransomware que cifraba archivos y exigía un pago para su recuperación, sentando un precedente para muchas amenazas posteriores.

Para lograr una mejor comprensión de CryptoLocker, este estudio se abordará desde una perspectiva tanto estática como dinámica. El análisis se realizará en un entorno controlado, utilizando la máquina virtual “Análisis de Malware” y herramientas especializadas para la inspección y monitorización del malware.

Los objetivos específicos de este estudio se dividen en tres áreas principales:

- **Funcionalidad:**
 - Analizar las técnicas de propagación de CryptoLocker, incluyendo sus principales vectores de infección.
 - Examinar en detalle sus capacidades de malware, especialmente el proceso de cifrado de archivos (tipos de archivos afectados, algoritmos criptográficos utilizados), la gestión de claves, la comunicación con los servidores de Comando y Control (C&C) para obtener claves o enviar información, las técnicas de persistencia en los sistemas infectados y la presentación del mensaje de rescate a la víctima.
- **Origen:**
 - Investigar la historia de CryptoLocker, desde sus primeras detecciones y versiones hasta su evolución y las posibles variantes que surgieron.
- **Impacto Potencial:**
 - Evaluar el impacto económico y la pérdida de datos causada por las infecciones de CryptoLocker en individuos y organizaciones.
 - Analizar casos de estudio relevantes donde CryptoLocker haya causado daños significativos, para comprender mejor sus consecuencias directas e indirectas.

El objetivo final de este estudio es obtener la mayor cantidad de información posible mediante el uso de herramientas de análisis de malware.

2.2. Metodología.

El análisis de CryptoLocker se llevará a cabo mediante una combinación de técnicas de análisis estático y dinámico, con el fin de obtener una comprensión completa de su funcionamiento interno, sus mecanismos de propagación, sus capacidades de cifrado y sus métodos de persistencia. La investigación se realizará en un entorno controlado y aislado para garantizar la seguridad y evitar cualquier riesgo de infección real.

1. Entorno de Análisis: Para este estudio, se configurará un entorno de laboratorio virtual. Este entorno consistirá en la máquina Análisis de Malware con un sistema operativo Windows y con herramientas específicas para la observación y manipulación del malware. La MV estará completamente aislada de la red principal para prevenir cualquier propagación accidental o interacción no deseada.

2. Análisis Estático: El análisis estático se centrará en la inspección del código binario del malware sin ejecutarlo. Este enfoque permitirá identificar características clave, como:

- **Identificación del archivo:** Recopilación de información básica como el hash del archivo (MD5, SHA1, SHA256) para identificar la muestra y buscar en bases de datos de inteligencia de amenazas.
- **Encabezados y secciones:** Examen de los encabezados PE (Portable Executable) para obtener datos sobre el compilador, las librerías importadas y las secciones del ejecutable.
- **Cadenas de texto:** Extracción de cadenas de texto significativas que puedan revelar nombres de archivos, URLs, direcciones IP, mensajes de error, claves de registro o comandos internos.
- **Ofuscación y empaquetamiento:** Detección de posibles técnicas de ofuscación o empaquetamiento que el malware pueda utilizar para dificultar su análisis.

3. Análisis Dinámico: El análisis dinámico implica la ejecución controlada del malware en el entorno virtualizado para observar su comportamiento en tiempo real. Este proceso permitirá documentar:

- **Cambios en el sistema de archivos:** Monitorización de la creación, modificación o eliminación de archivos, especialmente en directorios críticos o en las unidades de red mapeadas. Se prestará especial atención a la creación de nuevos archivos cifrados y a los archivos de rescate.
- **Cambios en el registro:** Observación de las modificaciones en el registro de Windows, que podrían indicar mecanismos de persistencia, configuraciones del malware o datos de cifrado.

- **Actividad de red:** Captura y análisis del tráfico de red para identificar conexiones a servidores de Comando y Control, intentos de descarga de archivos o comunicación para la gestión de claves de cifrado.
- **Actividad de procesos:** Monitorización de la creación de nuevos procesos, la inyección de código en otros procesos o la elevación de privilegios.
- **Mecanismos de cifrado:** Observación del proceso de cifrado, incluyendo la identificación de los tipos de archivos afectados.

2.3. Herramientas utilizadas.

Para llevar a cabo el análisis de CryptoLocker, he utilizado diversas herramientas especializadas, seleccionadas por su capacidad para facilitar tanto el análisis estático como el dinámico del malware. Estas herramientas, en su conjunto, permitirán obtener una visión completa del funcionamiento de CryptoLocker, desde su estructura interna hasta su comportamiento en ejecución.

Para el Análisis Estático:

- **PortexAnalyzer:** Esta herramienta se utilizará para realizar una inspección inicial y profunda de los archivos ejecutables asociados a CryptoLocker. PortExAnalyzer permite examinar detalladamente la estructura del archivo, incluyendo sus cabeceras (como la cabecera DOS, PE y Optional Header), las diferentes secciones (como .text, .data, .rsrc) y sus permisos, así como las tablas de importación y exportación de funciones. PortExAnalyzer será utilizado específicamente para generar una representación visual del binario en formato PNG, lo que puede ayudar a identificar patrones o estructuras inusuales en el código.
- **PE-Studio:** PE-Studio es una herramienta de análisis estático integral que ofrece una visión muy detallada de los ejecutables PE. Permite una inspección exhaustiva de todos los componentes de un archivo PE, incluyendo la identificación de cabeceras, la visualización de las secciones y sus propiedades, el examen de los directorios de datos (importaciones, exportaciones, recursos, etc.), la extracción de cadenas de texto (strings) y la revisión de librerías DLL importadas y funciones exportadas. Además, PE-Studio incorpora un sistema de "indicadores de compromiso" (IOCs) y un analizador de virustotal que pueden alertar sobre características sospechosas o conocidas del malware, ayudando a identificar rápidamente la naturaleza maliciosa de la muestra y posibles tácticas de ofuscación o anti-análisis.

- **DIE (Detect It Easy):** Esta utilidad es fundamental para la fase inicial del análisis estático, ya que está diseñada para determinar rápidamente el tipo de archivo, el compilador utilizado para su creación y, lo que es más importante, la presencia y el tipo de empaquetadores o protectores (como UPX, Themida, ASProtect, etc.). DIE será crucial para conocer si el binario contiene partes empaquetadas o comprimidas, lo cual dificultaría un análisis directo, y también para calcular el nivel de entropía del archivo, un indicador que puede sugerir la presencia de código cifrado u ofuscado, característico de muchos malwares. Conocer si un archivo está empaquetado es crucial, ya que esto afecta la capacidad de otras herramientas estáticas para analizar su contenido directamente.
- **Strings:** La herramienta strings es una utilidad de línea de comandos que se utiliza para buscar e imprimir secuencias de cadenas de texto dentro de archivos binarios. En el análisis de malware, es utilizada para extraer rápidamente información incrustada en el ejecutable que no es visible a simple vista. Estas cadenas pueden revelar datos críticos como URLs de servidores de Comando y Control, direcciones IP, nombres de archivos que el malware intenta crear o modificar, claves de registro, mensajes de error, comandos internos que el malware puede ejecutar, o incluso partes del mensaje de rescate de CryptoLocker. Su simplicidad y eficiencia la hacen indispensable para obtener una visión inicial del propósito del malware.

Para el Análisis Dinámico:

- **Any.Run:** Any.Run es una plataforma de sandboxing interactiva basada en la nube que permite ejecutar muestras de malware en un entorno seguro y aislado. Proporciona una visualización detallada de la actividad del proceso, el tráfico de red generado, conexiones a C&C, descargas, los cambios en el sistema de archivos, y las modificaciones en el registro de Windows. Es una gran herramienta para ver cómo CryptoLocker cifra los archivos y presenta la nota de rescate.
- **Wireshark:** Wireshark es un analizador de protocolos de red. Durante el análisis dinámico, se utilizará para capturar y examinar en detalle todo el tráfico de red generado por CryptoLocker mientras se ejecuta en la máquina virtual. Esto permitirá identificar las conexiones salientes, los protocolos utilizados (HTTP, HTTPS, DNS, etc.), los nombres de dominio o direcciones IP a los que se conecta, y el contenido de los paquetes. Es fundamental para entender cómo el malware se comunica con su infraestructura de control, qué información envía o recibe y si utiliza algún tipo de cifrado en sus comunicaciones.

Para el Análisis de Código:

- **Ghidra:** Ghidra es una potente herramienta open-source de ingeniería inversa desarrollada por la NSA. Permite desensamblar y descompilar código binario a pseudocódigo legible (similar a C), facilitando el análisis de la lógica interna de los programas. Aunque es muy buena para binarios nativos, en este informe se ha encontrado que su eficacia se ve limitada por la ofuscación del malware, impidiendo el análisis directo de su código real.
- **ILSpy:** ILSpy es un descompilador open-source específicamente diseñado para ensamblados de .NET. Es fundamental para analizar programas compilados en este framework, ya que traduce el código intermedio a lenguajes de alto nivel como C#. En este caso particular, ha sido crucial para confirmar que el malware es un binario .NET ofuscado con Confuser, permitiendo acceder a los metadatos y la estructura básica del código fuente.

3. Información de la muestra utilizada.

Esta sección detalla la información fundamental de la muestra específica de CryptoLocker utilizada para este análisis.

Nombre del archivo de la muestra: El nombre del archivo utilizado para este análisis, al descargarlo del github de theZoo es 1002.exe, pero es el Cryptolocker.

Hash de la muestra (SHA256): El hash SHA256 de la muestra es e908dca957b9cb7759feeabef0f2921e3cb236368acc5e124e87af0492308b14.

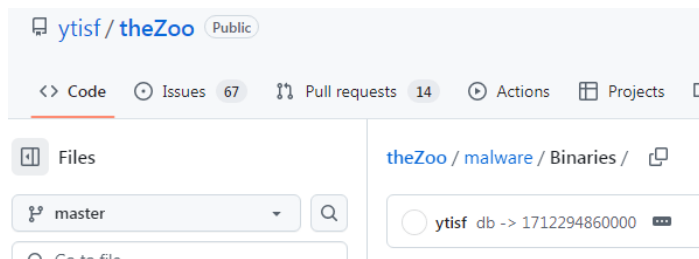
Hash de la muestra (MD5): El hash MD5 es 3c877dfd0d60572be7c939c08c39866d.

Tamaño del archivo: El archivo ejecutable tiene un tamaño de 251 Kb.

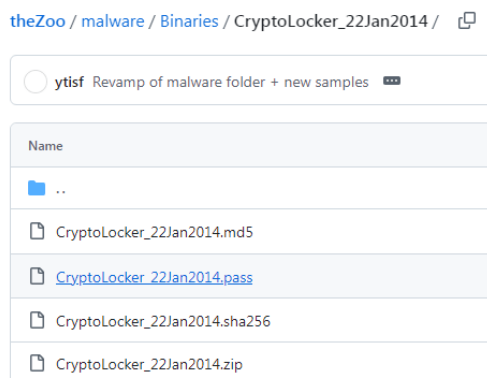
Fecha de compilación: El archivo fue compilado o al menos fue subido a github el 21 de Junio de 2021.

Tipo de archivo: El archivo es un ejecutable con la extensión .exe.

Contexto de la obtención de la muestra: Para obtener la muestra entramos en el siguiente github <https://github.com/ytisf/theZoo>. Una vez en él vamos a la carpeta malware y después a la carpeta Binarios.



Una vez aquí buscamos nuestro malware que es Cryptolocker y elegimos el archivo .zip para descargar.



Dentro de este .zip que descargamos encontramos el ejecutable .exe.

Versión de CryptoLocker: Según podemos ver en el github del proyecto The Zoo, la versión del malware CryptoLocker que estamos usando es la del 22 de Enero de 2014.

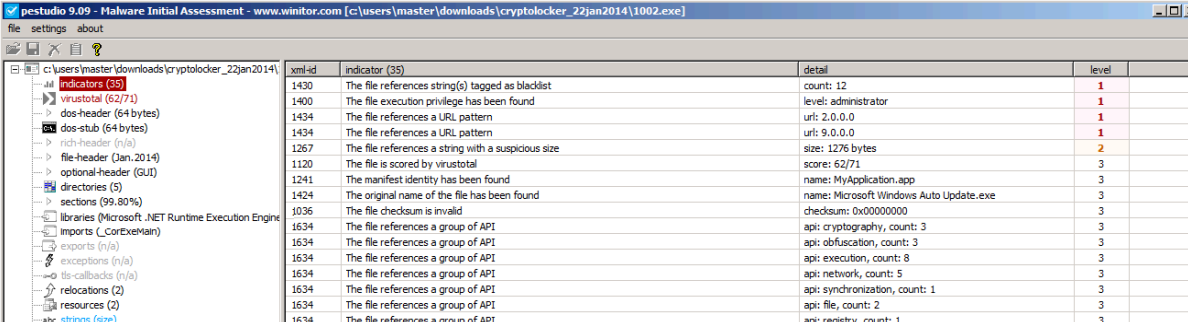
4. Análisis Estático.

4.1. PE Studio.

Para realizar el análisis estático voy a utilizar la herramienta PE Estudio, como hemos dicho anteriormente esta es una herramienta de análisis estático integral por lo que vamos a comenzar por ella para obtener las primeras pistas.

4.1.1. Indicadores.

Entramos en el programa y le añadimos uno nuevo archivo desde la opción file, comenzará a analizar pormenorizadamente nuestro malware y podemos consultar las distintas secciones, la primera sección en la que nos vamos a fijar es indicators, en ella podemos ver, clasificados por su severidad, distintos indicadores de peligro en el archivo que hemos pasado.



xml-id	indicator (35)	detail	level
1430	The file references string(s) tagged as blacklist	count: 12	1
1400	The file execution privilege has been found	level: administrator	1
1434	The file references a URL pattern	url: 2.0.0.0	1
1434	The file references a URL pattern	url: 9.0.0.0	1
1267	The file references a string with a suspicious size	size: 1276 bytes	2
1120	The file is scored by virustotal	score: 62/71	3
1241	The manifest identity has been found	name: MyApplication.app	3
1424	The original name of the file has been found	name: Microsoft Windows Auto Update.exe	3
1036	The file checksum is invalid	checksum: 0x00000000	3
1634	The file references a group of API	api: cryptography, count: 3	3
1634	The file references a group of API	api: obfuscation, count: 3	3
1634	The file references a group of API	api: execution, count: 8	3
1634	The file references a group of API	api: network, count: 5	3
1634	The file references a group of API	api: synchronization, count: 1	3
1634	The file references a group of API	api: file, count: 2	3
1634	The file references a group of API	api: registry, count: 1	3

Vamos a analizar aquellos indicadores que tienen un nivel de 1 y 2, estos son los indicadores que más riesgo tienen.

Los indicadores de nivel 1 o riesgo muy alto:

The file references string(s) tagged as blacklist, count: 12:

Descripción: Este indicador significa que el ejecutable contiene doce cadenas de texto que PE-Studio ha identificado previamente como asociadas a malware conocido o a comportamientos maliciosos. PE-Studio mantiene una lista negra interna de strings.

Implicación para el análisis de malware: La presencia de estas cadenas en la muestra es un fuerte indicio de actividad maliciosa.

The file execution privilege has been found, level: administrator:

Descripción: Indica que el manifiesto del ejecutable solicita o requiere privilegios de nivel de administrador para poder ejecutarse correctamente en el sistema. Los programas legítimos a menudo necesitan estos privilegios para instalar software o realizar cambios a nivel de sistema, pero también es una característica común en el malware para asegurar un control total sobre el sistema operativo.

Implicación para el análisis de malware: Para un ransomware como CryptoLocker, la necesidad de privilegios de administrador es crítica, ya que le permite realizar acciones como cifrar archivos protegidos, modificar claves de registro importantes, crear tareas programadas para persistencia, o deshabilitar software de seguridad.

The file references a URL pattern, url: 2.0.0.0:

Descripción: Este indicador muestra que el archivo contiene una cadena que coincide con un patrón de URL específico, en este caso, "2.0.0.0". Es importante tener en cuenta que "2.0.0.0" no es una dirección IP válida de Internet público, sino un rango reservado para uso interno o especial. Su referencia podría ser un marcador de posición, una IP de red interna, o parte de una configuración de red poco corriente.

Implicación para el análisis de malware: Es un hallazgo sospechoso. El malware podría estar intentando contactar con una dirección IP interna, o esta cadena podría formar parte de un algoritmo para generar direcciones de C&C, o incluso ser un indicador de un servidor de prueba o de backup del atacante.

The file references a URL pattern, url: 9.0.0.0:

Descripción: Similar al caso anterior, el ejecutable hace referencia a una cadena que coincide con el patrón de URL "9.0.0.0". Al igual que "2.0.0.0", "9.0.0.0" no es una dirección IP pública habitual y podría indicar un uso interno, un placeholder, o un intento de contactar con una red específica.

Implicación para el análisis de malware: Esto refuerza la idea de que el malware podría estar buscando establecer comunicación de red, posiblemente con direcciones IP de C&C codificadas o generadas dinámicamente.

En cuanto indicadores de nivel 2 solo nos aparece el siguiente:

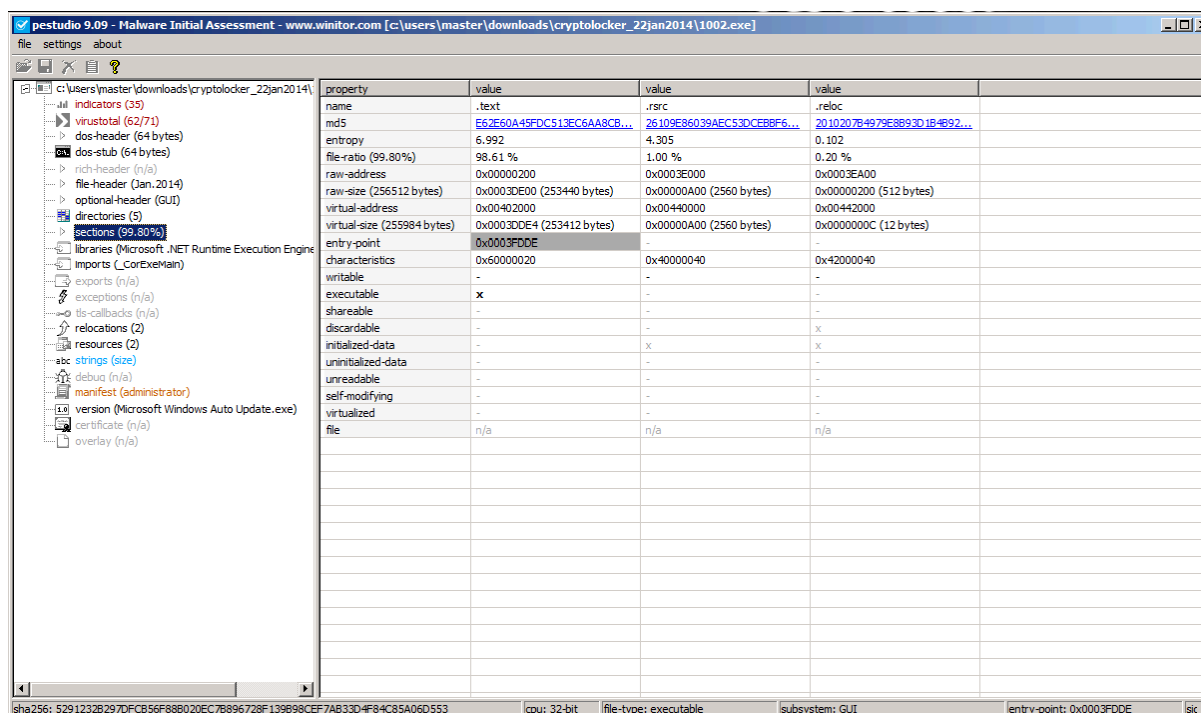
The file references a string with a suspicious size, size: 1276 bytes:

Descripción: Este indicador señala que el ejecutable contiene una cadena de texto extrañamente larga, con un tamaño de 1276 bytes. Las cadenas de texto largas pueden ser una señal de que el malware está incrustando datos significativos, como URLs codificadas, claves de cifrado, partes de un payload, scripts, o incluso la nota de rescate completa.

Implicación para el análisis de malware: Una cadena de este tamaño es sospechosa y no es común en software legítimo sin un propósito claro.

4.1.2. Secciones.

Después de analizar los indicadores vamos a ver que ha encontrado el apartado de sections.



El análisis de estas secciones puede revelar mucho sobre la funcionalidad del malware y la presencia de técnicas de ofuscación:

.text:

Esta es la sección principal de código ejecutable. La entropía de 6.99 es extremadamente alta, lo que indica que el código en esta sección está altamente comprimido o cifrado. La tamaño RAW es muy similar a la tamaño virtual. Los permisos Read, Execute son normales para una sección de código.

.rloc:

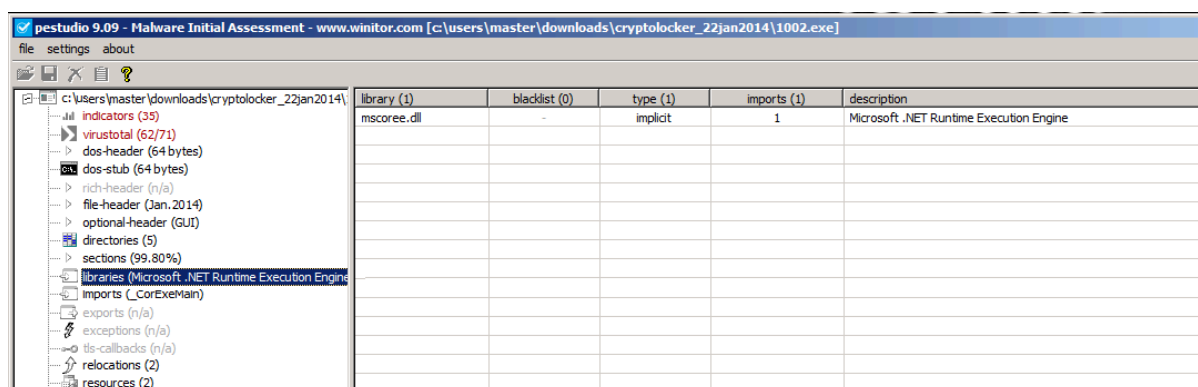
La sección .reloc contiene información que permite al cargador del sistema operativo ajustar las direcciones de memoria en el código si el ejecutable no se carga en su dirección base preferida. Una entropía de 0.10 es muy baja, lo cual es un valor esperado y normal.

.rsrc:

Contiene recursos del programa, como iconos, imágenes, cursores, o cadenas de texto. Una entropía de 4.30 se considera de nivel medio. Esto sugiere que los recursos pueden no estar fuertemente cifrados o comprimidos como el código principal, pero tampoco son simplemente datos de texto o imágenes básicas.

4.1.3. Librerías.

La sección de "Librerías Importadas" en PE-Studio muestra las Dynamic Link Libraries (DLLs) externas de las que el ejecutable depende para funcionar.

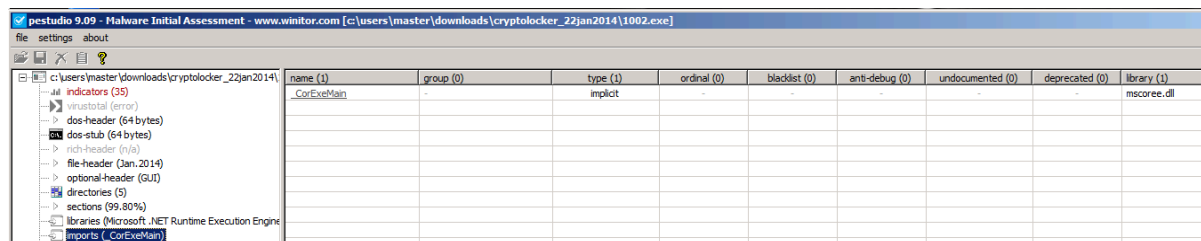


Como vemos en la imagen solo aparece la librería mscoree.dll, esta es una biblioteca de ejecución del motor de Microsoft .NET Framework. Es una librería fundamental para que las aplicaciones escritas en lenguajes .NET puedan ejecutarse en un sistema Windows.

La presencia de mscoree.dll es un indicador directo de que la muestra de CryptoLocker ha sido desarrollada o compilada utilizando Microsoft .NET Framework. Esto que hemos encontrado tiene sentido con la alta entropía observada previamente en la sección de código .text, ya que los ensamblados .NET ofuscados o empaquetados suelen presentar una entropía elevada.

4.1.4. Importaciones.

Otro de los apartados que podemos consultar dentro del análisis de PE Studio es de las importaciones.



En la imagen vemos la importación de `_CorExeMain`, este es el punto de entrada principal para las aplicaciones escritas en .NET Framework. Cuando un sistema operativo ejecuta un programa .NET, el cargador de Windows redirige la ejecución a `_CorExeMain` dentro de `mscorlib.dll`.

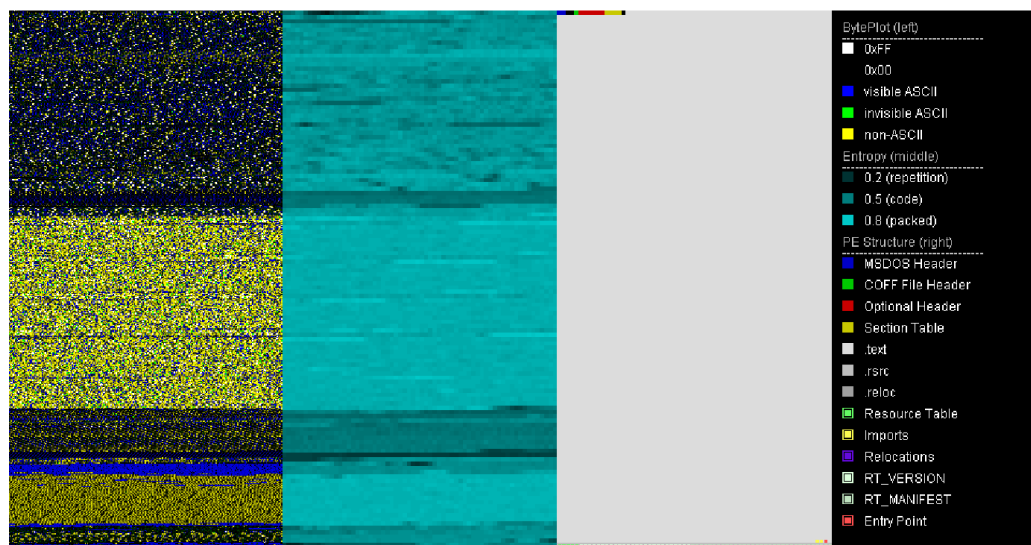
4.2. PortexAnalyzer.

PortexAnalyzer es una herramienta para el análisis estático de malware que nos permite crear una imagen en PNG del binario para encontrar patrones o estructuras inusuales en el ejecutable.

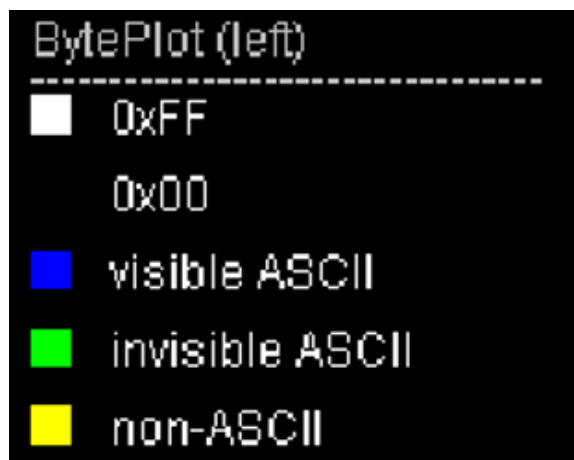
Para utilizarlo, tenemos que usar la terminal de comandos de windows y usar el siguiente comando.

```
C:\Users\master\Desktop\Análisis Estático>PortexAnalyzer.jar -p cryptolocker.png
C:\Users\master\Downloads\CryptoLocker_22Jan2014\1002.exe
```

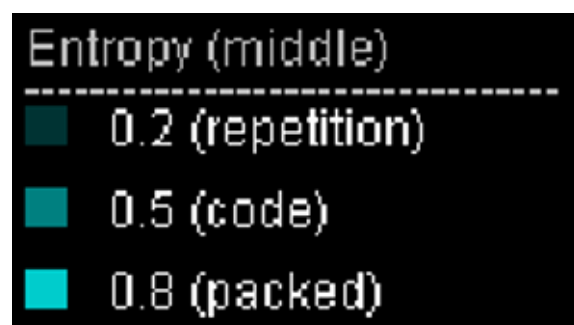
Este comando nos genera la siguiente imagen PNG.



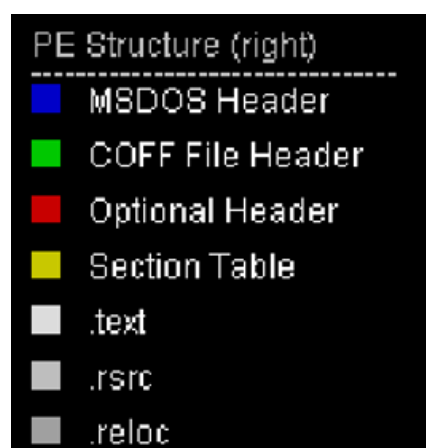
Como podemos ver en la primera sección, existen un gran número de caracteres non-ASCII lo que puede ser altamente sospechoso.



La entropía que nos encontramos es de 0.8 en la parte de Packed, esto es lo que más nos llama la atención. Tiene una entropía alta lo que indica la cantidad de desorden o aleatoriedad en los datos de un archivo.



En la estructura, si nos fijamos en las que empiezan por punto son las secciones, de momento, son las más importantes. Vemos que el binario tiene 3 secciones: .text, .rsrc y .reloc. Como vemos en la imagen general, la que más tamaño ocupa es la sección .text, como hemos visto anteriormente.



4.3. Strings.

Las cadenas de texto son secuencias legibles incrustadas en un binario. Son cruciales para revelar URLs de C&C, rutas de archivos o mensajes de rescate. Proporcionan una visión rápida del propósito y las capacidades del malware antes de un análisis más profundo.

Para sacar las strings de nuestro malware he utilizado el siguiente comando en el CMD.

```
C:\Users\master\Desktop\Análisis Estático>strings.exe -n 10 C:\Users\master\Downloads\CryptoLocker_22Jan2014\1002.exe > C:\Users\master\Desktop\strings.txt
```

Si analizamos los strings obtenidos encontramos las siguientes líneas sospechosas:

1. Ofuscación con Confuser v1.9.0.0:

```
Confuser v1.9.0.0
```

El análisis revela la presencia de Confuser v1.9.0.0, confirmando el uso de esta herramienta profesional de ofuscación .NET. Esta presencia explica la alta entropía detectada anteriormente e indica el uso de técnicas avanzadas.

2. Técnicas Anti-Análisis y Evasión de Sandboxes.

```
Loop broken  
Debugger detected (Managed)
```

```
IsDebuggerPresent  
OutputDebugString  
GetEnvironmentVariable
```

La muestra implementa detección de entornos de análisis mediante strings como Debugger detected (Managed), IsDebuggerPresent y Loop broken. Estas capacidades detectan profilers .NET, debuggers nativos y herramientas managed, implementando respuestas activas que dificultan significativamente el análisis dinámico en entornos controlados.

3. Cadenas de Criptografía Características de Ransomware

```
AesManaged
StreamWriter
AesCryptoServiceProvider
RSACryptoServiceProvider
BinaryReader
SHA1CryptoServiceProvider
CreateEncryptor
CreateDecryptor
```

Los strings confirman funciones de criptográficas como, RSACryptoServiceProvider, CreateEncryptor y CryptoStream. Esta combinación implementa el esquema de encriptación típico de CryptoLocker, donde AES cifra archivos y RSA protege las claves.

4. Suplantación de Identidad.

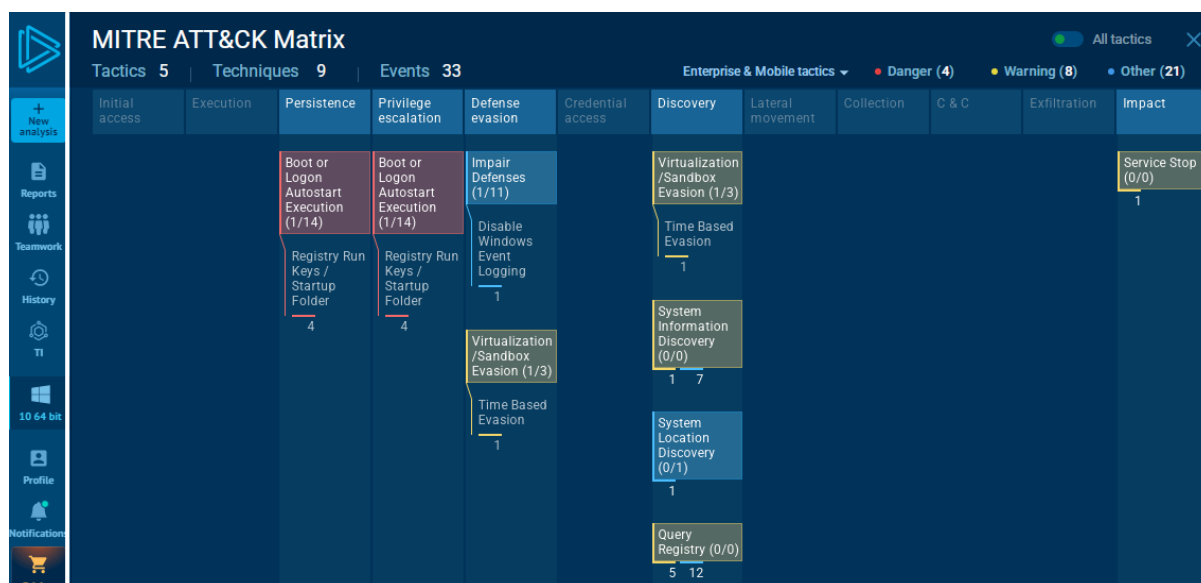
```
Microsoft Windows Auto Update.exe
Microsoft Windows Auto Update
```

El malware se camufla completamente como Microsoft Windows Auto Update.exe, falsificando todos los metadatos del archivo. Esta suplantación constituye ingeniería social técnica efectiva, ya que los usuarios no cuestionan procesos de actualización de Windows, reduciendo significativamente la probabilidad de detección.

5. Análisis Dinámico.

5.1. Any.run.

El análisis dinámico en Any.run ha revelado varias tácticas y técnicas del marco MITRE ATT&CK, lo que sugiere el comportamiento de un posible software malicioso.



A continuación, se detalla cada una de las detecciones:

5.1.1. Táctica: Persistencia.

- **Boot or Logon Autostart Execution / Registry Run Keys / Startup Folder.**

Esta es una táctica crítica para el malware. Significa que el programa ha modificado o intentado modificar puntos de inicio automático en el sistema. El objetivo es asegurar que el software malicioso se ejecute automáticamente cada vez que el sistema se inicia o un usuario inicia sesión, garantizando así su persistencia en la máquina comprometida.

5.1.2. Táctica: Evasión de Defensas.

- **Impair Defenses**

Indica que el software ha intentado deshabilitar o interferir con las capacidades de defensa del sistema. Esto puede incluir acciones como deshabilitar el cortafuegos, el software antivirus o el registro de eventos de Windows (como se sugiere con "Disable Windows Event Logging"), para evitar ser detectado o analizado.

- **Virtualization/Sandbox Evasion**

Esta técnica sugiere que el software ha detectado que se está ejecutando en un entorno virtualizado e intentó evadir el análisis dinámico o modificar su comportamiento para no revelar sus verdaderas intenciones. Es común en malware sofisticado para evitar ser descubierto.

- **Time Based Evasion Detecciones: 1.**

Se refiere a la capacidad del software de esperar un cierto tiempo o una condición específica antes de ejecutar su carga útil o comportamiento malicioso. Esto se hace para eludir la detección en entornos de análisis rápidos, que podrían no permitir que el programa alcance su fase activa.

5.1.3. Táctica: Descubrimiento.

- **System Information Discovery**

El software ha intentado recopilar información sobre el sistema en el que se ejecuta. Esto puede incluir detalles como el nombre del equipo, la versión del sistema operativo, la arquitectura de la CPU, las direcciones IP, etc. Esta información es útil para el atacante para adaptar su ataque o para el reconocimiento de la víctima.

- **System Location Discovery**

El software ha intentado determinar la ubicación geográfica del sistema. Esto podría hacerse a través de la dirección IP o la configuración regional del sistema. Podría ser relevante para ataques dirigidos geográficamente o para evitar infectar máquinas en ciertas regiones.

- **Query Registry**

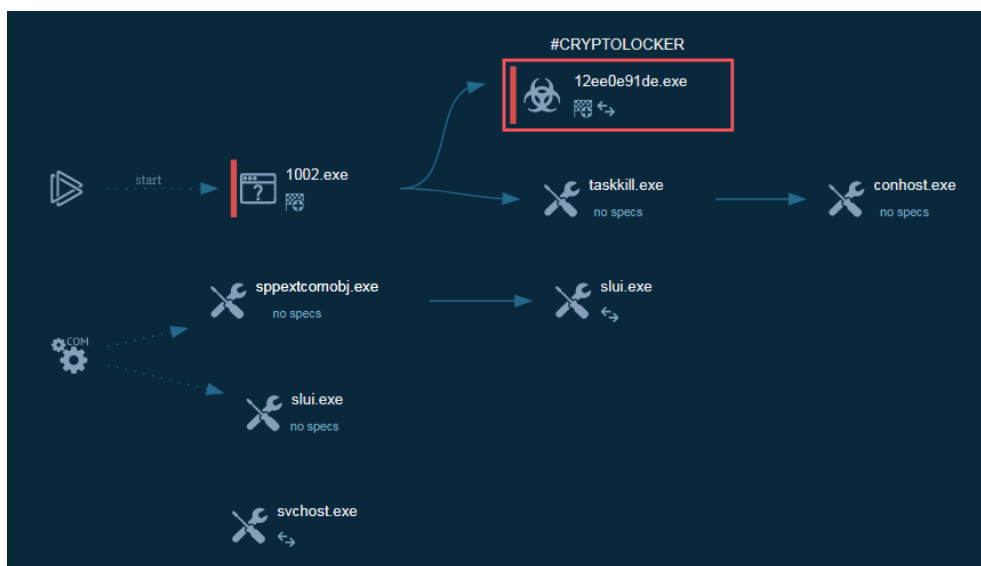
El software ha intentado consultar o leer claves del registro de Windows. El registro contiene una gran cantidad de información sobre la configuración del sistema, software instalado, usuarios, etc., lo que el malware puede usar para recopilar datos o para determinar el entorno.

5.1.4. Táctica: Impacto.

- **Service Stop**

Esta es una técnica de alto impacto. Indica que el software ha intentado detener o deshabilitar servicios críticos del sistema. Esto puede causar denegación de servicio, interrumpir el funcionamiento normal del sistema o facilitar otras acciones maliciosas al deshabilitar servicios de seguridad o de monitoreo.

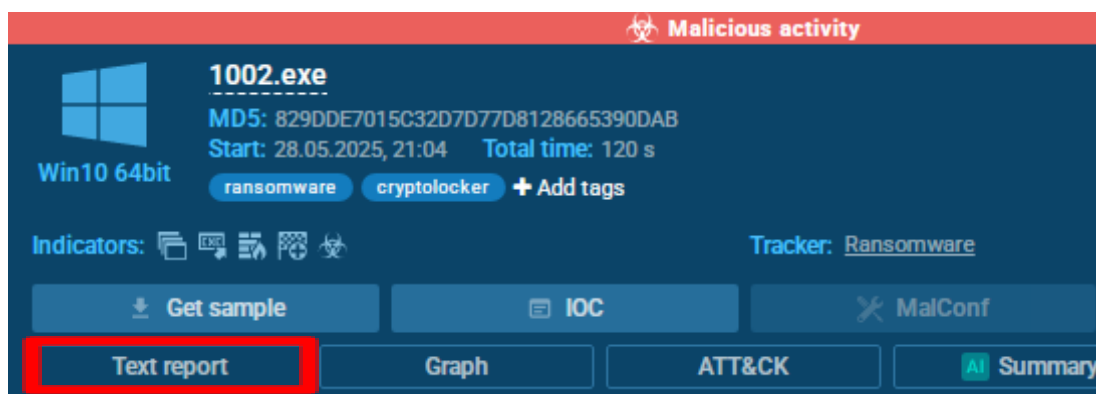
Otro de los recursos que nos ofrece any.run a la hora de realizar el análisis es el de generar un gráfico sobre cómo se ejecuta e interacciona el malware cuando se ejecuta.



El gráfico de procesos revela un comportamiento típico de ransomware, identificado específicamente como CryptoLocker. El malware (12ee0e91de.exe) es lanzado por un ejecutable inicial (1002.exe) y procede a interactuar con múltiples procesos legítimos del sistema. Destaca el uso de taskkill.exe para terminar procesos, lo que sugiere una preparación del entorno para sus operaciones. Además, la interacción con conhost.exe indica la ejecución de comandos de línea.

La presencia de spptxcomobj.exe y slui.exe podría apuntar a intentos de manipular los mecanismos de licencia o activación de Windows, mientras que la interacción con svchost.exe es un indicio fuerte de inyección de código o persistencia a nivel de servicio, buscando operar con privilegios elevados y evadir la detección.

Además de estos apartados, any.run nos permite ver un análisis a fondo en texto plano seleccionando el siguiente botón.



En este apartado podemos analizar otros aspectos del análisis realizado por any.run como por ejemplo los eventos de modificaciones y los procesos.

Registry activity

☒ Add for printing

Total events	Read events	Write events	Delete events
2 236	2 217	19	0

Modification events

(PID) Process: (6700) 1002.exe	Key: HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
Operation: write	Name: 12EE0E91DE
Value: C:\Users\admin\AppData\Roaming\12EE0E91DE.exe	

Análisis del Comportamiento del Malware.

El análisis revela una cadena de infección donde el proceso inicial 1002.exe (PID 6700) ejecuta y despliega la carga útil principal, identificada como 12EE0E91DE.exe (PID 5392). Esta secuencia instala el malware principal, el cual utiliza un nombre aleatorio generado dinámicamente para su ejecución.

Técnicas de Persistencia Críticas

El malware establece una persistencia redundante y robusta en el sistema mediante la modificación de dos claves críticas del registro de Windows. Específicamente, se ha detectado la alteración de las claves Run y RunOnce dentro de HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion, ambas apuntando a la ruta C:\Users\admin\AppData\Roaming\12EE0E91DE.exe.

(PID) Process: (5392) 12EE0E91DE.exe	Key: HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
Operation: write	Name: 12EE0E91DE
Value: C:\Users\admin\AppData\Roaming\12EE0E91DE.exe	

(PID) Process: (5392) 12EE0E91DE.exe	Key: HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
Operation: write	Name: *12EE0E91DE
Value: C:\Users\admin\AppData\Roaming\12EE0E91DE.exe	

Esta duplicación garantiza que el software malicioso se ejecute automáticamente tanto en cada inicio de sesión del usuario como tras cualquier reinicio del sistema, asegurando su supervivencia incluso si una de las entradas fuera eliminada.

Evasión y Ocultación.

El malware emplea técnicas para evadir las defensas y ocultar su actividad. Se ha detectado una manipulación del menú contextual del explorador de Windows a través de la modificación de la clave

HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SlowContextMenuEntries. Esta alteración podría tener como objetivo deshabilitar opciones del menú contextual relacionadas con la seguridad, ocultar funcionalidades de análisis de archivos o incluso ralentizar la respuesta del sistema para dificultar el análisis dinámico.

(PID) Process: (6700) 1002.exe	Key: HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer
Operation: write	Name: SlowContextMenuEntries
Value: 6024B221EA3A6910A2DC08002B30309D0A0100008D0E0C47735D584D9CEDE91E22E23282770100000114020000000000C0000000000000468D00000006078A409B011A54DAFA526D86198A780390100009AD298B2EDA6DE11BA8CA68E55D895936E000000	

Además, el malware desactiva activamente los mecanismos de tracing del sistema. Múltiples entradas en HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\ han sido modificadas para establecer EnableFileTracing: 0, EnableAutoFileTracing: 0 y EnableConsoleTracing: 0. Esta configuración elimina por completo la capacidad del sistema para registrar la actividad del malware, dificultando enormemente la investigación forense y el análisis post-infección.

(PID) Process: (5392) 12EE0E91DE.exe	Key: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\12EE0E91DE_RASAPI32
Operation: write	Name: EnableFileTracing
Value: 0	

(PID) Process: (5392) 12EE0E91DE.exe	Key: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\12EE0E91DE_RASAPI32
Operation: write	Name: EnableAutoFileTracing
Value: 0	

(PID) Process: (5392) 12EE0E91DE.exe	Key: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\12EE0E91DE_RASAPI32
Operation: write	Name: EnableConsoleTracing
Value: 0	

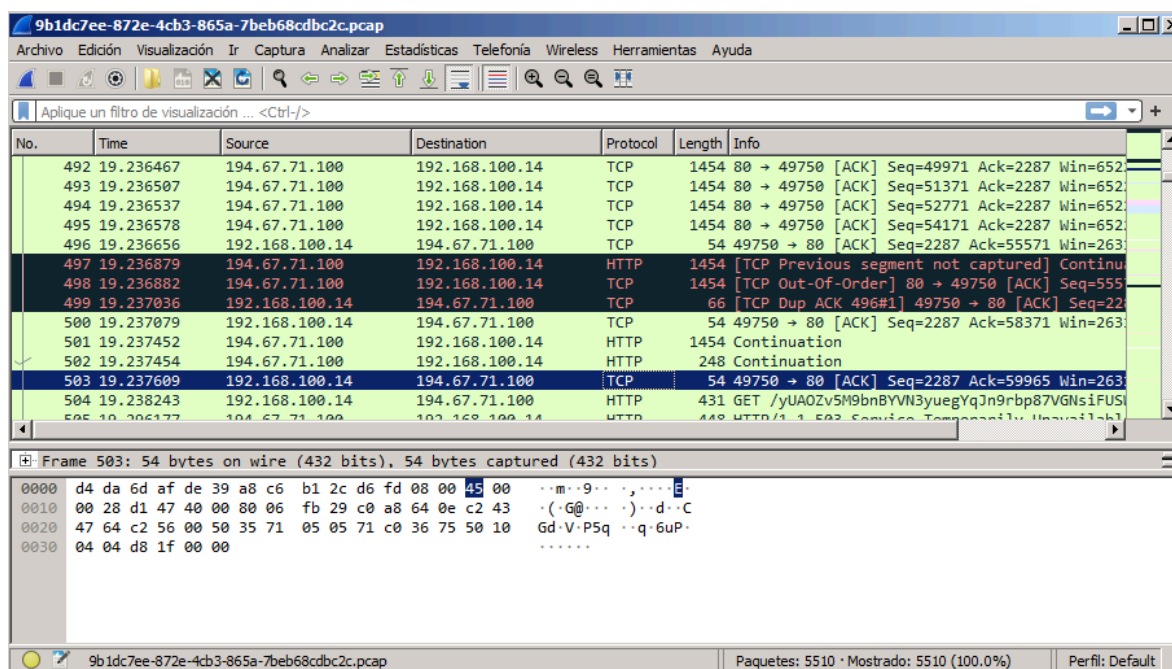
5.2. Wireshark.

Wireshark es una herramienta de análisis de tráfico de red que permite capturar y examinar en detalle los paquetes de datos que circulan por una red. En este caso, se utilizará para analizar el comportamiento de la comunicación de red generada por el malware CryptoLocker. Normalmente se buscan patrones sospechosos como conexiones a servidores C&C, IPs en listas negras, y tráfico cifrado o codificado típico de ransomware.

Para conseguir el archivo .pcap necesario para analizar las conexiones del malware vamos a utilizar la función de any.run que nos permite descargarlo desde el siguiente botón de la ventana de resultados.



Una vez que descargamos el archivo podemos abrirlo con wireshark para analizarlo.



5.1.1. Resumen del Tráfico de Red

El tráfico analizado muestra una comunicación repetitiva entre la máquina local con la dirección IP 192.168.100.14 y un servidor remoto ubicado en la dirección IP 194.67.71.100. Esta interacción se realiza principalmente mediante solicitudes HTTP GET hacia un recurso que contiene una URL codificada de gran longitud, lo cual sugiere un intercambio de información estructurado y automatizado.

El protocolo utilizado es HTTP, sin presencia visible de capas cifradas como HTTPS/TLS en la mayor parte del tráfico. Los paquetes tienen un tamaño constante de 431 bytes, lo cual refuerza la hipótesis de que se trata de un proceso automatizado típico de malware. La frecuencia regular de las conexiones indica que probablemente se está manteniendo una comunicación persistente con un servidor de control (C&C), común en ransomware como CryptoLocker.

5.1.2. Características Sospechosas Identificadas

Se han observado varias características que indican comportamiento malicioso, el número elevado de peticiones idénticas realizadas desde la máquina infectada apunta a un proceso no interactivo, típico de malware.

Las URLs utilizadas contienen cadenas extensas que parecen estar diseñadas para evitar la detección por parte de sistemas de seguridad. Es probable que estas cadenas contengan información sensible como identificadores únicos del sistema afectado o datos de cifrado.

En los registros analizados no se ha podido identificar el campo User-Agent, lo que sugiere que las solicitudes no están siendo generadas por un navegador convencional, sino por una herramienta o librería personalizada.

5.1.3. Análisis de IPs Involucradas

IP Local: 192.168.100.14. Máquina desde la cual se originan todas las peticiones. Se encuentra dentro del rango de direcciones IP privadas, por lo que corresponde al entorno interno de la red comprometida.

IP Remota: 194.67.71.100. Servidor externo que responde a las solicitudes realizadas por la máquina infectada. Esta dirección IP tiene reputación sospechosa y ha sido vinculada anteriormente a actividades relacionadas con CryptoLocker.

Otras IPs encontradas. 4.245.163.56: Aparece varias veces en el análisis y pertenece a la infraestructura de Microsoft Azure.

5.1.4. Comunicación C&C.

La comunicación establecida entre la máquina local (192.168.100.14) y el servidor remoto (194.67.71.100) cumple con las características típicas de una conexión C&C utilizada por malware.

La función más probable de esta comunicación sería el registro del dispositivo infectado, obtención de clave pública, confirmación del estado de cifrado o envío de identificadores únicos del sistema. Las conexiones periódicas con payloads constantes, típico de malware que mantiene sincronización activa con sus servidores de control.

Algunas respuestas indican errores HTTP 503 ("Service Temporarily Unavailable"), lo cual puede deberse a que el servidor C&C esté fuera de servicio o sea intencionado para evitar el análisis.

Esta infraestructura de control es fundamental para el funcionamiento de CryptoLocker, ya que permite la coordinación del cifrado y la recepción de claves necesarias para descryptar los archivos tras el pago del rescate.

6. Análisis de Código.

6.1. Ghidra.

El análisis del código del archivo sospechoso realizó empleando la herramienta Ghidra. El objetivo inicial era desensamblar y descompilar el ejecutable para comprender su lógica interna y funciones clave.

Tras la importación del binario en Ghidra, se procedió a su análisis automático. Sin embargo, las primeras observaciones en la vista del descompilador y del desensamblador revelaron una cantidad de información extremadamente limitada y poco interpretable. A continuación podemos ver la función que importa la librería mscoree.dll.

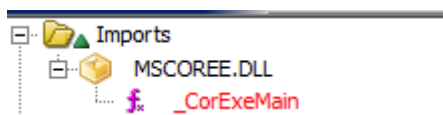
```

undefined      undefined _CorExeMain()
                <UNASSIGNED> <RETURN>
0 _CorExeMain <<not bound>>
PTR__CorExeMain_00402000      XREF[3]: 004000ac(*), 00400158(*),
                                00400184(*)

00402000 c0 fd 03 00      addr      MSCOREE.DLL::_CorExeMain
00402004 00              ??      00h
00402005 00              ??      00h
00402006 00              ??      00h
00402007 00              ??      00h

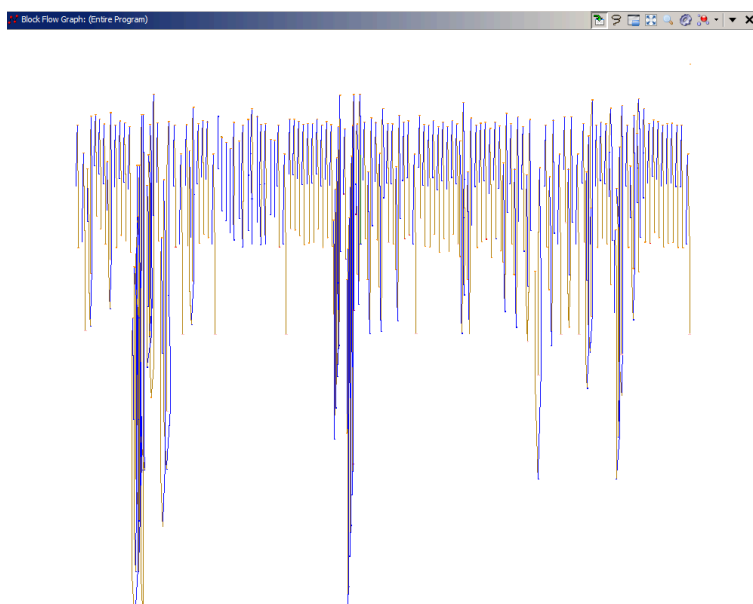
                                IMAGE_COR20_HEADER_00402008      XREF[1]: 00400168(*)
    
```

La pestaña "Imports" mostraba un estado prácticamente vacío, y el código inicial se reducía a una función genérica (_CorExeMain), o a funciones con nombres genéricos (nullsub_X).



La generación de un gráfico de flujo de bloques a nivel de programa completo también arrojó una representación excesivamente densa y ruidosa, haciendo bastante complicado el realizar un análisis pormenorizado.

La incapacidad de Ghidra para presentar un código legible o para identificar claramente las funciones internas y sus relaciones, a pesar de sus potentes capacidades de descompilación, es un indicador clave de que el código real del malware está oculto o protegido.



La incapacidad de Ghidra para presentar un código inteligible o para identificar claramente las funciones internas y sus relaciones, a pesar de sus potentes capacidades de descompilación, es un indicador clave de que el código real del malware está oculto o protegido. Este hallazgo indica que no se puede analizar este malware eficazmente con Ghidra o IDA Pro en su modo estándar de desensamblaje nativo. Necesitamos un descompilador de .NET como ILSpy.

6.2. ILSpy.

He usado la herramienta ILSpy para ver que podía obtener con ella y conseguí la siguiente información.

```
1002 (1.0.0.0, .NETFramework, v2.0)

// C:\Users\master\Downloads\CryptoLocker_22Jan2014\1002.exe
// Microsoft Windows Auto Update, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null
// Global type: <Module>
// Entry point: 1002.exe
// Architecture: AnyCPU (64-bit preferred)
// Runtime: v2.0.50727
// Hash algorithm: SHA1

using System;

[assembly: AssemblyCopyright("Copyright © 2013")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyProduct("Microsoft Windows Auto Update")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyCompany("")]
[assembly: Debuggable(DebuggableAttribute.DebuggingModes.IgnoreSymbolStoreSequencePoints)]
[assembly: CompilationRelaxations(8)]
[assembly: AssemblyFileVersion("1.0.0.0")]
[assembly: ComVisible(false)]
[assembly: Guid("cfe6d6f2-f8aa-45fa-bab9-3522be66bf82")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyTitle("Microsoft Windows Auto Update")]
[assembly: RuntimeCompatibility(WrapNonExceptionThrows = true)]
[assembly: AssemblyVersion("1.0.0.0")]
[module: SuppressIldasm]
[module: ConfusedBy("Confuser v1.9.0.0")]
```

Entre los metadatos obtenidos del ensamblado encontramos la siguiente información que nos confirma el carácter malicioso del malware y reafirma otros hallazgos que hemos realizado anteriormente.

AssemblyProduct("Microsoft Windows Auto Update") y AssemblyTitle("Microsoft Windows Auto Update"): El malware se está haciendo pasar por una actualización legítima de Microsoft. Este es un método clásico de ingeniería social para engañar al usuario y lograr la ejecución. Es una táctica de evasión y ocultación de identidad.

AssemblyDescription(""), AssemblyCompany(""), AssemblyTrademark(""): La ausencia de estas descripciones también es sospechosa. Un software legítimo suele tener estos campos rellenos.

AssemblyVersion("1.0.0.0"), FileVersion("1.0.0.0"): Una versión genérica que a menudo usan los malwares para no revelar su verdadera identidad.

[module: SuppressIldasm]: Esta es una directiva muy interesante. Ildasm.exe es la herramienta de desensamblado oficial de Microsoft para el código IL de .NET. SuppressIldasm significa que el autor ha intentado evitar que el programa sea fácilmente desensamblado con la herramienta oficial. Esto es una forma de ofuscación o anti-análisis.

[module: ConfusedBy("Confuser v1.9.0.0")]: Esta línea nos confirma que se utilizó la herramienta Confuser con su versión 1.9.0.0. Este Confuser es un ofuscador de código diseñado para programas desarrollados en la plataforma .NET. Su propósito es dificultar la ingeniería inversa y el análisis del código, haciendo que sea más complicado de leer y comprender incluso con descompiladores. Los ciberdelincuentes lo utilizan frecuentemente para proteger su malware, dificultando su detección y estudio.

7. Conclusiones.

7.1. Resumen de los hallazgos más importantes de los análisis.

El análisis de la muestra de malware, identificada como una variante de CryptoLocker, ha revelado una serie de comportamientos y características clave tanto a nivel estático como dinámico.

Desde la perspectiva del análisis estático, la muestra mostró una gran ofuscación, con varios indicadores de empaquetamiento y protecciones anti-análisis, dificultando la inspección directa de su código. Se observó el uso de librerías y APIs de Windows asociadas comúnmente con la manipulación de archivos, procesos y la red, lo que muestra su capacidad para interactuar profundamente con el sistema operativo y establecer comunicaciones externas.

El análisis dinámico en any.run confirmó y nos ayudó a profundizar en los hallazgos, revelando una cadena de infección donde un dropper inicial (1002.exe) despliega el payload principal (12EE0E91DE.exe). Este proceso malicioso mostró una persistencia agresiva mediante la modificación de las claves Run y RunOnce del registro, asegurando su ejecución automática.

Además, se detectaron varias técnicas de evasión de defensas, incluyendo la deshabilitación del tracing del sistema y la manipulación del menú contextual para dificultar la detección y el análisis forense. El malware también realizó actividades de descubrimiento del sistema e intentó detener servicios (observado con taskkill.exe), también se observó una interacción o inyección en svchost.exe, lo que indica intentos de ocultación y posible escalada de privilegios.

El análisis de tráfico de red con Wireshark se identificó una comunicación saliente crítica desde la máquina comprometida (192.168.100.14) hacia una dirección IP externa 194.67.71.100. Este tráfico, era realizado por solicitudes GET a rutas ofuscadas, es muy probable que sea comunicación de Comando y Control (C&C) típica del ransomware. Aunque se observaron respuestas HTTP 503 Service Temporarily Unavailable, el intento continuo de comunicación confirma la intención del malware de contactar con su infraestructura para funciones como el registro de la infección, la obtención de claves de cifrado o la recepción de comandos.

7.2. Evaluación de la peligrosidad y complejidad de CryptoLocker.

CryptoLocker, tal como se ha analizado, se clasifica como una amenaza de alta peligrosidad y complejidad considerable.

Su peligrosidad radica principalmente en:

- **Capacidad de Impacto:** Al ser un ransomware, su objetivo final es el cifrado de archivos críticos del usuario y del sistema, lo que lleva a la denegación de acceso a la información y la potencial pérdida de datos irrecuperable sin la clave de descifrado, o el pago de un rescate.
- **Persistencia Robusta:** La implementación de múltiples puntos de persistencia (claves Run y RunOnce) asegura que el malware sobreviva a los reinicios del sistema, manteniendo la infección activa y prolongando el impacto.
- **Evasión Activa:** Las técnicas empleadas para deshabilitar logging y manipular el explorador de Windows demuestran un intento deliberado de operar de forma sigilosa, dificultando tanto la detección por parte de las soluciones de seguridad como la investigación forense post-incidente.
- **Comunicación C&C:** La capacidad de comunicarse con un servidor externo para coordinar su operación (obtener claves, reportar estado) lo convierte en una amenaza adaptativa y controlada, lo que eleva su peligrosidad.

La complejidad de CryptoLocker se manifiesta en:

- **Ofuscación y Empaquetamiento:** El uso de técnicas para dificultar el análisis estático sugiere un esfuerzo por evadir la detección basada en firmas y complicar la ingeniería inversa.
- **Comportamiento Multi-fase:** La ejecución de un dropper inicial que instala el payload principal indica un diseño modular y una cadena de ataque estructurada.
- **Uso de Utilidades Legítimas:** El uso de herramientas y procesos legítimos de Windows (taskkill.exe, sppextcomobj.exe, slui.exe, svchost.exe) para realizar acciones maliciosas o para ocultarse, es una táctica avanzada que complica la detección.

8. Bibliografía y referencias.

- <https://www.seguridad.unam.mx/historico/noticia/index.html-noti=1751>
- <https://www.avast.com/es-es/c-cryptolocker>
- <https://www.proofpoint.com/es/threat-reference/cryptolocker>
- <https://www.hornetsecurity.com/es/knowledge-base/cryptolocker-ransomware/>
- <https://www.incibe.es/ciudadania/ayuda/ransomware>
- <https://stackoverflow.com/questions/12151308/confuser-net-obfuscator-is-it-safe>
- <https://learn.microsoft.com/es-es/dotnet/api/system.reflection.assemblytitleattribute?view=net-9.0>