Jugando con antivirus.

Enrique Casale Linde

12 de abril de 2018

Índice

1. Presentación

Hola, soy Enrique. Voy a hablaros un poco sobre el mundo de los antivirus, cómo funcionan y qué hace el malware para evitarlos.

2. ¿Qué es y cómo funciona un antivirus?

2.1. Antivirus

Un antivirus es un software que se encarga de detectar malware. El malware es todo software creado con un objetivo malicioso. Existen muchos tipos de malware, pero no hablaremos específicamente de ninguno de ellos en esta charla. Simplemente, veremos las técnicas que utilizan para lograr su objetivo sin ser detectados por el malware.

Tipos de malware:

- Adware
- Ramsoware
- Rootkit
- Spyware
- Torjan horse
- Virus
- Worms

2.2. Pasado

Antes de iniciar la investigación para esta charla, tenia la certeza, de que el malware había ido avanzando desde principios del 2000 hasta nuestro estado actual. Tras investigar un poco sobre los métodos que se usan para hacer que el malware no sea reconocido por un antivirus pensé que el mundo del malware se había revolucionado desde aquellos años. Resulta que estaba muy muy equivocado. Como ya veremos mas adelante, la técnica mas avanzada que usa el malware para evitar a los antivirus, se usó en el año 2002. En aquella época, los antivirus se basaban plenamente en un sistema de firmas. Las firmas, eran hash del malware en cuestión. Para comprobar si un archivo es o contiene un malware, se comprueba el hash del archivo con la base de datos. Si se produce una coincidencia, enhorabuena, tienes un virus.

2.3. Presente

Actualmente el malware ha crecido excepcionalmente en cantidad. Seguro que también habrá crecido en complejidad, pero la mayoría de técnicas usadas por el malware actual para ocultarse de los antivirus ya habían sido usadas en la década del 2000. Conforme el malware fue creciendo, los antivirus tuvieron que mejorar su capacidad para detectar el malware. Con el tiempo se dieron cuenta de que el sistema de firmas estaba obsoleto. Era imposible detectar una

amenaza a no ser que estuviera registrada en la base de datos del antivirus. Por este motivo se implementó el análisis heurístico y la simulación o sandbox. El análisis heurístico tiene como fin encontrar comportamientos sospechosos dentro de un programa: Ganar privilegios, intentar borrar o encriptar ciertas carpetas, abrir un puerto... La principal ventaja de este tipo de análisis es que puede encontrar malware antes de que sea activado y de que se encuentre en la base de datos de firmas, pero, también presenta un inconveniente. Los falsos positivos. Al permitir al software reconocer por el mismo las amenazas te arriesgas a que cometa fallos. Estos fallos son los falsos positivos. A veces el antivirus reconoce un archivo con un malware aunque no lo sea. La simulación o Sandbox, consiste en la simulación de la ejecución del programa en un entorno controlado antes de ser ejecutado en el sistema. Mientras se ejecuta el archivo, el antivirus analiza su comportamiento e interpreta los resultados. La principal desventaja de este tipo de análisis es el coste de eficiencia que conlleva ejecutarlos.

2.4. Futuro

Los antivirus se han vuelto sofisticados, pero no nos dejemos engañar. Aun hay malware que el antivirus no es capaz de reconocer. Esto se debe a que las técnicas heurística y de simulación no funcionan perfectamente y pueden ser evitadas por el malware. Actualmente se está investigando las aplicaciones del aprendizaje automático a el reconocimiento de amenazas. Una de las formas de utilizar el aprendizaje automático para el reconocimiento de amenazas, es el entrenar un algoritmo con el trafico de una red. De esta forma, si el programa detecta alguna anomalía puede avisar a un administrador / encargado o aislarla directamente.

3. El antivirus libre, clamAV. ¿Mejores antivirus?

3.1. ClamAV

3.1.1. ¿Que es ClamAV?

ClamAV es un anti-virus abierto, licenciado bajo GPL. Principalmente, ClamAV se usa como antivirus en los servidores de correo electrónico. Aun así, ClamAV tiene un escáner que se puede usar por linea de comandos, de forma que puede ser usado como antivirus. ClamAV no ofrece protección en tiempo real, es decir, solo actuá cuando es invocado por algún script o por el propio usuario. Ademas, como veremos ahora, ClamAV permite crear nuestras propias bases de datos.

3.1.2. Creando nuestra propia base de datos para ClamAV

Cada base de datos oficial de ClamAV comienza con una cabecera de 512 bytes que sigue el mismo esquema:

```
ClamAV-VDB:build time:version:number of signatures:functionality
level required:MD5 checksum:digital signature:builder name:build
time (sec)
```

Por defecto, ClamAV usa las bases de datos que se encuentren en su ruta local */usr/local/share/clamav*. Al crear nuestra propia base de datos, tenemos dos opciones para usarla. La primera, guardarla en la ruta por defecto de clamAV y la segunda usar la opción -d seguida de la ruta a nuestra base de datos. También podemos especificarle al escáner que solamente utilice las bases de datos oficiales.

ClamAV soporta varios algoritmos para crear firmas. Cada una se guarda en un fichero diferente:

- MD5. Sus bases de datos se guardan en archivos *.hdb*.
- SHA1. Sus bases de datos se guardan en ficheros *.hsb*.
- SHA256. Sus bases de datos se guardan en ficheros *.hsb* al igual que en con el algoritmo SHA1.
- PE sections. Sus bases de datos se guardan en archivos *.mdb* o *.msb*.

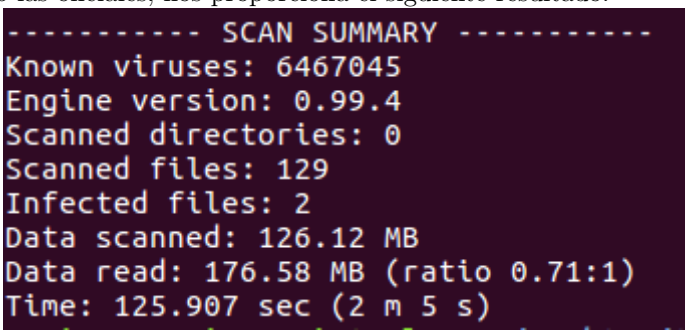
Hora de probar los que hemos aprendido: ClamAV no detecta el malware en un .zip si este esta protegido con contraseña. Esto se debe a que al comprimir de esta forma, cada contraseña genera un archivo .zip diferente. Como se puede observar en la foto, ClamAV detecta solamente el malware comprimido sin contraseña.

```
enrique@enrique-VirtualBox:~/Desktop/theZoo-master/malwares/Binaries/Dino$ ls
dino.bin  Dino.md5  Dino.pass  Dino.sha256  dino-sin-pass.zip  Dino.zip
enrique@enrique-VirtualBox:~/Desktop/theZoo-master/malwares/Binaries/Dino$ clamscan dino-sin-pass.zip
dino-sin-pass.zip: Win.Trojan.Dino-1 FOUND

----------- SCAN SUMMARY -----------
Known viruses: 6467045
Engine version: 0.99.4
Scanned directories: 0
Scanned files: 1
Infected files: 1
Data scanned: 0.48 MB
Data read: 0.24 MB (ratio 1.98:1)
Time: 12.828 sec (0 m 12 s)
enrique@enrique-VirtualBox:~/Desktop/theZoo-master/malwares/Binaries/Dino$ clamscan Dino.zip
Dino.zip: OK

----------- SCAN SUMMARY -----------
Known viruses: 6467045
Engine version: 0.99.4
Scanned directories: 0
Scanned files: 1
Infected files: 0
Data scanned: 0.24 MB
Data read: 0.24 MB (ratio 1.00:1)
Time: 14.165 sec (0 m 14 s)
enrique@enrique-VirtualBox:~/Desktop/theZoo-master/malwares/Binaries/Dino$
```

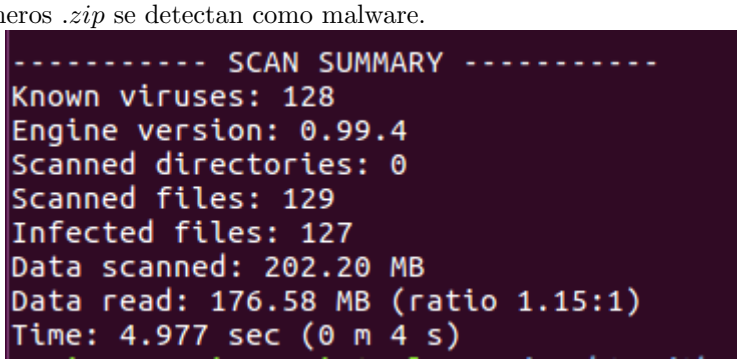
Comencemos a crear nuestra base de datos. Simplemente, creamos un archivo *.hdb*. En mi caso lo llamare destructor-de-mundos.hdb. Primero voy a aclarar el entorno en el que estamos. Nos encontramos en un directorio que contiene un directorio por cada malware. Los malware, están comprimidos en un *.zip* con contraseña. Para crear la base de datos he usado el siguiente script:

```
#!/bin/bash



for i in $(ls */*.zip);
do
sigtool --md5 $i >> destructor-de-mundos.hdb
done
```

El análisis de los ficheros .zip sin usar nuestra base de datos personalizada, solo las oficiales, nos proporciona el siguiente resultado:



Y con destructor-de-mundos obtenemos.hdb el resultado esperado. Todos los ficheros *.zip* se detectan como malware.



Si queremos usar los algoritmos SHA1 o SHA256 para generar la base de datos, solo hay que seguir el mismo procedimiento, cambiando el archivo por uno *.hsb* y utilizando en sigtool el flag –sha1 o –sha256

En el caso de querer crear una base de datos basada en secciones de código, la forma mas fácil de hacerlo es copiando la sección de código a un fichero vació y creando la firma a partir de dicho fichero.

3.2. ¿Mejor antivirus?

La tabla habla por si sola. Estos resultados fueron obtenidos por av-comparatives, una empresa que se dedica a realizar comparativas de diferentes antivirus con una gran base de datos de malware.

Manufacturer	Product	Detection Rate Windows Malware	Detection Rate Linux Malware
ESET	NOD32 Antivirus for Linux Desktop 4.0.81.0	99.8%	99.7%
Kaspersky	Anti-Virus for Linux File Server 8.0.3.265	99.8%	98.8%
AVG	Server Edition for Linux 2013.3118	99.3%	99.0%
Avast	File Server Security 1.2.1	99.7%	98.3%
Symantec	Endpoint Protection Manager 12.1.6 Build 6168	100.0%	97.2%
Kaspersky	Endpoint Security for Linux 8.0.1.50	96.3%	100.0%
Sophos	Antivirus for Linux 9.9.0	99.8%	95.0%
F-Secure	Linux Security 10.20 Build 358	99.9%	85.7%
Bitdefender	Antivirus Scanner for Unices 7.141118	99.8%	85.7%
Microworld	eScan Antivirus for Linux Desktop 7.0-18	99.8%	85.7%
G Data	Client Security Business for Linux 13.2.251	99.8%	81.2%
Dr. Web	Antivirus for Linux 10.1	67.8%	91.6%
McAfee	VirusScan Enterprise for Linux 2.0.1.29052	85.1%	41.9%
Comodo	Antivirus for Linux 1.1.268025.1	83.0%	33.1%
ClamAV	ClamAV 0.98.7	15.3%	66.1%
F-Prot	F-Prot Antivirus for Linux 6.2.39	22.1%	23.0%

Note: Test under Ubuntu Desktop 12.04 LTS 64 bit; AV-Test 09/2015

4. Técnicas para burlarte de los antivirus

4.1. Compresión

Aunque parezca sencillo, al comprimir un archivo su Firma cambia lo que hace que el antivirus no detecte el archivo comprimido como una amenaza.

4.2. Automodificacion

Como hemos hablado anteriormente, los antivirus funcionan con Firmas. Normalmente los antivirus utilizan fragmentos de código para identificar al malware, son estos fragmentos de código los que el malware cambia cada vez que se duplica. En el caso de los antivirus basados unicamente en firmas, este tipo de malware es imposible de detectar, ya que su firma cambia cada vez que infecta un nuevo archivo.

4.3. Autoencriptacion

En este caso, el virus encripta la parte detectable o delatadora del mismo, esperando hasta que el antivirus se desactive, ya sea por parte del usuario o de una actualización. En ese momento, el malware se desencripta. Actualmente, los antivirus reconocen grandes secciones de código encriptadas como posible

malware. Para evadir esta medida, actualmente se encripta los virus en un archivo ejecutable, de forma que el antivirus no puede reconocer que está encriptado.

4.4. Basureros

Este tipo de de malware hace uso de segmentos de código basura que en realidad no tiene ninguna utilidad. Los inserta por todo el código con la intención de que el antivirus ignore la parte peligrosa del código. Debido a esto se cambia la firma del malware con cada nueva infección.

4.5. Malware polimorficos

Aquí es donde la gente se empezó a volver loca. Alguien un día se levantó por la mañana, desayunó, se lavó los dientes y pensó: Hostias, pues si los antivirus detectan la autoencriptacion, la modificación y la inyección de código basura, ¿Por qué no hago un código que cree un código nuevo con cada nueva infección o incluso cada x tiempo? Y lo hizo. Y funcionó. La única forma de parar este tipo de malware es obtener la parte del código que nunca cambia e introducirla en la base de datos de firmas. Esto ya requiere una inversión de tiempo importante, el cual el malware utilizara para propagarse.

4.6. Malware metamórficos(Polimorficos dopados)

Son la evolución lógica de los malware polimorficos. Este tipo de malware reescribe todo su código. Esto los hace imposibles de identificar por el sistema de firmas de cualquier antivirus, por lo que se deben recurrir unicamente a métodos de simulación(sandbox) y al análisis heurístico para descubrirlos.

Un ejemplo es el W31/Simile, un virus detectado en 2002 escrito por completo en ensamblador . El virus en si era inofensivo, sólo buscaba la forma de propagarse y en determinadas fechas mostraba diferentes tipo de mensajes, pero cada vez que se propagaba cambia su código por completo. Simile tiene cerca de 14000 lineas de código de las cuales el 90 por ciento es parte del motor de creación del virus.

5. Conclusion

La protección que ofrecen los antivirus contra los nuevos tipos de malware es insuficiente. Las técnicas que se utilizan hoy en día no son lo suficientemente sofisticadas como para proteger un sistema. Aun así, los antivirus pueden resultar útiles en la detección de malware ya conocidos o nuevos que no sean lo suficientemente avanzados. Quizás en el futuro con la entrada del aprendizaje automático en la seguridad informática se pueda obtener software contra malware mas eficaz.

6. Bibliografía

Cronología de los malware

`https://www.lifewire.com/brief-history-of-malware-153616`

Repositorio de malware utilizado

https://github.com/ytisf/theZoo

Como el malware evita la virtualizacion

https://www.first.org/resources/papers/conf2017/Countering-Innovative-Sandbox-Evasion-T
pdf

Otros enlaces

http://www.toptenreviews.com/software/articles/what-is-heuristic-antivirus-detection/
https://www.opswat.com/blog/understanding-heuristic-based-scanning-vs-sandboxing
https://en.wikipedia.org/wiki/Heap_spraying
https://www.securityweek.com/clever-techniques-help-malware-evade-av-engines
https://www.bankvaultonline.com/news/security-news/viruses-how-they-hide-from-antivirus
https://en.wikipedia.org/wiki/Simile_(computer_virus)
https://www.bankvaultonline.com/knowledge-base/explainers/viruses-metamorphic-code
https://www.malwarefox.com/malware-types/
https://www.av-comparatives.org/linux-reviews/