Cifrado XOR: Cuarentena y One-time pad

Objetivos

- Que el alumno conozca formas de uso para el cifrado XOR, utilizado tanto por herramientas de seguridad como por programas maliciosos, particularmente una de las formas en que los antivirus procesan los archivos sospechosos que son puestos en Cuarentena.
- Que el alumno se familiarice con los ataques de fuerza bruta a contraseñas y el uso de un editor hexadecimal, comúnmente utilizado en el análisis forense, análisis de malware y análisis de tráfico de red.

Requisitos

- Editor hexadecimal
- Descargar este malware.

PARTE 1: Malware en cuarentena

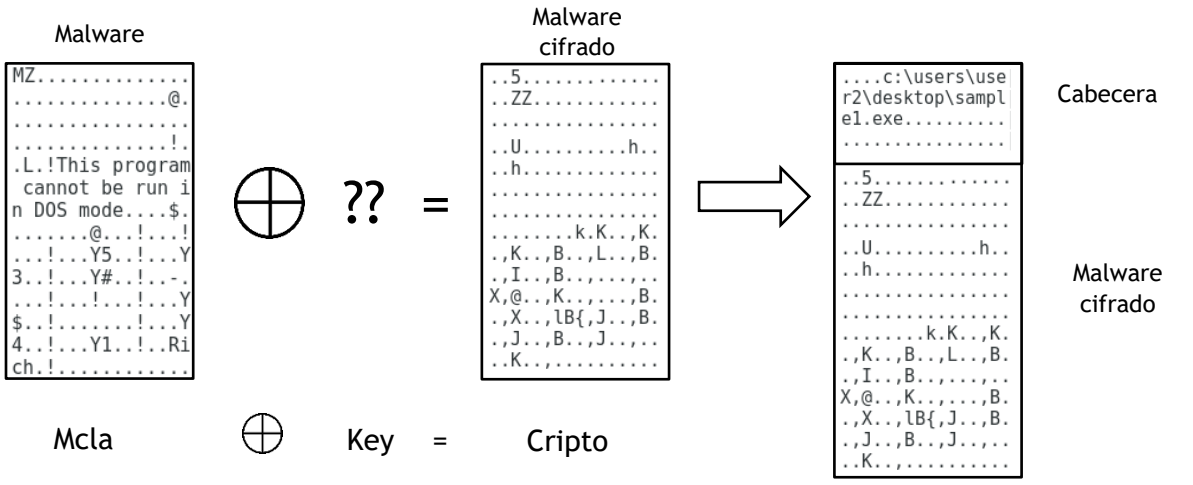
Introducción

La Cuarentena es una ubicación especial en la que los antivirus colocan archivos identificados como malware.

Comúnmente, los antivirus realizan un procedimiento sobre los archivos infectados que se moverán a Cuarentena, esto con el fin de evitar su operación. Dicho procedimiento puede ser, por ejemplo, comprimir el archivo sospechoso en formato zip, añadiendo una contraseña. Otra técnica común es cifrar el archivo (utilizando XOR con una clave de uno o más bytes), para después agregar datos de referencia al inicio del archivo. Algunos de los formatos utilizados son:

- ESET (NQF)
- Kaspersky (KLQ)
- MalwareBytes Data files (DATA)
- MalwareBytes Quarantine files (QUAR)
- McAfee Quarantine files (BUP)
- Symantec Quarantine Data files (QBD)
- Symantec Quarantine files (VBN)

Para esta práctica se usará el archivo `57FD6325.VBN`, el cual corresponde a un malware puesto en cuarentena por una solución antivirus. El antivirus cifró el malware aplicando una función XOR con una llave de **un byte**, y después agregó una cabecera al malware cifrado.



Desarrollo

- Programe una función que obtenga, a través de fuerza bruta, la llave con la que se cifró el archivo. Recuerde que el malware es un archivo binario que se puede ejecutar en Windows, es decir, contiene cadenas comunes a los archivos ejecutables[1]. También tenga en cuenta las propiedades de la operación XOR.

- Programe una función que reciba la llave obtenida en el paso anterior y descifre el archivo, guardándolo en un archivo diferente.

- El programa completo recibirá al menos dos argumentos desde la línea de comandos: el archivo a descifrar y el nombre del archivo en donde se guardará ya descifrado.

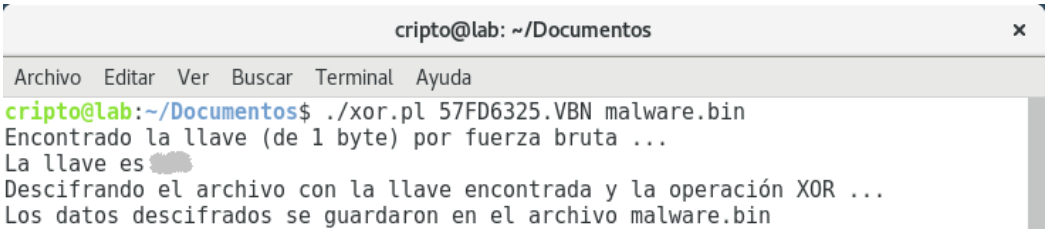


Fig. 1 Ejemplo de la ejecución del programa

[1] Dichas cadenas se conocen como “números mágicos” (*file signatures* o *magic numbers* en inglés).

- Con ayuda de un editor hexadecimal, elimine del archivo descifrado los datos de referencia agregados por el antivirus (cabecera). El tamaño del contenido a eliminar será delimitado por la ubicación en el archivo del *número mágico* de los archivos ejecutables.

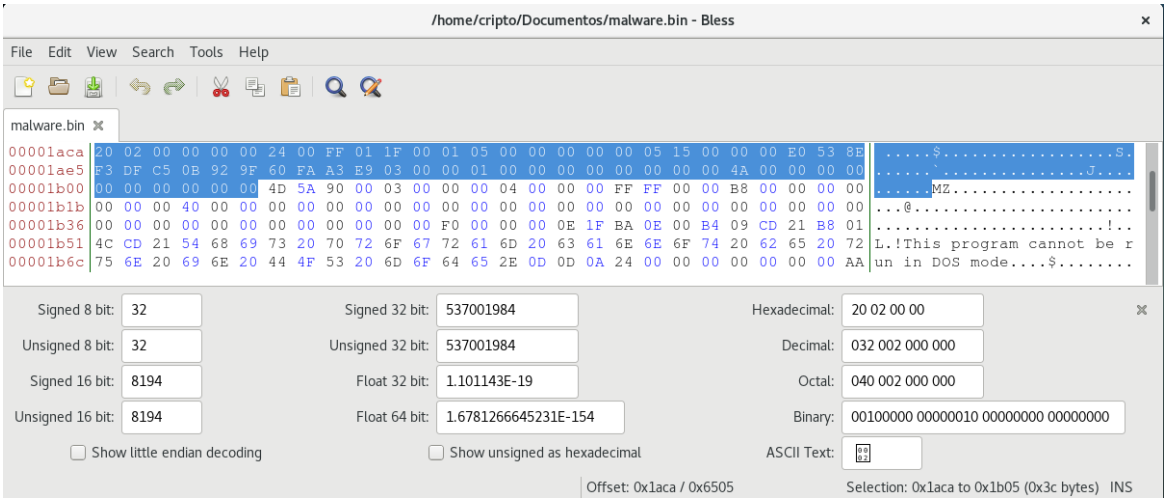


Fig. 2 Borrar cabecera con ayuda del editor hexadecimal

¡¡¡Advertencia!!!

Este archivo no debe ser ejecutado en sistemas operativos Windows,
ya que puede infectarse. Se recomienda hacer todo el procedimiento con Linux.

- Una vez guardado el archivo del malware, subirlo al sitio de análisis Virus Total https://www.virustotal.com

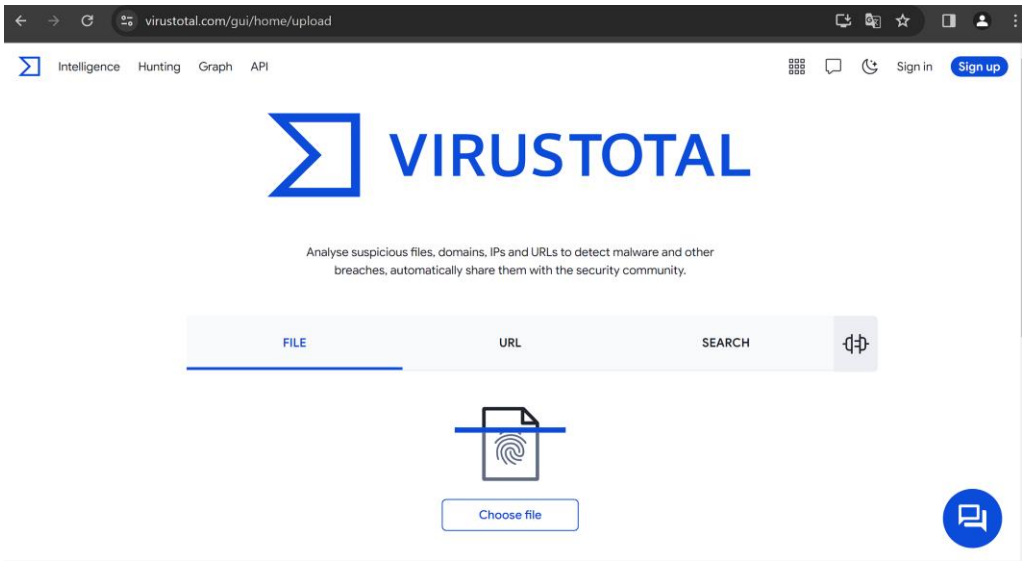


Fig. 3 Interfaz para subir malware a VirusTotal

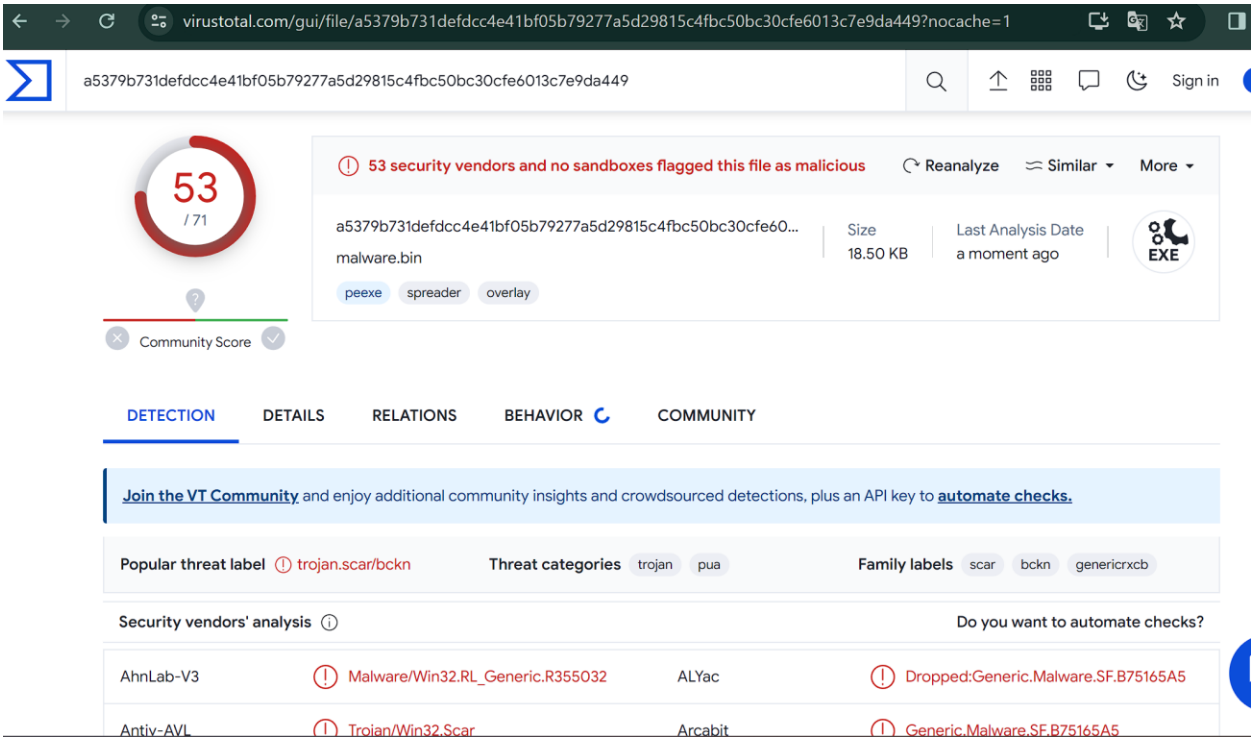


Fig. 4 Ejemplo de resultados del análisis

PARTE 2: One-time pad

Introducción

El *One-time pad* (OTP) o "Libreta de un solo uso" es una técnica de cifrado en la que, hablando de la llave y su uso, se debe cumplir con lo siguiente:

- Debe ser aleatoria
- Debe ser de la misma longitud que el mensaje que se quiere cifrar
- Debe emplearse **una sola vez**

Esta técnica de cifrado es considerada *irrompible*[2], siempre y cuando se cumplan las características mencionadas anteriormente y se mantenga secreta la llave de cifrado. No obstante, cumplir con dichas características vuelve al OTP impráctico para cifrar un elevado número de mensajes.

¿Se puede modificar esta técnica de tal manera que sea más práctica? Por ejemplo, ¿qué pasa si generamos una sola llave cuya longitud sea igual o mayor a la del mensaje más grande que necesitemos cifrar?

Desarrollo

A continuación, se lista una serie de mensajes cifrados con una variante del OTP que podríamos nombrar "Many-time pad", ya que todos los mensajes fueron cifrados utilizando la misma llave:

- `72212b29402b66342437255717562d652e5748692f2452206b17153b36364b2236172f2b372b2a2c2c6f2d6e3e3347`
- `5a74202b412f243123743459175c3737394b1224382f5d332244747436364b35365b3426276a3d762e2e206e273f46312a30203b`
- `752122394f23233e39316d555e412c653949472c212957326745372726215725791 70b2b742e21762720226e192f582726263527`
- `403b283d412722316d356d4d5952632d2d555b25212456352217313a21235d35345e21296e6a2837633c29203e2854653a74323d23`
- `7e3b213d463a277039213b5717423620784b473a3d2056252245782727734c3725522f67302f643b2c23293c7137542c39`
- `5e353731512166316d253851525d63373d4b422623215d336b17303d38324b7636172226276a3523266f282b333515212226`

[2] Shannon, C.: Communication Theory of Secrecy Systems. Bell System Technical Journal 28 (1949) 656–715

- 62212c2b5a6e212224202c4a174a63362d18553b2431576121423d74213c5e3934562a28743a2b246320383c307a572a2035

- 63313737153a233e292624591b1337243418442c37654d2f675f313e3d735c3377553b223a2b642522212b3c34

- 7b353c78522b282428743c4d52132d2a784c5b2c232018243443373933345776275 63c26742f3722266f232838395c2a

- 603d65365a6e35356d38281854462e35345153276d364d3267543924203a5b3e38446e65382f6432222d2d6e323254332620206a

- 7274353d462f347029316d5d44472c3678515c2a24215d2f33522b743e3c4b76395e2028276a212422216c273f2950352226202a3c5040

- 13382a2b152d2e393f26245c584063213d1842263f31572f22447825273618253217 2f25262f2a7a632a206e3b3b51202c

- 62212078533c2f313e743e5759132f242b185f282324562034173d3a7210512333562a67062f253a ← Descifrar este mensaje

Los criptogramas fueron generados utilizando un programa que puede descargar de aquí.

El objetivo es descifrar el último criptograma de la lista, por lo que debe realizar un programa que le ayude a llevar a cabo esta tarea.

Dado que se empleó la misma llave para cifrar todos los mensajes, comience aplicando XOR del último criptograma con el resto y tome en cuenta qué ocurre cuando se hace XOR de una letra con un espacio (p.ej. ‘a’ ⊕ ‘ ‘ = ‘A’).

Al descifrar el último criptograma y obtener el mensaje original, se vuelve trivial obtener parte de la llave utilizada para cifrar el resto de los mensajes. Emplee dicho fragmento de la llave para descifrar el resto de criptogramas.

Puede realizar el programa en el lenguaje de su elección.

Cuestionario

1. ¿Para qué se usa la herramienta XORsearch? https://blog.didierstevens.com/programs/xorsearch/
2. ¿De cuántos bytes es la cabecera que le agregó el antivirus al malware?
3. ¿Qué son los números mágicos? (relacionado con archivos)
4. ¿Qué es VirusTotal?
5. De acuerdo a VirusTotal, ¿qué tipo de malware es?
6. En la parte 2 de la práctica, ¿a qué obra y a qué autora pertenecen los textos que logró descifrar?

Elementos a calificar

1. Redacte un reporte en el que indique los pasos que considere necesarios para explicar cómo realizó la práctica, incluyendo capturas de pantalla que justifiquen su trabajo.
2. Incluya en su reporte tanto las respuestas del Cuestionario, como un apartado de conclusiones referentes al trabajo realizado.
3. Puede agregar posibles errores, complicaciones, opiniones, críticas de la práctica o del laboratorio, o cualquier comentario relativo a la misma.
4. Deberá subir el reporte en PDF a Classroom y el código a un repositorio de GitLab que sea privado, mismo que deberá ser compartido con el profesor y los ayudantes, además de colocar el enlace en el reporte.

Referencias

- Didier Stevens. *XORSearch & XORStrings*. https://blog.didierstevens.com/programs/xorsearch/
- McAfee. *How to restore a quarantined file not listed in the VSE Quarantine Manager*. https://kc.mcafee.com/corporate/index?page=content&id=KB72755
- Hexacom. *DeXRAY*. http://www.hexacorn.com/blog/2016/03/11/dexray/

Universidad Nacional Autónoma de México

Paulo Contreras Flores
paulo.contreras.flores@ciencias.unam.mx

Jonathan Banfi Vázquez
jbanfi@ciencias.unam.mx

Tonatihu Sánchez Neri
tonatihus@ciencias.unam.mx