Analizando el programa “unknown.exe”

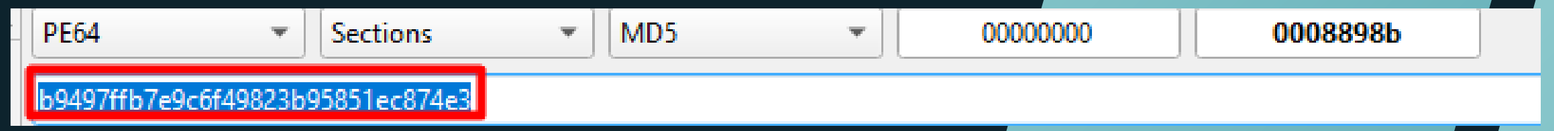
¿Será benigno?

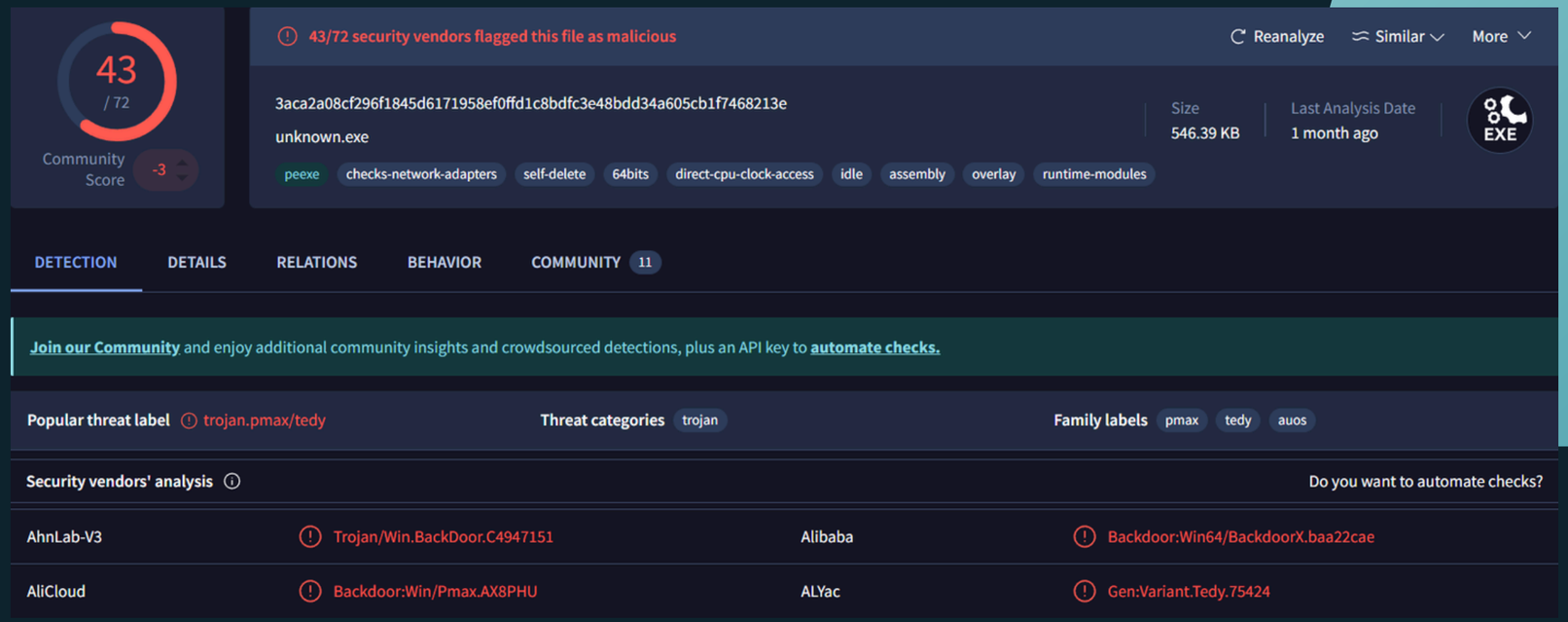
Elian Amarilla

Hash del programa

El hash del archivo es:

MD5: b9497ffb7e9c6f49823b95851ec874e3





Arquitectura y Compilación

Arquitectura

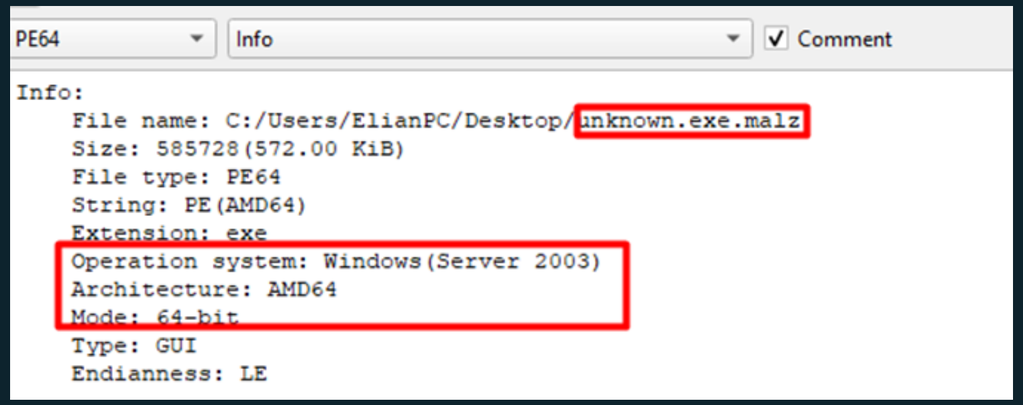
Está basado en la arquitectura **AMD de 64 bits**

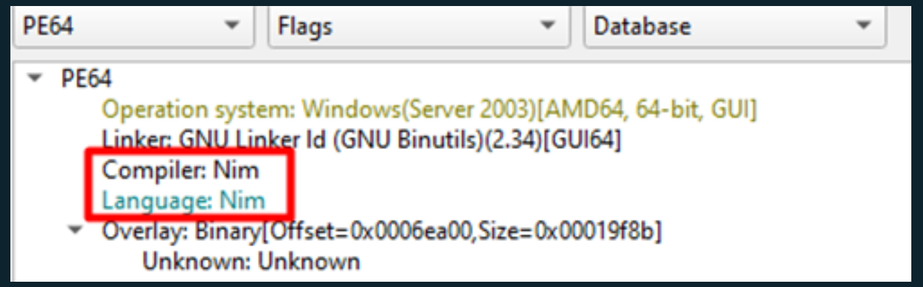
Fecha compilación

El programa fue compilado el día 08 de Enero del 2022 a las 13:29:18hs desde un sistema operativo Windows 2003.

Lenguaje

Está desarrolado en **NIM**





Strings relevantes

Indicadores de posible utilización de algoritmos de encriptación

11365	00080140		Overlay	1f	A	TM__Q5wkpxktOdTGvlSRo9bzt9aw_32
11366	00080160		Overlay	10	A	genericSeqAssign
11367	00080171		Overlay	10	A	raiseFieldError2
11368	00080182		Overlay	1f	A	TM__Q5wkpxktOdTGvlSRo9bzt9aw_33
11369	000801a2		Overlay	1f	A	TM__Q5wkpxktOdTGvlSRo9bzt9aw_34
11370	000801c2		Overlay	0c	A	nimBoolToStr
11371	000801cf		Overlay	1f	A	TM__Q5wkpxktOdTGvlSRo9bzt9aw_36
11372	000801ef		Overlay	1f	A	TM__Q5wkpxktOdTGvlSRo9bzt9aw_35

- Utilización del algoritmo RC4

11420	000806a5		Overlay	3d	A	toRC4__OOZOOZOOZOOZOOZOnimbleZpkgsZ8267…

URLs sospechosas y rutas de archivos.

- http://cdn.altimiter.local/feed?post=
- C:\Users\Public\passwrd.txt
- Desktop\cosmo.jpeg.

Strings relevantes

Otras cadenas de texto importantes

- **@SikoMode**: Buscando información al respecto, nos dice que es un stealer encontrado el 2 de julio de 2023.
- **SID_BindHost**: Proporciona acceso a un host que puede cargar recursos.
- **IID_IInternetBindInfo**: Permite definir cómo se debe vincular un recurso de red.
- **IID_IInternetBindInfoEx**: Igual que el anterior, pero con más control de seguridad y contexto.
- **WSAStartup**: WSAStartup inicializa la biblioteca de sockets en Windows (Winsock), permitiendo que una aplicación pueda usar funciones de red como crear sockets, enviar y recibir datos
- **@:houdini**
- **@Password**
- **@Hostname**
- ***was_here**

Algunos módulos que utiliza en lenguaje NIM

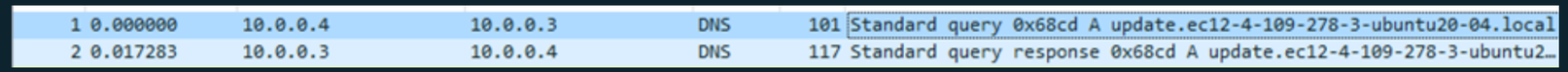
- **net.nim**: permite hacer programas de red en Nim, como clientes y servidores, usando sockets.
- io.nim
- streams.nim
- strutils.nim

Primer ejecución.
¿Qué pasa?

- Al ejecutarse, el malware intenta establecer conexión con un dominio llamado **ec12-4-109-278-3-ubuntu20-04.local**.

- Si no lo logra, se **borra** por sí solo.

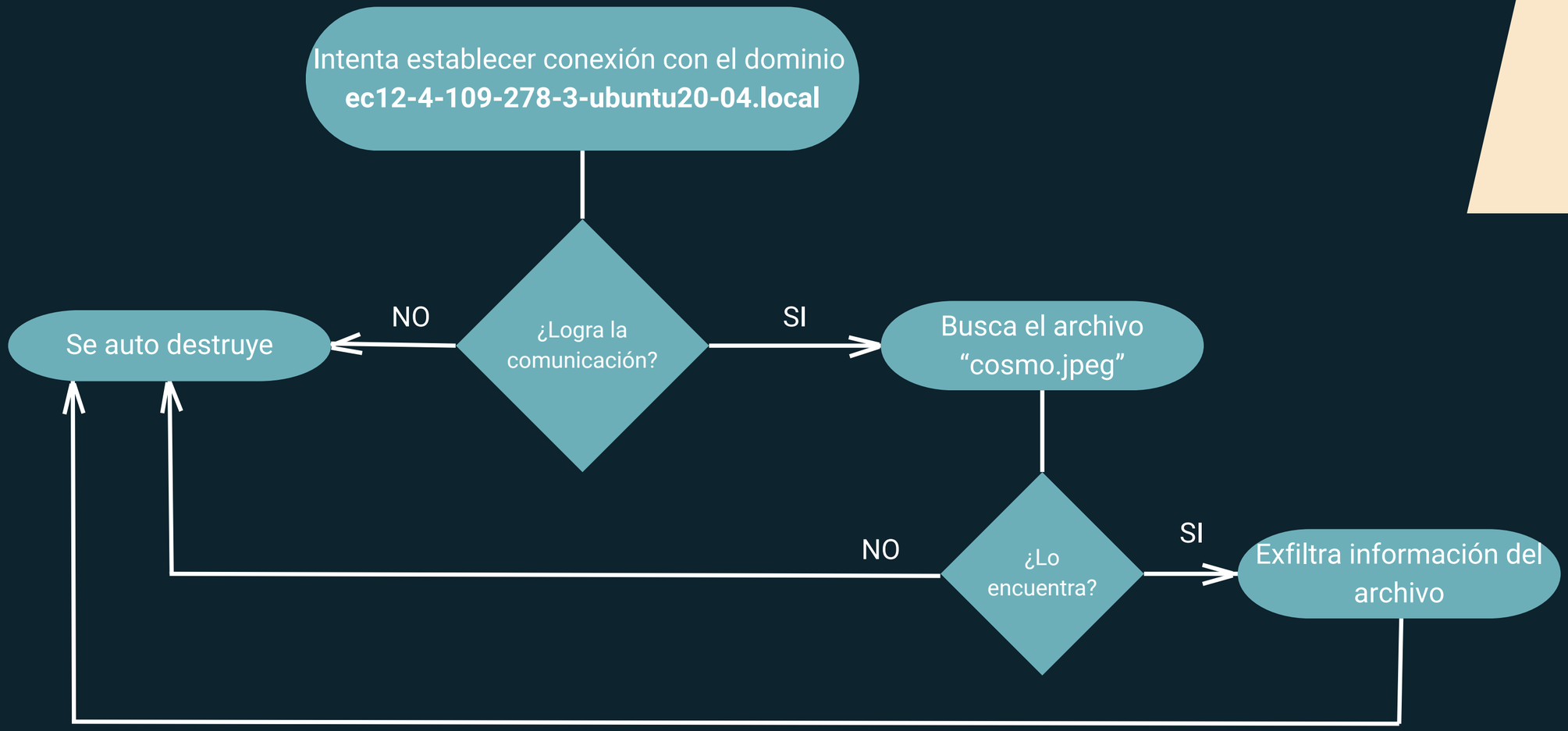
- En caso de tener éxito, busca en el escritorio un archivo llamado cosmo.jpeg. Si ese archivo no existe, **nuevamente se elimina automáticamente**. Pero si lo encuentra, lo extrae (**exfiltra**) y, una vez completada esa acción, también se autodestruye.



La función houdini es una parte oculta del código que se encarga de borrar o desactivar el malware para evitar ser detectado o analizado.

Funcionamiento

Flujo



¿Genera **persistencia**?

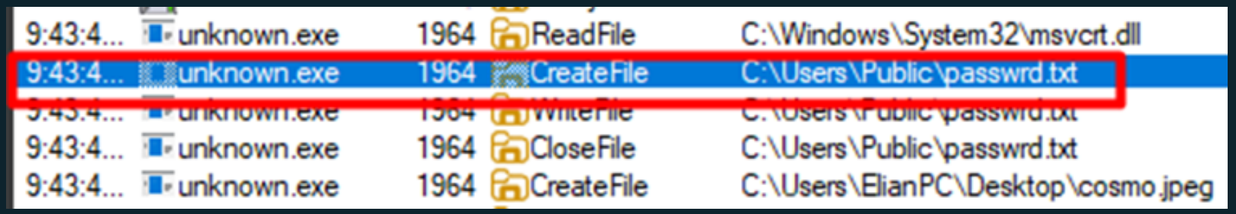
No implementa ningún mecanismo de persistencia.

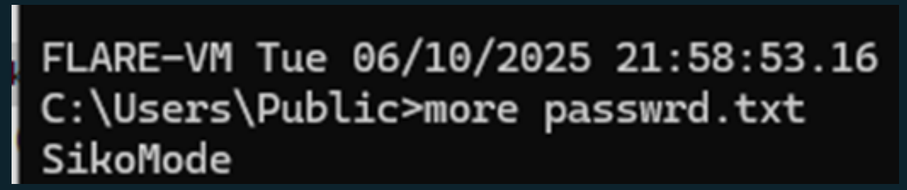
Su comportamiento es puntual: una vez que se ejecuta y logra completar la exfiltración de datos se elimina por completo.

Por dicho motivo es sigiloso y de único uso, minimizando la posibilidad de ser detectado en análisis posteriores.

¿Crea **archivos**?

Sí, en su ejecución el malware crea un archivo llamado “passwrd.txt” en la ruta **C:\Users\Public**

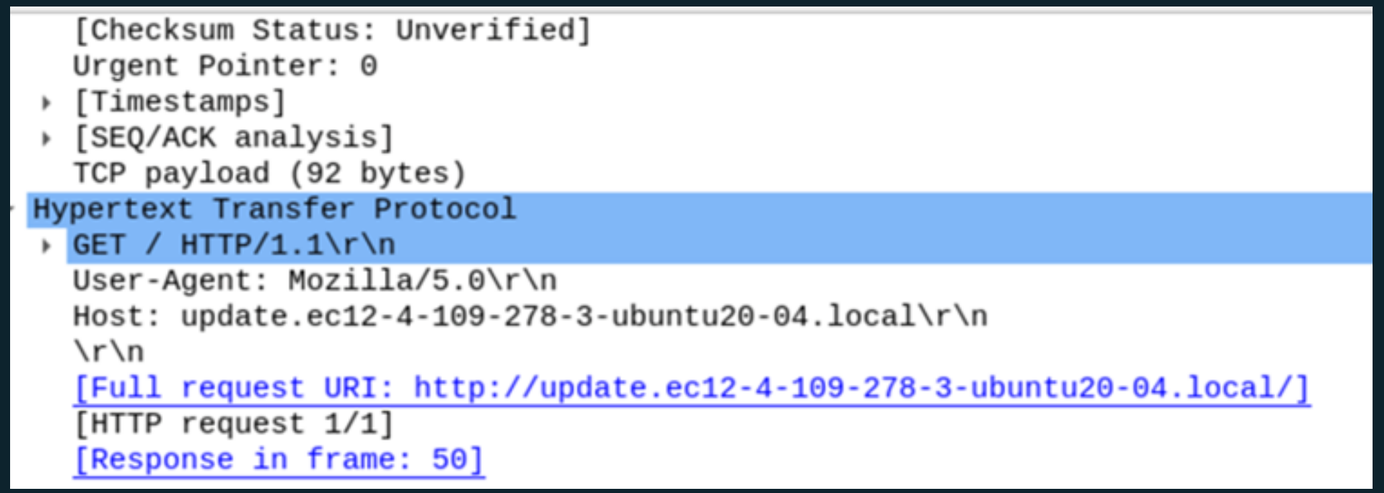




¿Qué **dominio** invoca inicialmente?

La primera actividad de red que realiza el malware puede observarse con Wireshark y consiste en una conexión al dominio **update.ec12-4-109-278-3-ubuntu20-04.local.**

Evidencia:

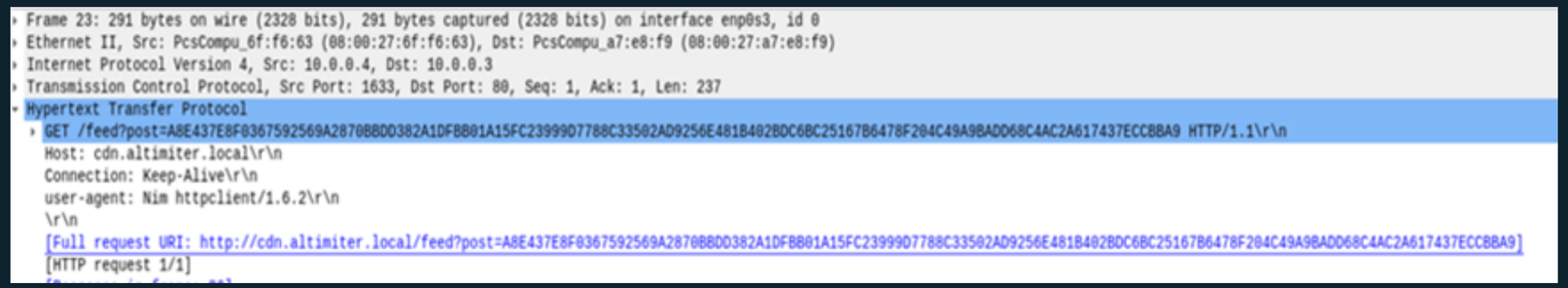


¿**Exfiltra** información?

Sí, si el archivo **cosmo.jpeg** está presente donde está el malware, el malware lo accede, lo lee y lo convierte a formato base64.

Luego, aplica un proceso de cifrado sobre esos datos codificados para finalmente exfiltrarlos. De esta manera dificulta el análisis.

Evidencia:



La exfiltración se realiza mediante solicitudes HTTP GET, enviando los datos cifrados al dominio de exfiltración **cdn.altimiter.local**.

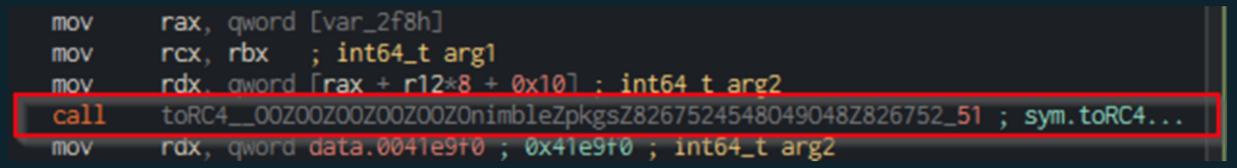
Información que **exfiltra**

La información que el malware exfiltra corresponde al contenido del archivo cosmo.jpeg, ubicado en el escritorio del usuario.

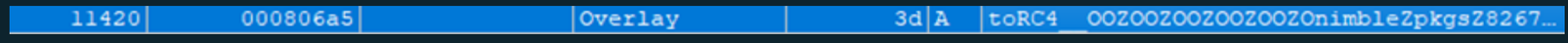
¿Tiene algún tipo de **cifrado** de información?

Sí, el malware utiliza el algoritmo de cifrado **RC4** para proteger los datos antes de enviarlos. Fue analizado con cutter para encontrar esta información, además teníamos el indicio de los strings en el análisis estático básico.

Análisis desde cutter:



Análisis desde DIE:



Clave del cifrado

La clave utilizada para descifrar los datos es "**SikoMode**", y está almacenada en un archivo llamado passwrd.txt, el cual es generado por el propio malware.

¿Tiene algún tipo de ofuscación de **código**?

El código del malware no presenta técnicas de ofuscación, lo que hace que tanto el análisis estático como el análisis dinámico sean más sencillos de llevar a cabo.

¿Qué tipo de **Malware** es?

Este malware es un **stealer**, diseñado específicamente para robar información del sistema infectado

Su comportamiento es selectivo, ya que solo extrae datos en determinadas condiciones, como la presencia del archivo cosmo.jpeg y la conexión exitosa con su servidor de comando.