INFORME DEL PROYECTO

ANÁLISIS DE MALWARE

XCLIENT.EXE

Propuesta creada por
Alejandro Delgado

ÍNDICE

INTRODUCCIÓN

En la era actual de la tecnología y la información, la seguridad cibernética se ha convertido en una prioridad para individuos, empresas y gobiernos. Entre las diversas amenazas en el ciberespacio, el malware representa uno de los desafíos más persistentes y evolutivos. Este proyecto tiene como objetivo realizar un análisis exhaustivo de dos muestras específicas de malware, proporcionando una comprensión profunda de sus características, comportamientos y potenciales amenazas. El análisis del malware no solo es crucial para defenderse contra ataques cibernéticos específicos, sino también para entender y anticipar las estrategias y evoluciones futuras de las amenazas cibernéticas.

Contexto del Análisis de Malware

El malware, o software malicioso, es un término general que abarca varios tipos de programas dañinos o intrusivos diseñados para afectar negativamente el funcionamiento de dispositivos informáticos. A lo largo de los años, el malware ha evolucionado de simples virus creados por aficionados a sofisticadas herramientas de ciberataque utilizadas por criminales y, en ocasiones, por actores estatales. Este cambio ha llevado a un aumento en la complejidad y el impacto de los ataques de malware, haciendo que su análisis y comprensión sean cruciales para la seguridad informática moderna.

ANTECEDENTES

Para comprender plenamente la importancia y el alcance de este proyecto de análisis de malware, es esencial contextualizarlo dentro del panorama más amplio de la seguridad informática y la evolución del malware.

Historia del Análisis de Malware

El concepto de malware no es nuevo. Desde los primeros días de la informática, ha habido instancias de software diseñado para infiltrarse o dañar sistemas informáticos. Inicialmente, los virus y el malware eran principalmente el producto de experimentos o bromas entre programadores, pero con el tiempo, estos programas maliciosos se convirtieron en herramientas para el cibercrimen, el espionaje y la guerra cibernética.

En las últimas décadas, la proliferación de internet y la creciente dependencia de la tecnología en todos los aspectos de la vida han hecho que el malware sea una amenaza significativa y en constante evolución. El análisis de malware, por lo tanto, ha pasado de ser una curiosidad técnica a una necesidad crítica para proteger la información y los sistemas.

Evolución del Malware

El malware ha evolucionado en términos de sofisticación, impacto y diversidad. Las primeras formas de malware eran relativamente simples y su propagación era limitada. Sin embargo, con el avance de la tecnología, el malware ha adquirido la capacidad de causar daños extensos, robar grandes volúmenes de datos y eludir las medidas de seguridad avanzadas.

Los tipos de malware incluyen virus, gusanos, troyanos, ransomware, spyware, adware, entre otros. Cada tipo tiene sus características y métodos de ataque únicos, lo que hace que el análisis de malware sea un campo complejo y multifacético.

Importancia del Análisis de Malware

El análisis de malware es fundamental para comprender cómo operan estos programas maliciosos y cómo se pueden mitigar sus efectos. A través del análisis, los expertos en seguridad pueden identificar vulnerabilidades, desarrollar medidas

de seguridad efectivas y prepararse mejor para enfrentar futuros ataques. Este conocimiento es crucial no solo para la protección individual, sino también para la seguridad nacional e internacional, dada la naturaleza global de las amenazas cibernéticas.

ANÁLISIS DE MALWARE - XClient.exe

Información Básica

File name: XClient.exe

md5: e47a5d191f0acb69797087ab43b24370

Sha256: 53c52719a416751e1171b9a3234ae4e12965b59a468aba486191a6327d54c739

Sha1: f62ae64a9ac4f8a8c48aa7ee012b1f69c8c59108

En una primera revisión tras el análisis de CAPE encontramos los siguientes puntos relevantes. Por lo que vemos, el fichero es un ejecutable con un nombre que parece no querer llamar la atención. Un fichero client.exe puede confundirse con herramientas de aplicaciones de red o componentes que utilizan los juegos o aplicaciones instalados en el ordenador.

CAPE Yara: XWorm - XWorm Payload - Author: ditekSHen

Por lo que vemos, una de las reglas yara residentes en CAPE a matcheado con el malware y lo ha identificado. Estamos hablando de XWorm, un malware sofisticado que presenta varias características y tácticas avanzadas de evasión y persistencia. Se trata de un Remote Access Trojan (RAT) que permite a los atacantes acceder y controlar de forma remota los sistemas infectados.

Aunque sea un malware conocido, seguiremos buscando información sobre el mismo para conocer en profundidad cuál es su funcionamiento y si podemos extraer la ubicación de quien puede estar utilizándolo con motivos maliciosos.

Importaciones

mscoree.dll:

- _CorExeMain: función de mscoree.dll que indica que el programa está utilizando .NET Framework.

Esta función es crucial para iniciar y administrar la ejecución de aplicaciones compiladas con .NET Framework. Cuando un programa que utiliza .NET Framework se ejecuta, _CorExeMain es la función que efectivamente arranca el programa y coordina su ejecución con el entorno de tiempo de ejecución de .NET.

Esto puede tener varias implicaciones:

- **Obfuscación y Evasión**: Los malware que utilizan .NET pueden aprovechar diversas técnicas de ofuscación que están disponibles para los programas .NET. Esto puede hacer más difícil su análisis y detección por parte de los software antivirus y herramientas de seguridad.
- **Inyección de Código:** El malware puede utilizar .NET para inyectar código malicioso en procesos legítimos en ejecución. Esto puede permitirle operar bajo el radar, evitando ser detectado, ya que se esconde dentro de procesos que pueden parecer legítimos.
- **Acceso a Funcionalidades Avanzadas:** .NET Framework ofrece una amplia gama de funcionalidades y API que el malware podría explotar para realizar actividades maliciosas, como recolección de datos, comunicaciones de red, manipulación de archivos, etc.
- **Compatibilidad y Portabilidad:** Al utilizar .NET Framework, el malware puede ser más compatible con diferentes versiones de Windows y, potencialmente, con otros sistemas operativos que soportan .NET. Esto podría aumentar el alcance de los sistemas que el malware puede infectar.
- **Facilidad de Desarrollo y Actualización:** Para los desarrolladores de malware, usar .NET puede facilitar el desarrollo y la actualización del malware debido a la flexibilidad y las potentes herramientas de desarrollo disponibles para .NET.

NTdll.dll:

- RtlSetProcessIsCritical: Esta función puede ser usada para marcar un proceso como crítico. Si XWorm utiliza esta función, podría hacer que el sistema operativo se detenga o muestre una pantalla azul de la muerte (BSOD) si el proceso del malware se cierra. Esto se puede utilizar como una táctica de coacción o para mantener la persistencia del malware en el sistema.

kernel32.dll:

- GetModuleHandle: Permite al malware encontrar módulos (bibliotecas o ejecutables) cargados en su espacio de memoria, lo cual puede ser útil para realizar inyecciones de código u otras formas de manipulación.
- CheckRemoteDebuggerPresent: Esta función se utiliza para detectar si el proceso está siendo depurado. XWorm podría usarlo para identificar y evadir herramientas de análisis de seguridad.
- SetThreadExecutionState: Permite al malware manipular el estado de ejecución del sistema para evitar que la computadora entre en modo de suspensión, lo cual puede ser útil para mantener operaciones maliciosas en curso.

user32.dll:

- SetWindowsHookEx, UnhookWindowsHookEx y CallNextHookEx: Estas funciones permiten instalar ganchos que pueden monitorizar la entrada del usuario, como las pulsaciones de teclas, lo cual es esencial para el keylogging.
- GetForegroundWindow y GetWindowThreadProcessId: Estas funciones permiten al malware determinar qué ventana y proceso están activos en primer plano, lo que puede ser útil para el robo de información específica de la aplicación.

- GetKeyState, GetKeyboardState, GetKeyboardLayout, ToUnicodeEx y MapVirtualKey: Estas funciones son útiles para interpretar correctamente las pulsaciones de teclas capturadas, permitiendo un keylogging más preciso.
- GetLastInputInfo y GetWindowText: Estas funciones pueden ser utilizadas para recopilar información sobre la actividad del usuario y las aplicaciones en uso.

avicap32.dll:

- capCreateCaptureWindowA y capGetDriverDescriptionA: Estas funciones se utilizan para interactuar con dispositivos de captura de video como cámaras web. El malware podría utilizarlas para capturar video o imágenes sin el conocimiento del usuario.

STRINGS

Tras analizar los strings, hemos verificado los puntos más relevantes que nos aportarán más conocimiento sobre el RAT que estamos analizando:

- **User-Agent: Mozilla/5.0...**: Indica que el malware podría estar imitando un navegador web para evadir la detección o para realizar actividades de red.
- **System.Threading, AsyncCallback, ThreadStart**: Sugeriría que el malware utiliza múltiples hilos o realiza operaciones asincrónicas, lo cual es común en actividades maliciosas como la comunicación de red en segundo plano.
- **ICryptoTransform, AES_Decryptor, CreateEncryptor, GetRandomString, ComputeHash**: Estos strings sugieren que el malware podría estar utilizando cifrado para proteger su comunicación o datos.
- **powershell.exe, cmd.exe**: La presencia de estos strings indica que el malware podría estar ejecutando scripts o comandos a través de la línea de comandos de Windows o PowerShell.
- **System.Management, ManagementObjectSearcher**: Podría ser utilizado para recopilar información sobre el sistema o realizar operaciones de gestión del sistema.

- **GetWindowThreadProcessId, IsUpdate, ProcessStartInfo, ProcessCritical**: Implica que el malware puede interactuar con otros procesos o modificar su propio estado de proceso para evadir la detección.
- **RegistryKey, OpenSubKey, DeleteSubKey, HKEY_LOCAL_MACHINE**: Sugiere manipulación del registro de Windows, lo que podría significar cambios en la configuración del sistema o persistencia del malware.
- **WebClient, HttpWebRequest, DownloadString, POST / HTTP/1.1**: Indica capacidad de comunicación de red, posiblemente para filtrar datos o recibir instrucciones de un servidor de comando y control.
- **USBCode, USBStart, USBStop**: Esto puede indicar que el malware tiene la capacidad de propagarse o realizar operaciones a través de dispositivos USB.
- **DetectDebugger, isDebuggerPresent, DetectSandboxie, anyrun**: Estos strings sugieren que el malware puede tener la capacidad de detectar si se está ejecutando en un entorno de análisis o sandbox.
- **shutdown.exe, Restart, PCShutdown, PCLogoff**: Podría indicar que el malware tiene la capacidad de apagar o reiniciar el sistema infectado.
- **Service Pack, System.IO.Compression, .NET Framework Strings**: Muestra dependencia o interacción con componentes específicos del sistema operativo.
- **System.Net.Sockets, Socket, Connect, WebClient**: Implica funcionalidades de red, posiblemente para la comunicación remota o actividades de botnet.
- **Encryptor, Decryptor, RijndaelManaged**: Indica el uso de algoritmos de cifrado, posiblemente para cifrar archivos (como en ransomware) o comunicaciones.
- **Microsoft.VisualBasic.ApplicationServices, My.Computer**: Sugiere que el malware está utilizando librerías específicas de VB.NET, lo que puede dar pistas sobre su lenguaje de programación.

MITRE

Command and Control → T1071 - Application Layer Protocol

- ***stealth_network:*** Esta técnica permite a XWorm comunicarse de manera discreta con un servidor de comando y control (C2). Al usar protocolos de la capa de aplicación para la comunicación, el malware puede camuflar su tráfico como tráfico de red normal, lo que hace que sea más difícil para los sistemas de seguridad detectar su presencia. Esto es esencial para recibir instrucciones del operador del malware, enviar datos recopilados o descargar componentes maliciosos adicionales sin ser detectados.
- ***cape_detected_threat:*** El objetivo aquí es pasar desapercibido, especialmente evitando la detección por parte de firewalls y otros sistemas de seguridad de red. Al utilizar técnicas de evasión de red, como la fragmentación del tráfico o el uso de protocolos comunes que no generan sospechas, XWorm puede mantener su operatividad sin ser bloqueado o identificado por sistemas de seguridad.

Execution → T1106 - Native API

- antidebug_guardpages: Esta técnica se refiere a la capacidad de XWorm para detectar y responder a intentos de depuración. Si XWorm detecta que está siendo depurado, puede cambiar su comportamiento para ocultar su funcionalidad real o, en algunos casos, puede autodestruirse para evitar el análisis. Esto es crucial para eludir el análisis forense y evitar la detección de sus verdaderas capacidades.

FIRMAS

Se ha detectado conexiones a redes: ip: 93.184.220.29

- Comunicación externa con el atacante.

SetUnhandledExceptionFilter detected: posible antidebug.

- Puede detectar cuando se está intentando analizar y cambiar su comportamiento o cerrarse.

Guard pages use detected: métodos antidebbuging.

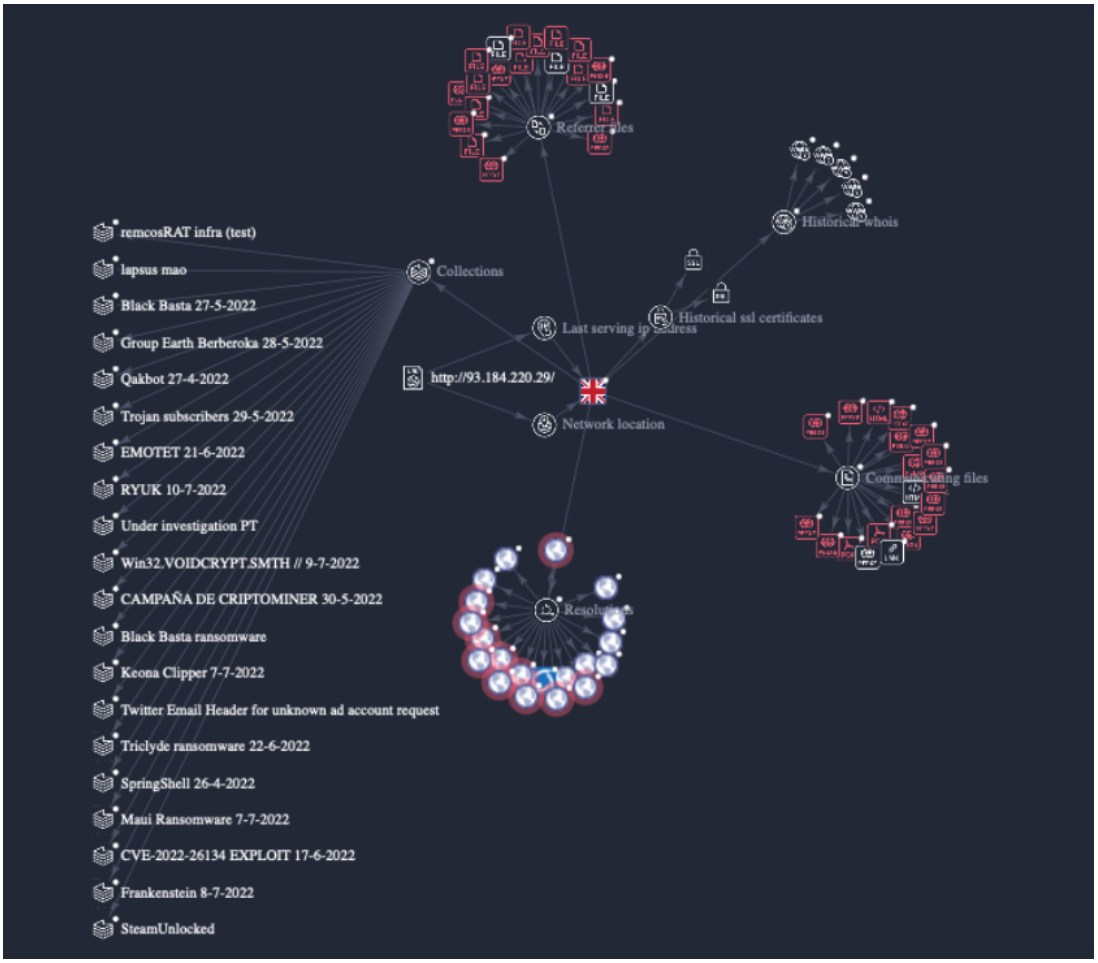
- La creación de páginas de guardia (guard pages) es una técnica antidebugging que impide que los depuradores accedan a ciertas áreas de la memoria, lo que dificulta el análisis del malware.

Operaciones en Memoria:

- APIs como NtAllocateVirtualMemory, VirtualProtectEx: Estas funciones de la API de Windows se utilizan para manipular la memoria virtual. Esto permite al malware crear y modificar regiones de memoria para sus operaciones, lo que incluye la inyección de código en otros procesos o la modificación de su propio código.
- Creates RWX memory: La creación de memoria con permisos de lectura, escritura y ejecución (RWX) indica que el malware puede estar generando código en tiempo de ejecución o modificando su propio código en memoria. Esto hace que el malware sea más resistente a la detección, ya que no depende de archivos en el disco que puedan ser escaneados por soluciones antivirus.

ANÁLISIS DE RED

La ip que analizaremos es la que hemos extraído del informe propuesto por CAPE y VirusTotal: **93.184.220.29**



Al analizar la red a través de VirusTotal, comprobamos que la ip que se está utilizando es una botnet (proxy) que está utilizando el atacante para ocultar su verdadera dirección y asi no desvelar su ubicación.

Por lo que podemos ver de la red compuesta, tiene relación con los siguientes malware y otros elementos maliciosos:

- remcosRAT
- Lapsus mão
- Black Basta
- Group Earth Berbería
- Qakbot
- Trojan subscribers

- EMOTET
- RYUK
- Under investigation PT
- Win32.VOIDCRYPT.SMTH
- CAMPAÑA DE CRIPTOMINER
- Black Basta ransomware
- Peona Clipper
- Twitter Email Header for unknown ad account request
- Tryclyde ransomware
- SpringShell
- Maui Ransomware
- CVE-2022-26134 EXPLOIT
- Frankenstein
- SteamUnlocked

La relación entre la dirección IP analizada y una variedad de amenazas cibernéticas y campañas maliciosas sugiere una infraestructura de ataque compartida o una táctica conocida como "infraestructura como servicio" en el ámbito del ciberdelito.

Infraestructura de Proxy y Ataques Múltiples

- **Proxy Interpuesto:** El uso de un proxy interpuesto entre el atacante y sus víctimas es una táctica común para ocultar la verdadera ubicación y la identidad del atacante. Este método dificulta el rastreo y la atribución de los ataques a una fuente específica.
- **Relaciones con Diversas Amenazas:** La asociación de esta IP con una amplia gama de amenazas, desde RATs (Remote Access Trojans) hasta ransomware y campañas de criptominería, indica que esta infraestructura podría estar siendo utilizada por múltiples actores de amenazas, posiblemente vendida o alquilada en el mercado negro cibernético.

Como podemos analizar, la infraestructura de red que estamos visualizando puede ser un servicio compartido entre varios grupos de ciberdelincuentes, por lo que será complicado atribuir el ataque a un sólo grupo.

CONCLUSIONES

En nuestro proyecto, hemos explorado el complejo y fascinante mundo del análisis de malware, centrándonos en un caso específico: XWorm. Este proyecto nos ha llevado a descubrir cómo el malware puede esconderse y operar de manera muy astuta, usando técnicas para pasar desapercibido y hacer cosas maliciosas sin ser detectado fácilmente.

Lo primero que hemos aprendido es que XWorm es como un ladrón muy hábil en el mundo digital. Utiliza herramientas y trucos para evitar ser atrapado, como esconderse en procesos legítimos del sistema y disfrazar su comunicación para parecer normal. También hemos visto cómo puede cambiar su comportamiento o incluso autodestruirse si se siente amenazado, como un espía que borra sus huellas.

Además, este proyecto nos ha mostrado que XWorm no trabaja solo. Hemos descubierto que se comunica con una red más amplia, posiblemente una infraestructura compartida usada por otros malwares. Esto es como si varios ladrones usaran la misma casa segura para planear sus crímenes. Esta conexión con otros malwares y campañas maliciosas nos ha hecho pensar en la importancia de estar siempre un paso adelante en la seguridad informática.

En conclusión, este proyecto nos ha abierto los ojos sobre la importancia de proteger nuestros sistemas informáticos. Hemos aprendido que el mundo del malware es ingenioso y siempre está evolucionando, lo que significa que nosotros también debemos ser astutos y proactivos en nuestra defensa. La colaboración y el compartir información son clave para mantenernos seguros en este eterno juego del gato y el ratón en el ciberespacio.

TICKETING

ID: 20

Usuario: alumno04@keepcoding.io