

MÁSTER EN ANÁLISIS DE MALWARE Y REVERSING
MÓDULO 2. ENTORNOS DE ANÁLISIS DE MALWARE

MÁSTER EN *ANÁLISIS DE MALWARE Y REVERSING*



Campus Internacional
CIBERSEGURIDAD

ENIIT
INNOVA IT BUSINESS SCHOOL



UCAM
UNIVERSIDAD
CATÓLICA DE MURCIA

Tabla de contenido

1	El Malware.....	3
1.1	¿Qué es el malware?	3
1.1.1	El significado de la palabra malware	4
1.2	Primeros virus mediáticos	4
1.3	Coincidencias entre los virus biológicos y los informáticos	5
1.4	El genuino concepto de virus informático	6
1.5	¿Cómo se desarrolla el malware?.....	7
1.5.1	Reutilización de código y componentes	7
1.5.2	Frameworks de explotación.....	8
1.6	Soportes que pueden ser usados por el malware	8
1.6.1	Archivos ejecutables	9
1.6.2	Archivos multimedia	10
1.6.3	Páginas web y extensiones del navegador	10
1.6.4	Documentos ofimáticos.....	11
1.6.5	Documentos PDF	13
1.7	Vectores de entrada del malware	14
1.7.1	Correo electrónico.....	15
1.7.2	Dispositivos USB	15
1.7.3	Exploits.....	16
1.7.4	Navegación web.....	17
1.8	Ciclo de vida de un malware	17
1.8.1	Fase de infección	18
1.8.1.1	Downloader	19
1.8.1.2	Dropper	19
1.8.1.3	Fileless	20
1.8.2	Fase de persistencia.....	21
1.8.2.1	Menú de inicio	22
1.8.2.2	Registro de Windows	22
1.8.2.3	Tareas programadas.....	23
1.8.2.4	Servicios del sistema.....	24
1.8.3	Fase de comunicación.....	25
1.8.3.1	Canales de aplicaciones y protocolos de comunicación.....	26
1.8.3.2	Canales usando aplicaciones web	26
1.8.3.3	Canales usando redes sociales	27
1.8.3.4	Canales usando servidores.....	27
1.8.4	Fase de desencadenamiento	28
1.8.4.1	Criptominado.....	28
1.8.4.2	Adware.....	29
1.8.4.3	Destrucción de infraestructura	29
1.8.4.4	Gusanos.....	29
1.8.4.5	Ransomware	30
1.8.4.6	Botnets	30
1.8.4.7	Robo de información	31

1.9	Elevación de privilegios	31
1.10	La importancia de la ingeniería social en la difusión del malware	32

1. EL MALWARE

Vamos a realizar un recorrido necesario por el malware. Desde su definición más o menos formal a su comportamiento y una clasificación que, hablando de malware, siempre se queda corta puesto que la eterna carrera entre “gato y ratón” produce nuevas vías para que quienes crean este tipo de artefactos puedan evadir las defensas, aunque sea por un intervalo suficiente de tiempo para seguir lucrándose.

1.1 ¿Qué es el malware?

Si nos hacemos esa pregunta todos creemos saber la respuesta de una manera más o menos intuitiva, pero curiosamente, al intentar dar una definición exacta, posiblemente cada uno de nosotros daría una diferente. Unos dirían que el malware es un “virus”, mientras que otros dirían que es un “troyano”, etc. No importa, todos estarían en lo cierto en cuanto a la naturaleza, aunque no precisemos la respuesta con cierta formalidad.

Vamos a intentar dar aquí una definición de una vez por todas.

El malware es todo aquello que puede ser ejecutado por un sistema y cuyo resultado es perjudicial para los intereses de un usuario u organización.

¿Vamos bien?

El malware no es más que un programa informático más. Da igual el lenguaje de programación o formato en el que se presente (un binario, un script...). Siguen siendo instrucciones que ejecutadas por un procesador y acatadas por un sistema operativo afectan a este modificando su estado en uno que perjudica, como hemos comentado, los intereses de los usuarios y organizaciones.

Así pues, consideramos malware todo aquello que posee una acción o acciones maliciosas en el contexto de los sistemas informáticos.

Nota curiosa: Linus Torvalds, el conocido creador de Linux arregló un bug en el kernel debido a que se descubrió un error afortunado que hacía que un virus que afectaba a sistemas Linux no funcionase. No obstante, la prensa generalista no entendió las razones completamente bien intencionadas de dicho parche.

Home > Security

NEWS

Torvalds patches Linux kernel, fixes broken virus



By Robert McMillan

IDG News Service | APR 19, 2006 5:02 PM PST

The hacker who created a widely reported cross-platform virus that could affect both Windows and Linux PCs may have inadvertently done some free bug testing for the Linux operating system. On Wednesday Linux creator Linus Torvalds said he had patched his operating system kernel to fix a bug that had been preventing the virus from running.

The virus, called Virus.Linux.Bi.a/ Virus.Win32.Bi.a, was first reported by security vendor Kaspersky Lab Ltd. on April 7, which labeled it an interesting proof of concept program, because of its ability to affect both Windows and Linux. <http://www.viruslist.com/en/weblog?weblogid=183651915>

IMAGEN 1 NOTICIA EN LA QUE SE SEÑALA QUE TORVALDS ARREGLA UN BUG GRACIAS A UN VIRUS

1.1.1 El significado de la palabra malware

“Malicious software”. La palabra malware proviene de la conjunción de los términos **malicious** (malicioso) y **software**.

- El malware es un programa informático. Creado de la misma forma que las aplicaciones que usas a diario en tu ordenador o dispositivo.
- La principal diferencia entre un programa, “normal” y el malware es que este último se ha programado explícitamente para que tenga un **comportamiento malicioso**.

1.2 Primeros virus mediáticos

Los primeros virus informáticos fueron pruebas de laboratorio para demostrar que un programa podía replicarse a sí mismo. No tenían ninguna intencionalidad maliciosa. En los años 80 se crearon diversos virus informáticos, aunque en la mayoría de los casos su comportamiento era más una molestia que un peligro real.

El primer virus informático cuyo comportamiento llevó a su autor a juicio fue el conocido como [Gusano Morris](#), creado por el entonces estudiante estadounidense Robert Morris en 1988.

Ya por aquel entonces, en una Internet primigenia de tan solo 60.000 nodos, el gusano Morris consiguió infectar al 10% de ellos.

Robert hace hincapié en que no tuvo intención alguna en liberar el virus y que fue un acto accidental. Morris también aclara que el virus contenía un fallo: no debía replicarse de forma continua (haciendo que la máquina dejara de responder) sino una sola vez por máquina afectada, lo que hubiera supuesto una infección sin complicaciones.

1.3 Coincidencias entre los virus biológicos y los informáticos

Existe un gran paralelismo entre los virus que padecemos los seres vivos y los virus informáticos. La palabra 'virus', proviene del latín y significa 'veneno'.

- Los virus naturales se replican continuamente cuando infectan un ser vivo
- Los virus informáticos se reproducen creando copias de si mismo
- Los virus naturales infectan a otro huésped a través de un medio físico: el aire, contacto físico, etc.
- Los virus informáticos infectan a través de varios vectores: correo electrónico, archivos infectados, conexiones de red, etc.
- Los virus naturales pueden ser potencialmente mortales (Ébola) o crear molestias pasajeras (Resfriado común)
- Los virus informáticos pueden acabar completamente con un sistema informático o ser una simple molestia fácil de erradicar
- Los virus naturales se combaten con higiene, medidas profilácticas, antivirales y vacunas.
- Los virus informáticos se combaten con concienciación en el adecuado uso de los sistemas, antivirus y parches específicos de los fabricantes de sistemas.

1.4 El genuino concepto de virus informático

En los 90, cuando surgieron los primeros virus que llegaban a ser incluso noticia en medios generalistas, no se tenía una idea clara de “qué hacer con ellos”. Incluso una gran parte de ellos no hacían nada grave, eran una molestia pasajera que mostraba un mensaje por pantalla cada cierto tiempo.

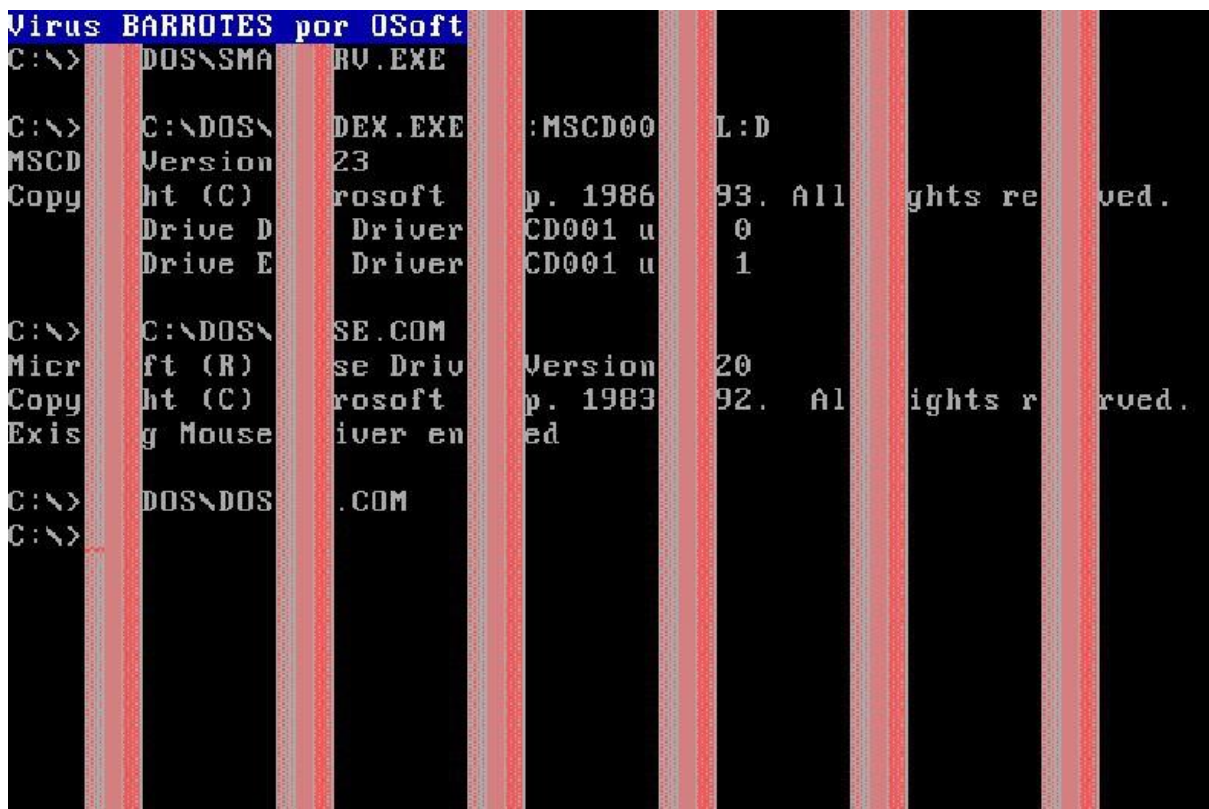


IMAGEN 2 EL INFAME VIRUS BARROTES

Fue una época “romántica” para el malware (ni tan siquiera se le llamaba popularmente malware aun) donde grupos de creadores de virus competían entre sí por obtener nuevas técnicas de replicación, infección y vectores innovadores, además de mecanismos para evadir las defensas que ya iba alumbrando el mercado.

Muchos de estos grupos ni tan siquiera tenía como objetivo la infección destructiva, sino que, como se ha dicho, usaban esta capacidad para “picarse” entre ellos.

Al margen de ellos, se fueron desarrollando los primeros grupos criminales que aprovechando dicho conocimiento (que era compartido en listas y e-zines del momento) sí vieron un cariz lucrativo y dieron luz a una incipiente economía de mercado negro basada en, ahora sí, el malware.

El termino virus se popularizo precisamente por las coincidencias con sus

homólogos biológicos. Prácticamente la primera las técnicas popularizadas consistía en “inocular” el código malicioso del virus informático en archivos “sanos” para posteriormente desde dicho binario ir saltando a otros archivos y sistemas (inicialmente, debido al “contagio” por el intercambio de discos flexibles entre usuarios).

1.5 ¿Cómo se desarrolla el malware?

Curiosamente, el malware se crea con las mismas herramientas y lenguajes de programación que el resto de los programas informáticos. Lo que cuenta para definir a un programa como malicioso es la capacidad de este para realizar acciones contra los intereses del usuario o sistema afectado.

Un **lenguaje de programación** es la forma en la que las personas indicamos a los ordenadores cómo y qué tienen que hacer. Estos lenguajes suman dos potentes conceptos en computación: los **algoritmos** y las **estructuras de datos**.

El malware puede ser escrito en prácticamente cualquier lenguaje de programación. Existe malware creado en Javascript, Visual Basic Script, Bash, Python, Ensamblador, etc. Lo importante y necesario para el malware es que este se ejecute y desencadene su contenido malicioso.

A los creadores de malware no les interesa el lenguaje de programación, para ellos simplemente es un medio para que se ejecuten sus acciones.

1.5.1 Reutilización de código y componentes

Sin embargo, si les interesan determinadas técnicas de programación...

Del mismo modo que la Ingeniería del Software estudia la reutilización de código y componentes de software, los creadores de malware copian funcionalidad de otros malware o directamente modifican uno ya existente.

Un ejemplo: Mirai es un malware que infecta dispositivos IoT mal configurados. Su código fuente fue liberado y a partir de él se han detectado cientos de variantes con nuevas funcionalidades.

<https://github.com/jgamblin/Mirai-Source-Code>

1.5.2 Frameworks de explotación

Conscientes del aumento de productividad, existen frameworks que aglutinan exploits del mismo modo que existen librerías con funciones en lenguajes de programación.

Los frameworks de explotación permiten utilizar su catálogo de exploits para emplearlos contra un objetivo determinado e incluso de forma genérica.

Los exploits de estos frameworks no pasan desapercibidos para los creadores de malware y son usados por estos en determinadas ocasiones.

Podría pensarse que los frameworks de explotación son ilegales, pero no lo son. Estos también se usan a diario por los equipos de seguridad de empresas para determinar el nivel de explotabilidad de los recursos protegidos o analizados.

Existen decenas de este tipo de frameworks. El más popular de todos, **Metasploit**, posee cientos de exploits listos para ser usados, además de herramientas para la evasión de antivirus, escáner de red, etc. <https://www.metasploit.com/>

Empire es un proyecto que contiene un amplio catálogo de técnicas de evasión y explotación de sistemas Microsoft Windows. Está escrito en **PowerShell**, un lenguaje usado en el intérprete de comandos del mismo nombre. <https://www.powershell empire.com/>

Los creadores de malware usan estos dos y otros frameworks para acelerar el proceso de creación de un malware. No obstante, al ser un contenido muy popular, los antivirus poseen una gran capacidad para detectarlo y reaccionar a tiempo. Por ello, la reutilización en este caso ha de ser sopesada por los cibercriminales, ya que corren el riesgo de que su malware posea una corta vida.

1.6 Soportes que pueden ser usados por el malware

El malware no existe en su forma aislada, necesita de un soporte o huésped que lo transporte o le sirva de plataforma sobre la que actuar.

En el caso de los virus informáticos estos soportes son, en la mayoría de los casos, los archivos.

Existen multitud de tipos de archivos, pero estos se dividen en dos tipos fundamentales: binarios y texto. Los primeros no son legibles directamente por un ser humano mientras que los archivos de texto lo son (¡aunque no entendamos lo que vemos!)

Vamos a ver la siguiente jerarquía de archivos que podrían alojar malware:



1.6.1 Archivos ejecutables

Un archivo ejecutable es una “aplicación”; típicamente, aquella que puedes ejecutar haciendo “doble clic” sobre su icono.

También habrás visto su característica extensión en sistemas Windows, esta es “.exe”

Son conocidos como archivos binarios porque si lo abrimos no veremos texto sino una secuencia de símbolos que aparentan no tener sentido alguno.

Históricamente, los primeros virus afectaban casi exclusivamente a archivos binarios o ejecutables.

En sistemas Windows también era de uso la extensión “.com”, herencia del antiguo sistema MS-DOS.

Es la forma tradicional del malware y también la que mayor número de muestras genera a lo largo del año.

Hasta hace relativamente poco, el malware en forma de ejecutable incluso era

adjuntado en correos electrónicos. Ya quedan pocos proveedores que permitan este tipo de acciones por considerarlas altamente sospechosas.

Lo habitual es que los ejecutables sean alojados en algún servidor o servicio en nube de alojamiento de archivos y se genere un enlace para que, mediante otra técnica de explotación, sean descargados.

Es decir, el malware de esta forma espera a que sea otro tipo de técnica la que le abra la puerta al sistema. Por ejemplo, visitar una web activa la explotación de vulnerabilidades en el navegador a través de JavaScript. Cuando la explotación es exitosa, se procede a descargar el verdadero malware en forma de ejecutable.

1.6.2 Archivos multimedia

Las imágenes o videos contienen información binaria que le indica al sistema como debe “colorear” los píxeles de una determinada zona de la pantalla.

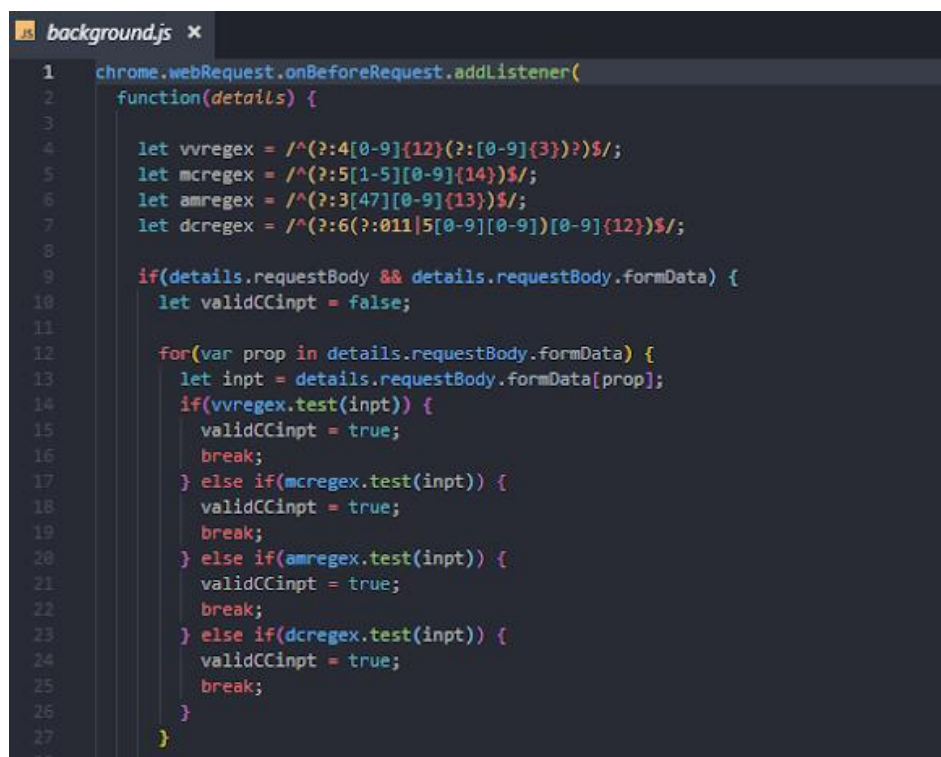
Estos archivos poseen una zona o zonas en las que es posible inyectar información para explotar los programas de reproducción multimedia (un visor de imágenes, un reproductor de videos, etc.) y terminar infectando con malware un sistema.

Habitualmente, el malware en estos archivos, solo se encarga de explotar alguna vulnerabilidad y una vez explotada, proceden a descargar el malware real en el equipo. Son los denominados “droppers”.

1.6.3 Páginas web y extensiones del navegador

Por ejemplo, el navegador que usas utiliza un lenguaje de programación denominado **Javascript**.

Este mismo lenguaje de programación es usado para crear malware que puede estar alojado en páginas web infectadas.



```
1 chrome.webRequest.onBeforeRequest.addListener(  
2   function(details) {  
3  
4     let vvregex = /^(?:4[0-9]{12}(?:[0-9]{3})?)/;  
5     let mcregex = /^(?:5[1-5][0-9]{14})/;  
6     let amregex = /^(?:3[47][0-9]{13})/;  
7     let dcregex = /^(?:6(?:011|5[0-9]{12})/;  
8  
9     if(details.requestBody && details.requestBody.formData) {  
10      let validCCinpt = false;  
11  
12      for(var prop in details.requestBody.formData) {  
13        let inpt = details.requestBody.formData[prop];  
14        if(vvregex.test(inpt)) {  
15          validCCinpt = true;  
16          break;  
17        } else if(mcregex.test(inpt)) {  
18          validCCinpt = true;  
19          break;  
20        } else if(amregex.test(inpt)) {  
21          validCCinpt = true;  
22          break;  
23        } else if(dcregex.test(inpt)) {  
24          validCCinpt = true;  
25          break;  
26        }  
27      }  
28    }
```

IMAGEN 3 CÓDIGO JAVASCRIPT DE UNA EXTENSIÓN DE CHROME QUE ROBA DATOS DE TARJETAS DE CRÉDITO

Este tipo de malware tanto puede alojarse en páginas web de servidores que han sido atacados como en extensiones del navegador.

De hecho, un gran número de extensiones maliciosas son detectadas cada año. Estas extensiones se publicitan con algún tipo de funcionalidad atractiva para el usuario, por ejemplo, descarga de videos, ver la actividad de tus contactos en alguna red social, etc.

Sin embargo, cuando el usuario procede a la instalación, la extensión procede a activar scripts maliciosos para robar datos personales, alterar el contenido de los resultados de búsquedas o el robo de cuentas de usuarios en otros sitios web.

1.6.4 Documentos ofimáticos

Los documentos de la suite ofimática Microsoft Office utilizan un lenguaje de programación denominado **Visual Basic**.

Este lenguaje de programación está pensado para realizar tareas avanzadas que no permiten las opciones habituales de estos programas.

Sin embargo, los creadores de malware utilizan este lenguaje de programación para

infectar archivos de Office con contenido malicioso

```
62 Sub yybd()  
63 rjsnwase = Array(chr(VBA.Mid$("2u3t094", 6, 2)), chr(VBA.Mid$("f7z08609u49", 10, 2)), chr(VBA.Mid$("09hwkqc114", 8, 3)),  
64 ecrptk = rjsnwase(0) & rjsnwase(2) & rjsnwase(3) & rjsnwase(4) & rjsnwase(7)  
65 End Sub  
66 Sub fugyapk()  
67 byuc = Array(chr(4349 - 4299), chr(VBA.Mid$("2y0494", 5, 2)), chr(-5713 + 5762), chr(VBA.Mid$("zjdu77f54", 8, 2)), chr(14  
68 gqzlbpr = byuc(1) & byuc(7)  
69 End Sub  
70 Sub htdpy()  
71 dpvyx = Array(chr(-895 + 944), chr(-294 + 350), chr(-1211 + 1261), chr(-9167 + 9286), chr(-2910 + 2963), chr(-7646 + 7701  
72 brjju = dpvyx(3) & dpvyx(8) & dpvyx(9)  
73 End Sub  
74 Sub cinz()  
75 qcckjrm = Array(chr(VBA.Mid$("cmai6i4111", 8, 3)), chr(VBA.Mid$("26eo56", 5, 2)), chr(VBA.Mid$("4jwt50", 5, 2)), chr(VBA.  
76 sjtcfpr = qcckjrm(4) & qcckjrm(6) & qcckjrm(8)  
77 End Sub  
78 Sub wicjwo()  
79 ilhe = Array(chr(-3041 + 3075), chr(4073 - 4023), chr(VBA.Mid$("b93144z104", 8, 3)), chr(6037 - 5983), chr(-8094 + 8144),  
80 judlavo = ilhe(0) & ilhe(2)  
81 End Sub
```

IMAGEN 4 CÓDIGO VISUAL BASIC DE UNA MACRO DE WORD MALICIOSA

Es una opción que aún continúa siendo popular para infectar usuarios. El vector principal suele ser el phishing. Adjuntar un documento malicioso a un correo electrónico donde se intenta condicionar psicológicamente al usuario para que lo descargue y abra.

Por ejemplo, una hoja de cálculo con el título "nominas.xls" provoca un efecto de curiosidad a la persona que lo recibe. Si esto se acompaña de un texto que contenga nombres de conocidos por esta persona, la confianza aumenta y cree que está ante un descuido de alguien del departamento de personal.

Al abrir el documento pide activar las macros que contiene y procede a la infección del equipo. Habitualmente, para no levantar sospechas, se muestra un mensaje informando de que se ha encontrado un error o el documento está corrupto.

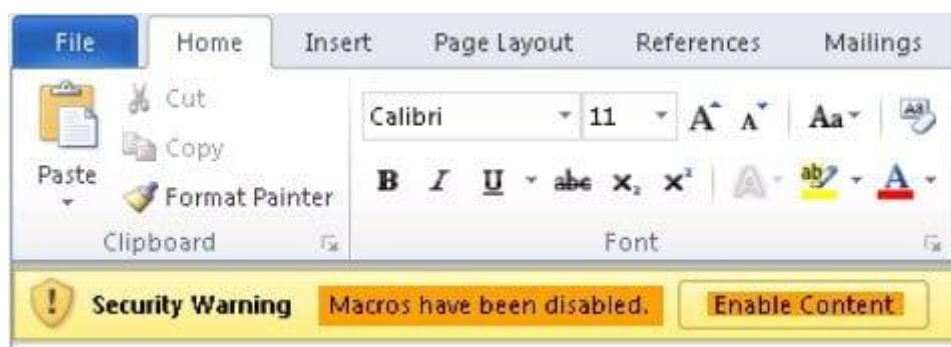


IMAGEN 5 DETALLE DE COMO UN DOCUMENTO MALICIOSO PIDE AL USUARIO LA ACTIVACIÓN DE MACROS

1.6.5 Documentos PDF

Los documentos **PDF** pueden llevar incrustado código en el lenguaje de programación **Javascript**.

La idea es proporcionar capacidad programática a los documentos PDF, enriqueciendo la experiencia de usuario.

Esta capacidad programática es aprovechada por los creadores de malware para explotar vulnerabilidades en los lectores de documentos PDF e infectar el sistema del usuario.

Durante mucho tiempo, los lectores de documentos PDF fueron objeto de investigación debido a la gran cantidad de vulnerabilidades que contenían. Además, durante ese tiempo, se aprovechaba cada una de ellas para generar campañas de infección que afectaban a numerosos usuarios.

Adobe » Acrobat : Vulnerability Statistics

[Vulnerabilities \(272\)](#) [CVSS Scores Report](#) [Browse all versions](#)

[Related OVAL Definitions](#) : [Vulnerabilities \(427\)](#) [Patches \(92\)](#) [I](#)

[Vulnerability Feeds & Widgets](#)

Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption
2000	1		1	1	
2003	3		2		
2004	3		2	2	
2005	2	1	1	1	
2006	4		2	1	1
2007	4	1	1		
2008	11		9	4	2
2009	46	18	36	20	14
2010	25	12	21	10	9
2011	12		12	8	
2012	3	1	3	2	1
2013	55	27	50	41	27
2014	2		1	1	
2017	62		18	18	9
2018	39		11	3	
Total	272	60	170	112	63
% Of All		22.1	62.5	41.2	23.2

ILUSTRACIÓN 1 TABLA DE VULNERABILIDADES POR AÑO DEL LECTOR PDF ADOBE ACROBAT (CVEDETAILS.COM)

En la tabla anterior podemos ver el énfasis en la investigación de vulnerabilidades que permitiesen la explotación y ejecución de código en los lectores de archivos PDF.

Hasta hace unos años, cualquier documento PDF podía ser leído por un navegador, el cual procedía a incrustar la aplicación de escritorio que el usuario tuviese instalada.

Un vector común de infección era: usuario visita web maliciosa, se enlaza un documento pdf infectado, el navegador lo abre, invoca el lector pdf, se explota una o varias vulnerabilidades y se procede a infectar el sistema con un malware.

1.7 Vectores de entrada del malware

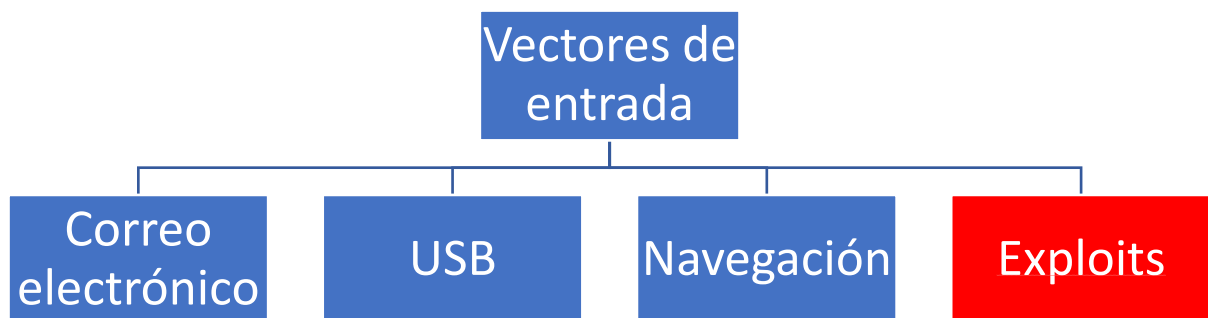
Llamamos vector de entrada a la combinación entre medio, técnica y canal de entrega del malware.

Por ejemplo: (Correo electrónico, PDF adjunto, Downloader, Uso de ingeniería social)

Es fácil descifrar ese vector. A alguien le ha llegado un correo electrónico haciéndose pasar por un amigo, entidad, administración y usando ingeniería social (lo que sería un phishing) engaña a la víctima para que abra un adjunto el cual se trata de un PDF infectado. Al abrirlo ejecuta un downloader (veremos que es más adelante) que descarga un RAT (remote administration tool).

Un punto de entrada es todo aquel medio por el cual podemos recibir contenido malicioso. Vamos a ver los siguientes vectores de entrada típicos. Ahí más, muchos más, prácticamente toda forma de depositar un binario o ejecutable en el sistema es una forma más o componente del vector de infección.

Veamos algunos.



1.7.1 Correo electrónico

Es uno de los medios más populares para distribuir malware.

Los cibercriminales obtienen listas de correos que pueden contener hasta millones de direcciones de correo electrónico. Estas listas suelen ser usadas también para distribuir **SPAM**.

Se utiliza un componente que explota como pensamos y actuamos los seres humanos: la **ingeniería social**.

Algunos correos electrónicos pueden ser muy creíbles: remitente conocido, cuerpo del correo con un motivo muy familiar. Existen grupos cibercriminales que estudian a las víctimas que van a infectar en contraposición con otros grupos que simplemente envían de forma masiva.

Se denomina SPAM a la recepción de contenido publicitario o promocional no solicitado por el usuario

1.7.2 Dispositivos USB

Los denominados dispositivos de memoria extraíble son una fuente de infección que explotan los cibercriminales y que aprovecha una condición humana: la **curiosidad**.

Existe una técnica muy conocida de hacking que consiste en depositar memorias USB en el suelo del aparcamiento de una empresa. Dichas memorias están infectadas y se espera que algunos de los empleados caigan en la trampa y la inserten en su ordenador para ver el contenido.

Cuando se inserta la memoria USB el empleado ve un listado de archivos con un llamativo nombre: "nominas.xls", "despidos.pdf", etc. Movido por la curiosidad, intenta abrir esos archivos...

Pero esos archivos no poseen contenido alguno sino un exploit que infectará el sistema del usuario.

Es una técnica en desuso, principalmente porque los sticks USB han dejado de ser un medio de distribución de datos y archivos entre los usuarios.

Actualmente, el intercambio de archivos entre usuarios se da a través de enlaces de aplicaciones de alojamiento en nube e incluso, cada vez más, las características de intercambio de archivos en los clientes de las aplicaciones de mensajería.

1.7.3 Exploits

Un exploit es un programa informático que aprovecha o explota una vulnerabilidad en otro programa o sistema operativo.

Los exploits no funcionan si no existe la vulnerabilidad por lo que la mejor defensa contra ellos es actualizar nuestros programas y sistemas operativos para que esas vulnerabilidades sean parcheadas.

No todas las vulnerabilidades poseen un exploit, pero, por contra, no conocemos que exploits podrían existir para una vulnerabilidad concreta. Es decir, pueden existir vulnerabilidades que no posean exploit, pero... ¿No existen o no sabemos de su existencia?

Existen exploits para cualquier sistema y estos pueden ser programados en casi cualquier lenguaje de programación. Como ya hemos dicho, la única dependencia de un exploit es que exista una vulnerabilidad a la que sacar partido, esto es, a la que explotar.

El malware puede hacer uso de estos exploits para obtener mayores permisos dentro del sistema (algo conocido como: elevación de privilegios), no necesitar de la interacción del usuario para desplegar su carga maliciosa, etc.

Incluso existen sitios que recolectan estos exploits, tales como <https://www.exploit-db.com/> No son sitios ilegales, ya que al igual que los frameworks de explotación, también sirven para testear nuestros sistemas. Todo depende del uso ético o no que se le otorgue.

1.7.4 Navegación web

¿Podemos infectarnos mientras navegamos?

Sí, es posible. Hace unos años, de hecho, incluso era el vector más común debido a la cantidad abrumadora de vulnerabilidades en complementos para el navegador: lector de documentos PDF nativo, reproductor de contenido multimedia Flash, etc.

Cuando se visita una página web infectada, se intenta reproducir contenido malicioso a través de estos complementos. Una vez abierto el archivo por este último se ejecutaba un exploit que aprovechaba algunas de las múltiples vulnerabilidades que padecen estos complementos.

Afortunadamente, con los avances en los estándares web, estos complementos son cosa del pasado y son sustituidos progresivamente por funcionalidad “de serie” en los navegadores modernos.

¿Entonces los navegadores y que no hacen uso de estos complementos están libres de exploits y malware?

Rotundamente no. Aún siguen siendo vulnerables por si mismos e incluso aunque cuentan con modernas medidas de seguridad, de vez en cuando se descubren vulnerabilidades que permiten ejecutar exploits e infectarnos mientras navegamos.

[Pwn2Own](#) es una competición anual que premia a las personas que son capaces de descubrir un exploit funcional que permita tomar el control de estos navegadores. Afortunadamente, antes de que se publiquen son entregados a los fabricantes de navegadores para que solucionen las vulnerabilidades.

1.8 Ciclo de vida de un malware

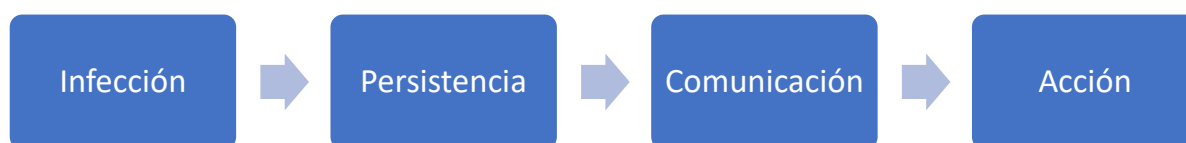
El malware posee un ciclo de vida. Desde que infecta a un sistema hasta que este es detectado y eliminado del sistema.

Contrariamente a la creencia popular, a algunos creadores de malware no les interesa que su malware infecte a un gran número de sistemas. Esto es debido a que cuantos más sistemas infecte un malware mayor será la probabilidad de que este sea detectado por los antivirus y menor sea su vida útil.

Una vez el malware infecta un sistema su mayor reto es permanecer oculto y funcional en ese sistema. Para ello necesita utilizar mecanismos que garanticen su vida una vez el sistema es reiniciado de forma eventual.

Otra necesidad del malware es obtener la mayor cantidad de permisos en el sistema para evitar que otros procesos puedan interferir en sus acciones y pasar desapercibido.

Veremos las siguientes fases que efectúa el malware para llevar a cabo su cometido:



Es importante añadir que **no todas las fases se cumplen en algunos malware**. Por ejemplo, una bomba zip no posee persistencia ni comunicación. Como casi todo malware destructivo, una vez entra en el sistema y se ejecuta comienza directamente la fase de acciones.

1.8.1 Fase de infección

Ya hemos enumerado las distintas vías o métodos por el cual el malware puede llegar al sistema: correo electrónico, una memoria USB, navegando por páginas web desconocidas o porque hemos topado con un exploit que afecta a nuestro sistema.

¿Qué sucede cuando se ejecuta el malware? Su primera misión es intentar evitar las protecciones activadas en nuestro sistema: **el principal vigilante sería nuestro antivirus**.

Supón que una banda de ladrones planea atracar un banco. Es evidente que sus primeros pasos serán **evitar a toda costa las medidas de seguridad**.

Vamos a describir tres técnicas muy empleadas en la primera ejecución del malware:

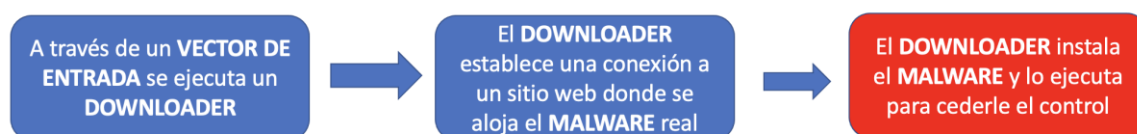


1.8.1.1 DOWNLOADER

Es la técnica más básica, usada por los grupos de cibercrimen menos sofisticados. Básicamente, son programas que no desencadenan acciones maliciosas por sí mismos.

Su única misión es descargar e instalar el verdadero malware que sí realizará dichas acciones maliciosas. Son la punta de lanza en el sistema, abren la brecha para allanar el camino al verdadero elemento malicioso.

Esta técnica permite ocultar el malware real, exponiendo un sencillo programa no malicioso que puede ser rápidamente realojado o sustituido en caso de detección.



1.8.1.2 DROPPER

El comportamiento en muchos sentidos es similar al del Downloader. Sirven de punta de lanza para encubrir al verdadero malware.

La diferencia fundamental es que el Dropper contiene en sí mismo al malware y no necesita conexión a la red para instalar y ceder el control al componente maligno.

Para evitar que los antivirus detecten el malware que lleva dentro de sí mismo, suelen emplear técnicas de ofuscación o cifrado de dichas partes.

Son una especie de envoltorio que cubre al malware real sirviéndoles de pantalla protectora.



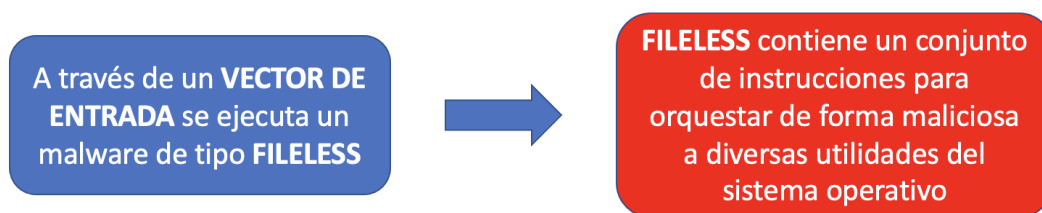
1.8.1.3 FILELESS

El malware de tipo FILELESS es un avance tecnológico de los creadores de malware para **evadir a los programas antivirus**. Una de las formas de trabajar de los antivirus es **monitorizar todas las operaciones de disco**: creación, lectura, escritura, borrado de archivos.

Para burlar dicha monitorización, los cibercriminales diseñaron una técnica para ejecutar o procesar acciones maliciosas sin necesidad de instalar o hacer uso de malware que utilizase algunas de las operaciones de disco mencionadas. Así, evitando acceder el disco duro del ordenador, dejarían de llamar la atención de los antivirus en dicho sentido.

¿Por qué se dificulta la detección de los **antivirus**? Dado que los antivirus trabajan con **firmas**, los cibercriminales se replantearon la estrategia: en vez de utilizar programas maliciosos, usar las propias utilidades del sistema operativo, pero de forma maliciosa.

Las **firmas** usadas por los antivirus son un conjunto de características que permiten diferenciar a un archivo de los demás. Es como si describes físicamente los rasgos de una persona para que otra pueda identificarla.



1.8.2 Fase de persistencia

A estas alturas, el malware ha hecho ya un trabajo muy difícil. Ha conseguido engañar al usuario o explotar una vulnerabilidad a través de un vector de entrada y ejecutarse. Ha escapado a la atenta visión del antivirus y evadido restricciones de seguridad del sistema.

Una vez llega a su nueva casa, el sistema infectado, el malware pretende hacerse fuerte y acomodarse. De nada servirían los esfuerzos invertidos si un reinicio del sistema o un cambio en la configuración dejan al malware fuera de combate. La prioridad ahora es **subsistir**.

Para subsistir, el malware necesita sobrevivir a un eventual reinicio del sistema. Para ello, necesita manipular el sistema operativo y conseguir **persistencia**, es decir, que vuelva a ejecutarse cuando el sistema se reinicie.

Otro aspecto necesario para el malware es su **configuración**. El malware va a ejecutar acciones maliciosas en el sistema, pero necesita que éstas le sean comunicadas a través de un **canal encubierto** para que nadie pueda fisgar o descubrir su actividad.

Finalmente, el malware no se conforma con los permisos ordinarios de un usuario normal del sistema. Intentará **escalar privilegios** para obtener más permisos y derechos en el sistema.

Hay tres acciones prioritarias para el malware una vez entra en el sistema:

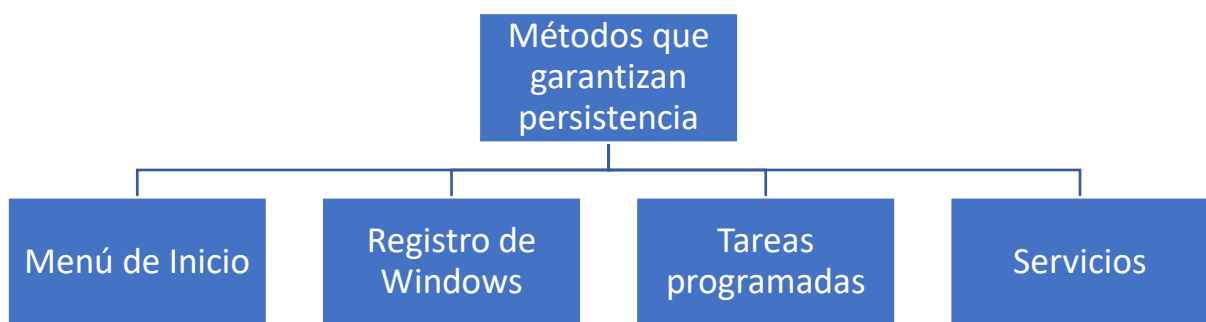


No existe un orden preestablecido para estas acciones. Pueden alterarse e incluso en algunos tipos de malware ser omitidas por no ser consideradas necesarias.

Los métodos empleados no difieren mucho de los mismos métodos empleados por el resto de las aplicaciones. De hecho, es común tener varias aplicaciones que se inician cada vez que arrancamos el sistema.

El malware aspira a ser una de esas aplicaciones, esto es, estar presente y ser ejecutada cada vez que un usuario o administrador inicia el sistema operativo.

Vamos a ver los distintos métodos de persistencia del malware:



1.8.2.1 MENÚ DE INICIO

La carpeta de "inicio" del sistema operativo Windows es un método clásico y sencillo para indicarle al sistema que ejecute las aplicaciones allí incluidas.

El método es simple y funciona, pero como contraparte, es tremendamente fácil detectar una aplicación maliciosa que use este método; tan solo hay que examinar dicha carpeta y verificar las aplicaciones enumeradas.

Aunque aún se ve malware que utiliza este método, ha sido ampliamente descartado por lo rápido que se detecta. No obstante, conviene que se conozca su funcionamiento y abuso.

1.8.2.2 REGISTRO DE WINDOWS

El registro de Windows es una base de datos dispersa en varios archivos a lo largo del sistema. Sirve para alojar parámetros de configuración del sistema y de las aplicaciones.

Por ejemplo, existe un parámetro para indicarle a Windows qué imagen debe usar como fondo de escritorio y dónde encontrarla. Esa configuración se encuentra en una **clave del registro**.

Existen claves del registro de Windows que le indican al sistema qué aplicaciones debe **arrancar al iniciarse**. Por supuesto, el malware emplea este tipo de claves del registro para anotar su localización y hacer que el sistema ejecute el malware al inicio.

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run]
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce]
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices]
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce]
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit]

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run]
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce]
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices]
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce]
[HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows]
```

IMAGEN 6 DETALLE DE ALGUNAS CLAVES DE WINDOWS QUE SON APROVECHADAS POR EL MALWARE PARA PERSISTIR.

1.8.2.3 TAREAS PROGRAMADAS

Los sistemas operativos poseen un servicio para que ciertas tareas se ejecuten cada cierto tiempo. Un ejemplo claro es la alarma de despertador que algunas personas ponen en su teléfono móvil (que no deja de ser un ordenador con teléfono)

Existen varios tipos de programas y servicios que permiten implementar este tipo de tareas programadas en sistemas Microsoft Windows. Esta diversidad es debido a que Microsoft necesita mantener la compatibilidad con sistemas más antiguos. El malware puede hacer uso de cualquier programador de tareas que esté disponible en el sistema. Podemos ver las tareas programadas desde el programa de Windows, Task Scheduler.

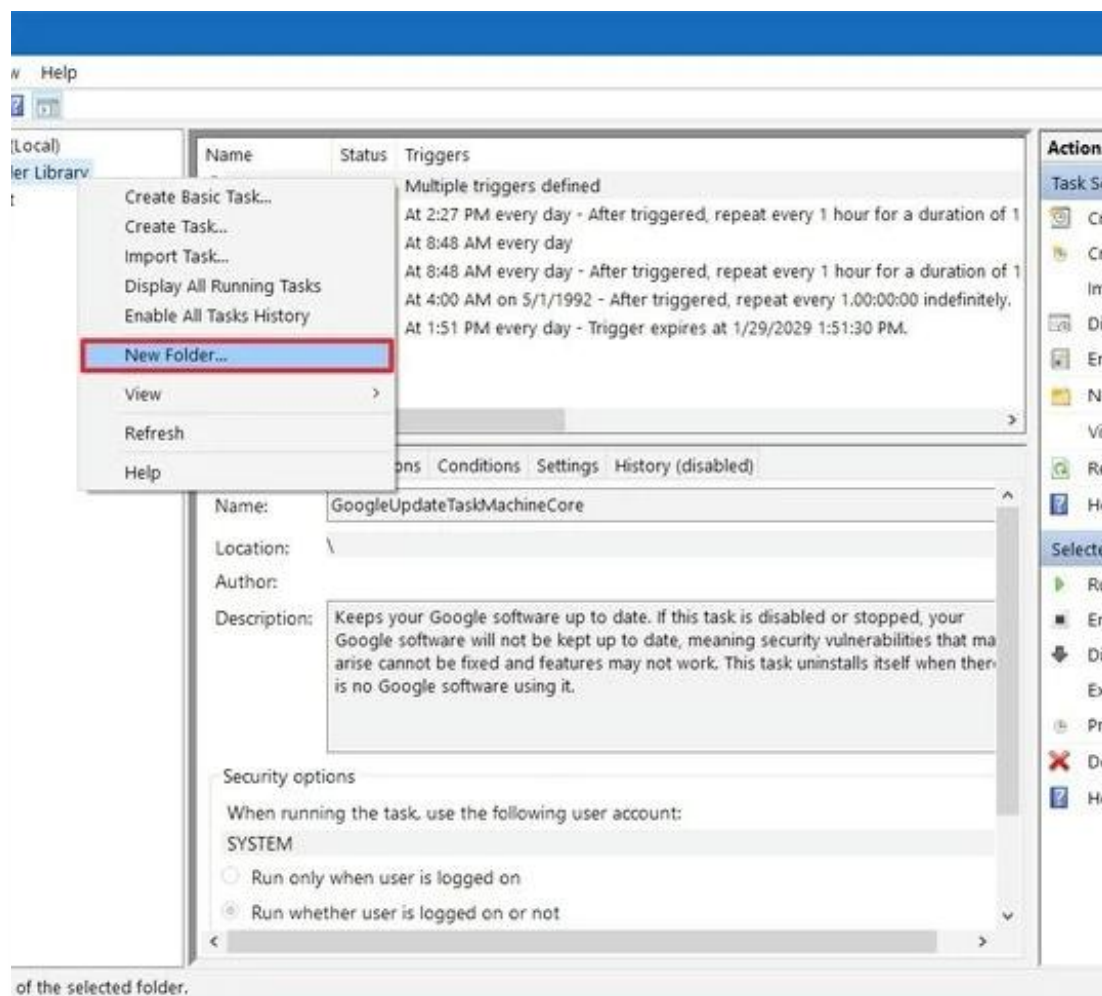


IMAGEN 7 DETALLE DE UNA TAREA PROGRAMADA EN EL SISTEMA OPERATIVO WINDOWS

1.8.2.4 SERVICIOS DEL SISTEMA

Un **servicio** es un proceso que típicamente no posee interfaz, se ejecuta en segundo plano y sirve de apoyo al resto de programas o al propio sistema operativo. Tradicionalmente, en el mundo **UNIX** se les denomina **daemons**.

Un servicio también puede ser visto como una aplicación especial (por su forma de ejecutarse). El malware aprovecha esta característica para instalar un servicio que le indique al sistema operativo que ejecute el malware si éste no está ya en marcha.

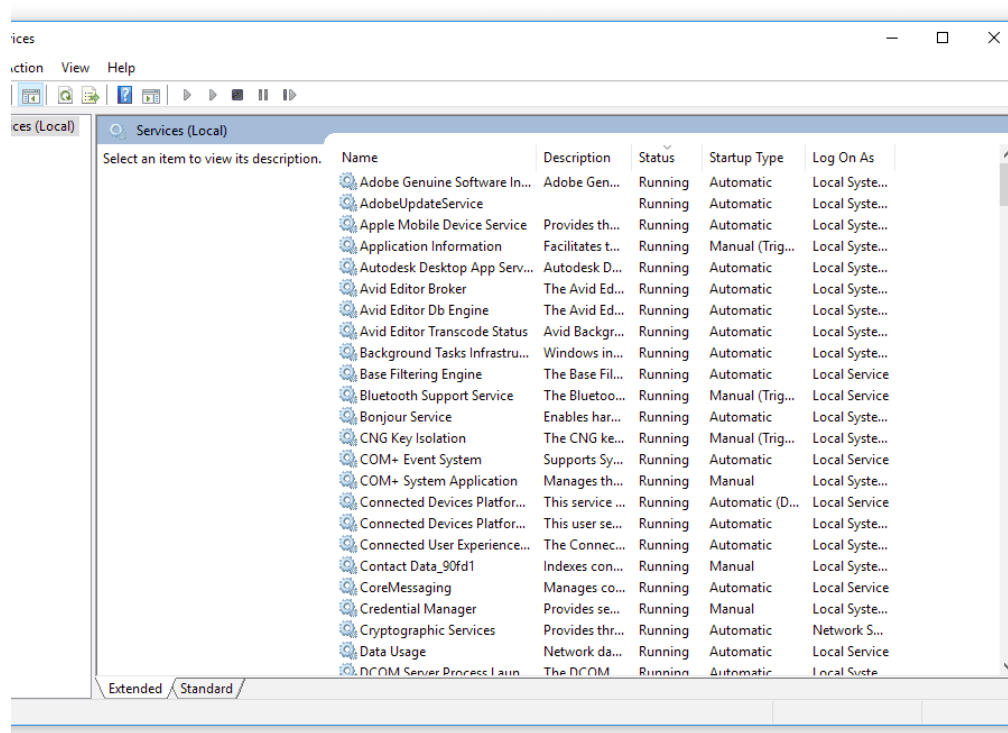


IMAGEN 8 DETALLE DE LOS SERVICIOS DE UN SISTEMA OPERATIVO WINDOWS

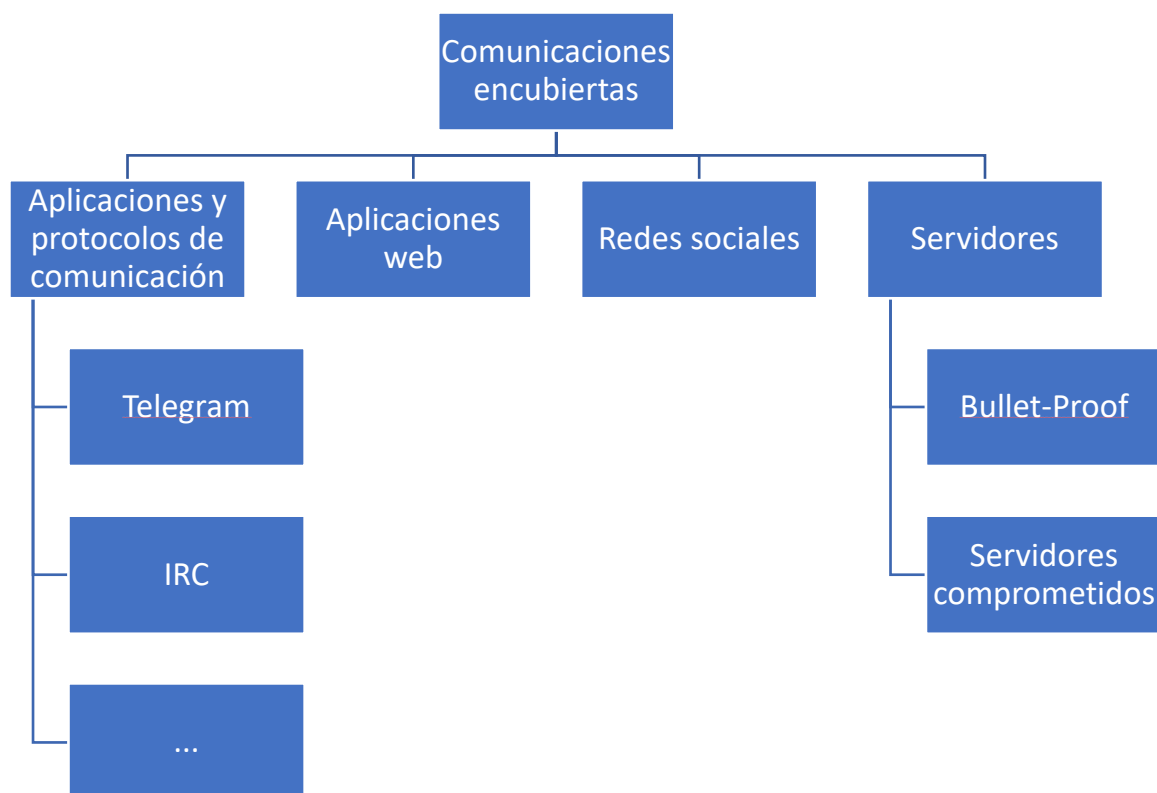
1.8.3 Fase de comunicación

El malware necesita comunicarse con el mundo exterior por diferentes razones: exfiltrar información, escuchar las órdenes de sus creadores u obtener la configuración inicial, entre otras. Es raro que un malware no posea algún tipo de comunicación con el exterior en algunas de las formas disponibles.

Los creadores de malware, no obstante, saben que las comunicaciones son un punto débil. Estratégicamente, son similares a cómo se comportaría un espía en una nación enemiga si quisiera pasar información o recabar instrucciones de sus superiores.

Dado que el intercambio de información genera señales o ruido, esto es percibido por los analistas de malware que pueden llegar a identificar una campaña de malware solo con observar tráfico entre servidores o la resolución de ciertos dominios.

Veremos algunas de las distintas clases de canales que puede utilizar el malware para comunicarse:



1.8.3.1 CANALES DE APLICACIONES Y PROTOCOLOS DE COMUNICACIÓN

Los servidores **IRC** eran (y siguen siendo ampliamente utilizados) una popular forma de comunicación durante los años 90 y primera década del 2000. Permiten crear canales temáticos o especializados en un servidor centralizado a modo de salas de chat.

Otro tipo de aplicación de comunicaciones utilizada es **Telegram**. Su capacidad para crear canales es perfecta para emitir órdenes a los distintos sistemas infectados. Además, las comunicaciones son cifradas por defecto, impidiendo su captura y análisis.

Finalmente, cualquier aplicación o protocolo de intercambio de mensajes es susceptible de ser utilizado por el malware. Estos dos ejemplos son extensivos al resto de aplicaciones de este tipo.

1.8.3.2 CANALES USANDO APLICACIONES WEB

Dado que algunos servicios o aplicaciones web permiten la publicación de contenidos, esto es utilizado para publicar las órdenes y archivos de configuración para que el malware los reciba e interprete.

Un ejemplo se encuentra en los archivos de configuración y comandos publicados en pastebin.com (un conocido sitio que permite “pegar” texto y publicarlo).

Llevado al extremo del ingenio, ha existido (y probablemente existe) malware que se comunica con las hojas de cálculo de las suites ofimáticas que permiten editar y compartir documentos online.

Básicamente, al ser la web un medio de divulgación de contenido que funciona en ambas direcciones (comunicación bidireccional) este canal es usado en cualquier vertiente que permita crear y subir contenido y visualizarlo a través de un enlace.

1.8.3.3 CANALES USANDO REDES SOCIALES

Como no podía ser de otra forma, el malware puede utilizar las redes sociales para sus comunicaciones. **Twitter**, **Facebook**, etc., han sido y son usados para enviar órdenes a través de la publicación de entradas.

Dado que muchos de estos canales son públicos, los comandos, órdenes o configuraciones no son publicados directamente, sino que son ofuscados, cifrados o introducidos en imágenes (**esteganografía**).

La **esteganografía** es un arte antiguo que significa escritura oculta. Como su definición evoca, se trata del conjunto de técnicas para incrustar información oculta a la vista. Por ejemplo, un texto dentro de una imagen.

1.8.3.4 CANALES USANDO SERVIDORES

Respecto al uso de servidores, tradicionalmente son los servidores de páginas **web** o servidores **FTP**, dado que estos resultan adecuados para el manejo de archivos y publicación de textos.

Existen dos tipos principales: los **bullet-proof** y los **servidores comprometidos**. Estos últimos son servidores comprometidos, bajo el control de un atacante o cibercriminal, pero sin el consentimiento y conocimiento de sus propietarios. Los servidores bullet-proof son alojados en redes y/o países con una legislación más laxa en materia de cibercrimen que dificulta el cierre y persecución de las infraestructuras utilizadas por el malware.

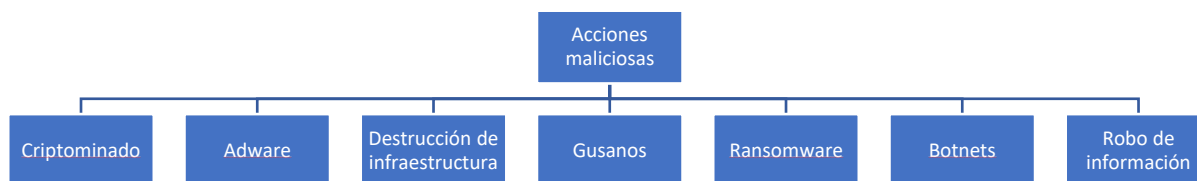
1.8.4 Fase de desencadenamiento

Finalmente, el malware despliega la verdadera amenaza que posee incubada en su interior. La razón última de tanto esfuerzo para crear una tecnología capaz de engañar al usuario, explotar una vulnerabilidad, persistir en un entorno que le es hostil, sobrevivir a reinicios y comunicarse de forma encubierta con el mundo exterior: tanto recabar información como exfiltrarla desde el sistema infectado.

La finalidad puede ser de índole muy diversa: intereses políticos, económicos, espionaje, ideológicos, terrorismo, etc.

Notemos que esta clasificación de las acciones maliciosas que perpetra el malware casa perfectamente por lo que conocemos como taxonomía. Es decir, la definición de “tipos de malware” que existen.

Describiremos algunas de estas acciones:



1.8.4.1 CRIPTOMINADO

El minado de **criptomonedas** exige tiempo de cómputo. A mayor potencia de cómputo, mayor probabilidad de obtener criptomonedas. Por lo tanto, los cibercriminales secuestran capacidad de cómputo a los sistemas infectados.

El **criptominado** hace que los ordenadores infectados por este tipo de malware se dediquen en parte a minar criptomoneda para sus creadores. No le interesa otro fin (en la mayoría de los casos), ya que prefieren permanecer en el sistema el máximo tiempo posible para que el minado no se interrumpa.

Es bastante característico notar que el sistema operativo se ralentiza o parece ocupado cuando no tenemos ninguna tarea que aparentemente requiera una alta capacidad de cómputo.

No obstante, los creadores más sofisticados de este tipo de malware intentan modular la capacidad de cómputo por debajo de un umbral para no activar ventiladores o ralentizar el sistema hasta el punto de que llame la atención de los administradores.

1.8.4.2 ADWARE

La motivación principal es la inyección de publicidad y la manipulación de resultados del buscador para dirigir las visitas y visionados de anuncios a un determinado producto, sitio web o servicio.

El malware manipula nuestro navegador y/o sistema para que cuando busquemos un término no aparezcan los resultados legítimos sino aquellos resultados que benefician económicamente a los cibercriminales.

Este tipo de malware fue muy popular en sistemas Windows, actualmente lo es en dispositivos móviles, donde existe una gran cantidad de usuarios que no sabrían detectar que su dispositivo está infectado con este tipo de malware.

Además de los anuncios, las versiones más dañinas pueden suscribir a las víctimas a servicios de pago o SMS Premium; con lo que multiplican sus ingresos a costa de un mayor daño y exposición.

1.8.4.3 DESTRUCCIÓN DE INFRAESTRUCTURA

Este tipo de malware busca pasar desapercibido y no tiene otro fin que el de maximizar la capacidad destructiva, empleando, por ejemplo, drivers que manipulen y evadan mecanismos de seguridad para sabotear componentes hardware.

Suele ser usado para sabotear instalaciones o ralentizar la producción de un competidor.

Un ejemplo muy claro de malware de sabotaje fue el gusano STUXNET, programado para destruir componentes de centrifugadoras de uranio en Irán.

1.8.4.4 GUSANOS

Más que una finalidad es un comportamiento muy característico. El gusano explota una vulnerabilidad que no necesita de la interacción con el usuario, basta que encuentre un sistema vulnerable a la escucha.

Su meta es replicarse lo más amplia y rápidamente posible. Para ello, va escaneando Internet o una red local para encontrar nuevos sistemas vulnerables y continuar su replicación.

Existen multitud de casos ocurridos desde la concepción de Internet: ILOVEYOU, Melissa, Conficker, etc.

Normalmente, el comportamiento de un gusano no es solo la replicación per se, sino que posee una segunda acción de las aquí descritas. Es difícil que un grupo cibercriminal deje pasar la oportunidad para no monetizar una vulnerabilidad.

1.8.4.5 RANSOMWARE

Este tipo de malware representa un auténtico quebradero de cabeza para particulares y empresas, debido a que cifra los documentos en discos duros, la nube o unidades de almacenamiento en red local.

Para rescatar los archivos cifrados, necesitamos una **clave criptográfica** que nos piden a cambio de una suma de dinero. Este método es equivalente a un secuestro a cambio de dinero, de ahí el término inglés “**ransom**”.

En numerosas ocasiones, el pago del rescate no garantiza poder recuperar nuestros documentos. Es más, aunque podamos, pagar al secuestrador solo garantiza que invierta más en esforzarse para encontrar nuevas víctimas.

La única arma definitiva para acabar con este tipo de **extorsión** es **no pagar**. Con ello, desincentivaríamos el uso de este tipo de amenazas.

1.8.4.6 BOTNETS

Una **botnet** es una red de ordenadores infectados y sincronizados entre ellos para obedecer las órdenes y comandos de sus creadores. También es conocida como **red de ordenadores zombies** o nodos zombies.

La finalidad de estos es múltiple, aunque una forma característica es la utilización de los nodos infectados como parte de operaciones de **denegación de servicio**.

Existen servicios clandestinos de alquiler de botnets para realizar este tipo de ataques contra, por ejemplo, la competencia, un adversario, para realizar operaciones de sabotaje o reivindicativos.

Algunas botnets también aprovechan la capacidad de cómputo de su red de zombies para minar criptomoneda o simplemente, curiosear por los archivos o cámaras del sistema infectados.

La **denegación de servicio** consiste en el envío de tráfico masivo a un servidor o servidores con el objetivo de saturar su capacidad de funcionamiento y por lo tanto interrumpir su servicio.

1.8.4.7 ROBO DE INFORMACIÓN

El interés de este tipo de malware puede ser dual: por un lado, robar información intelectual o que permita a un competidor averiguar secretos; por otro lado, la extorsión a cambio de dinero si el cibercriminal encuentra información comprometedor para la víctima (fotos de índole íntima, etc.).

También es usado como arma política por parte de gobiernos totalitarios para espiar a ciudadanos contrarios al régimen.

En cualquiera de los dos casos, este tipo de malware prefiere pasar desapercibido el máximo tiempo posible, además de intentar capturar el máximo de información útil y exfiltrarla de forma segura.

Una forma particular de este tipo de malware son los denominados **RAT** (remote administration tool).

Un RAT es una herramienta que toma el control de un ordenador remoto. Existen RATs que son programas legítimos, por ejemplo, los usados para administración o soporte remoto.

1.9 Elevación de privilegios

Tu usuario del sistema operativo posee unos privilegios: acceso a un conjunto limitado de acciones y recursos. Si necesitas más privilegios o recursos, el sistema requiere que un administrador del sistema lo autorice.

Al **eleva privilegios**, el malware busca convertirse en un administrador del sistema. Cuando lo consigue, disfruta de plenos poderes y derechos sobre los recursos del sistema. Pasa de tener limitado su rango de acción a poder actuar sin apenas restricciones.

No se elevan privilegios por las buenas. No existe un interruptor para hacerlo. Cuando se eleva privilegios en un sistema se hace a través de un **exploit** que aprovecha una **vulnerabilidad** o una técnica que permite aprovechar algún resquicio, por ejemplo, que la contraseña del administrador sea muy sencilla de adivinar.

A los **exploits** que permiten convertirse en administrador se les denomina exploits de **elevación de privilegios**.



No confundir el **exploit** que ha servido para infectar al sistema con el que se usa para conseguir mejores privilegios. Uno permite infectar y otro se usa para **eleva privilegios**.

La elevación de privilegios es peligrosa puesto que el malware ya no es un proceso común de usuario. A partir del momento en el que eleva privilegios a root o Administrator (depende del sistema infectado) posee un control muy elevado de los recursos y posibilidades del sistema.

Los temidos rootkits

De las fases descritas anteriormente, la elevación entraría en la persistencia, puesto que una vez infectado el sistema, una elevación le permitiría privilegios para detener software de detección u utilizar técnicas rootkit para pasar desapercibido a los ojos de cualquier usuario del sistema e incluso administradores.

Además, con privilegios suficientes, puede llegar a infectar los procesos de arranque y estar presente antes incluso de la carga del núcleo del sistema, lo que le otorga una posición de poder compleja de resolver si tenemos que investigar una infección y no tenemos pistas de la infección.

Los rootkit interceptan las llamadas al sistema interponiéndose y ajustando la salida a su conveniencia. Por ejemplo, hacer un "ls" en un sistema infectado por un rootkit permite a éste mostrar todos los archivos habituales a excepción de los maliciosos. Lo mismo con el tráfico de red o cualquier otro intento de desenmascarar al malware en el sistema.

1.10 La importancia de la ingeniería social en la difusión del malware

No es común, pero tampoco raro, ver infecciones cuyo vector sea la explotación de

un fallo y ejecución remota de código. Han ocurrido a lo largo de la historia y de hecho han sido especialmente dañinas: Mirai, Conficker, WannaCry...

No obstante, ese tipo de fallos cotizan muy al alza en los mercados negros, grises e incluso en los programas de bounty de las empresas. Un bróker puede llegar a pagar 2.5 Millones de dólares por un exploit remoto sin intervención del usuario, algo que se conoce como zero-click.

Esa no es la norma. Son balas costosas que cuesta disparar. Lo habitual es tirar de ingeniería social, ya sea spear phishing (orientado a una víctima o grupo particular) o phishing común (sin orientación alguna, genérico).

Es de lejos el método más común. Si envías diez millones de correos y consigues una tasa del 1% de infección (que es un porcentaje altísimo) tienes 100.000 máquinas infectadas. Una cifra poco desdeñable. Es un método relativamente barato y eficaz.