

**Escola Tècnica Superior d’Enginyeria**

**Electrònica i Informàtica La Salle**

Trabajo Final de Máster

Máster en Ciberseguridad

*Análisis de Keyloggers mediante ingeniería inversa.*

*(Estudio, Análisis y Desarrollo de Keyloggers)*

Alumno Profesor Ponente

Gabriel Martí Fuentes Marc Rivero López

|  |
| --- |
| **ACTA DEL EXAMEN**  **DEL TRABAJO FINAL DE MÁSTER** |

Reunido el Tribunal calificador en el día de la fecha, el alumno

D. Gabriel Martí Fuentes

expuso su Trabajo Final de Máster, el cual trató sobre el tema siguiente:

**Análisis de Keyloggers mediante   
ingeniería inversa.**

Acabada la exposición y contestadas por parte del alumno las objeciones formuladas por los Sres. miembros del tribunal, éste valoró dicho Trabajo con la calificación de

|  |
| --- |
|  |

Barcelona,

VOCAL DEL TRIBUNAL VOCAL DEL TRIBUNAL

PRESIDENTE DEL TRIBUNAL

# Resumen

La seguridad de los sistemas va más allá del control de acceso a estos, del análisis de vulnerabilidades del sistema o de infecciones por virus. Muchas veces el objetivo no es vulnerar el sistema en sí mismo, sino obtener información del usuario que lo usa. El espionaje comercial o industrial, y la revelación de secretos, es en la mayoría de veces el objetivo de los ciberdelincuentes, y una manera rápida de obtener información es saber todo lo que escribe el usuario que es objeto del espionaje. Por el teclado pasan los datos más importantes: nombres de usuario, contraseñas, números de cuenta y otros datos confidenciales.

Es por este motivo que los *keyloggers[[1]](#footnote-1)* han sido siempre una herramienta de uso bastante extendido, que ocupan poco espacio en memoria y pueden pasar inadvertidas durante mucho tiempo en un sistema.

En este documento se estudiarán los diferentes tipos de *keyloggers*, su origen, las técnicas utilizadas para catpurar las teclas, las técnicas para introducirlos en los sistemas, los métodos de ocultación, y como registran y envían la información a los ciberdelincuentes.

Para finalizar, se llevará a cabo el desarrollo de un *keylogger* con funcionalidades similares a alguno de los analizados.

***Nota***: Durante todo el documento se usará la palabra “*keylogger*” repetidas veces, y se podría haber usado su traducción al castellano, que vendría a ser “***registrador de teclas***” o bien “***dispositivo o software registrador de teclas***”, pero se ha preferido mantener el término original por ser más corto y facilitar así la lectura del documento.

Índice

[Resumen 1](#_Toc17659983)

[Índice 2](#_Toc17659984)

[1. Introducción 4](#_Toc17659985)

[1.1. Contexto 4](#_Toc17659986)

[1.2. Definición básica de KEYLOGGER 5](#_Toc17659987)

[2. Objetivos 6](#_Toc17659988)

[3. Historia y Estado del Arte 7](#_Toc17659989)

[4. Aspectos Legales 14](#_Toc17659990)

[5. Tipología 20](#_Toc17659991)

[6. Como captura los datos 22](#_Toc17659992)

[6.1. Captura en keylogger basado en Hardware 22](#_Toc17659993)

[6.1. Captura en keylogger basado en Software 23](#_Toc17659994)

[6.1.1. En una Máquina Virtual 23](#_Toc17659995)

[6.1.2. Basado en la API. 23](#_Toc17659996)

[6.1.3. Como rootkit 24](#_Toc17659997)

[6.1.4. Basado en Javascript 24](#_Toc17659998)

[6.1.5. Inyección de memoria. 26](#_Toc17659999)

[6.1.6. Event Tracing for Windows 26](#_Toc17660000)

[6.1.7. Basados en CSS 26](#_Toc17660001)

[6.1.8. Dispositivos móviles 27](#_Toc17660002)

[6.2. Otras técnicas usando canales laterales 28](#_Toc17660003)

[6.2.1. Criptoanálisis acústico. 28](#_Toc17660004)

[6.2.2. Emisiones electromagnéticas. 29](#_Toc17660005)

[6.2.3. Captura inalámbrica. 29](#_Toc17660006)

[7. Técnicas de distribución 31](#_Toc17660007)

[7.1. Ataques de incrustación de PDF 32](#_Toc17660008)

[7.2. Incrustación en documentos de Word 33](#_Toc17660009)

[7.3. Incrustación dentro de Scripts PowerShell 34](#_Toc17660010)

[8. Técnicas de ocultación y evasión 35](#_Toc17660011)

[9. Análisis de Keylogger Hardware 36](#_Toc17660012)

[9.1. KeeLog KeyGrabber Pico 16Mb 38](#_Toc17660013)

[9.2. Maltronics WiFi KeyLogger Pro 40](#_Toc17660014)

[10. Análisis de Keylogger Software 44](#_Toc17660015)

[10.1. Análisis de keylogger …. 46](#_Toc17660016)

[10.2. Caracteristica 1 46](#_Toc17660017)

[10.3. Caracteristica 2 46](#_Toc17660018)

[10.4. Caracteristica 3 46](#_Toc17660019)

[11. Anti-Keyloggers 48](#_Toc17660020)

[12. Reversing de un Keylogger 50](#_Toc17660021)

[12.1. Análisis preliminar del binario 50](#_Toc17660022)

[12.2. Análisis estático 57](#_Toc17660023)

[12.3. Análisis dinámico 64](#_Toc17660024)

[13. Desarrollo de un Keylogger 68](#_Toc17660025)

[13.1. Lenguaje elegido para el desarrollo. 68](#_Toc17660026)

[13.2. El Keylogger más corto 69](#_Toc17660027)

[13.3. Características del Keylogger 71](#_Toc17660028)

[13.4. Partes importantes del desarrollo 72](#_Toc17660029)

[13.5. Problemas y soluciones 72](#_Toc17660030)

[13.6. Implantación 72](#_Toc17660031)

[13.7. Mejoras posibles 72](#_Toc17660032)

[13.8. Notas finales 72](#_Toc17660033)

[14. Resultados finales 73](#_Toc17660034)

[15. Coste del proyecto 73](#_Toc17660035)

[15.1. Coste temporal 73](#_Toc17660036)

[15.2. Coste económico 74](#_Toc17660037)

[16. Conclusiones 75](#_Toc17660038)

[17. Líneas de futuro 75](#_Toc17660039)

[18. Referencias 76](#_Toc17660040)

[19. Referencias 79](#_Toc17660041)

[Índice de Ilustraciones 83](#_Toc17660042)

[Índice de tablas 85](#_Toc17660043)

# 

# Introducción

## Contexto

El espionaje digital es una actividad que va en aumento, y se realiza tanto a nivel gubernamental, como político o privado. Las herramientas y técnicas utilizadas son múltiples, y muchas utilizan sistemas combinados, pero el registro de pulsaciones en los teclados de los ordenadores, y más recientemente también en dispositivos móviles, es una de las características comunes a todos ellos.

El registro de audio o vídeo es más complejo, requiere de más recursos (más uso de memoria, almacenamiento o uso de procesador) y en muchos casos su actividad alerta más fácilmente al usuario.

En cambio, un pequeño programa que, a priori, simplemente registre las pulsaciones del teclado puede pasar desapercibido durante mucho tiempo y puede ser mucho más útil a los ciberdelincuentes o espías digitales.

Existen también dispositivos hardware que realizan esta misma función, y tienen sus ventajas e inconvenientes que comentaremos más adelante.



Ilustración 1. Dos modelos de keylogger hardware. Para teclado ps/2 y USB.

En este trabajo se citarán y analizarán brevemente un par de modelos de estos dispositivos hardware, pero nos centraremos principalmente en el estudio de los *keyloggers* por software.

Cabe decir, que también existe el keylogging en muchos sistemas cuyo objetivo no es espiar al usuario, sino como parte de estudio de cómo se interacciona con el sistema para ayudar a los fabricantes y desarrolladores en la mejora de futuros productos, o como parte de sistemas de predicción de texto en ayudas a la escritura.

## Definición básica de KEYLOGGER

Keylogger, proveniente de la unión de dos palabras inglesas: **key** (tecla) y **logger** (registrador) es, por lo tanto, un dispositivo o programa que registra pulsaciones de teclas.

Esta es la definición más corta que podríamos obtener para definir un keylogger.

No obstante, si profundizamos, podremos ver que los *keyloggers* (sobre todo los implementados por software), llegan a niveles de sofisticación más altos, y muchos no se limitan solo al registro de pulsaciones de teclado, sino que además pueden registrar movimientos y pulsaciones del ratón, información sobre las ventanas de las aplicaciones sobre las que se registran las pulsaciones del teclado o acciones del ratón, añaden capturas de pantalla, y en algunos casos, llegan a abrir un canal de «mando y control»[[2]](#footnote-2) para que se puede acceder y controlar remotamente el equipo dotándolo así de más funcionalidades y características. Pero como se ha dicho, si nos atenemos a su definición original estos últimos aspectos se salen del concepto de keylogger.

# Objetivos

Ante un hecho tan simple como el saber que escribe otra persona en el teclado de su ordenador, hay un mundo lleno de historias, técnicas y tecnología. El ser humano ha inventado métodos insospechados para saber que escriben otras personas y descubrir sus secretos que solo conocen sus teclados.

Es por esto, y por un conocimiento de todo lo que envuelve el simple hecho de conocer que tecla se ha pulsado en el teclado de un ordenador que fijaremos los siguientes objetivos para justificar la elaboración de este proyecto:

1. Mostrar el Estado del arte de los keyloggers.
2. Conocer los aspectos legales, y saber que dice la legislación Española y Europea sobre su uso.
3. Conocer las técnicas empleadas en la distribución e instalación de estos.
4. Estudiar las funcionalidades que incorporan.
5. Investigar las técnicas de ocultación o evasión que utilizan.
6. Desarrollo de un keylogger.

# Historia y Estado del Arte

La historia exacta de los keyloggers no se conoce (y es difícil que alguien la conozca), pero sí que existen algunos hechos históricos que nos sitúan cronológicamente en sus orígenes y evolución. Parte de este supuesto origen se puede establecer en la época de la “*Guerra Fría*”[[3]](#footnote-3) entre Estados Unidos y la Unión Soviética [1]. En esa época no era habitual el uso de ordenadores, sino que se utilizaban máquinas de escribir en oficinas y dependencias gubernamentales. En dicho contexto, hacia finales de los años 70, los Rusos desarrollaron unos dispositivos electromagnéticos que implantaron en las máquinas de escribir, en concreto los modelos ***IBM Selectric II y III*** [[4]](#footnote-4), para poder transmitir todo lo que se tecleaba en ellas [2].



Ilustración 2. IBM Selectric II (1971-1981)

En enero de 1983, mientras reparaban un Teletipo de la Embajada Francesa en Moscú, se descubrió un dispositivo que estaba diseñado para transmitir información hacia el exterior. Los franceses, aliados, alertaron a EEUU, y estos encontraron dispositivos similares en las máquinas de escribir de su Embajada en Moscú y el Consulado en Leningrado[[5]](#footnote-5). Tras este hallazgo, Lincoln Faurer[[6]](#footnote-6), NSA (Agencia de Seguridad Nacional, de EEUU) puso en marcha el “***Proyecto Gunman***” [3] [4] que tenía como objetivo detectar, analizar, y responder ante el implante de estos dispositivos, llamados “bugs”[[7]](#footnote-7). Estos se pueden considerar los precursores de los keyloggers que surgieron posteriormente.

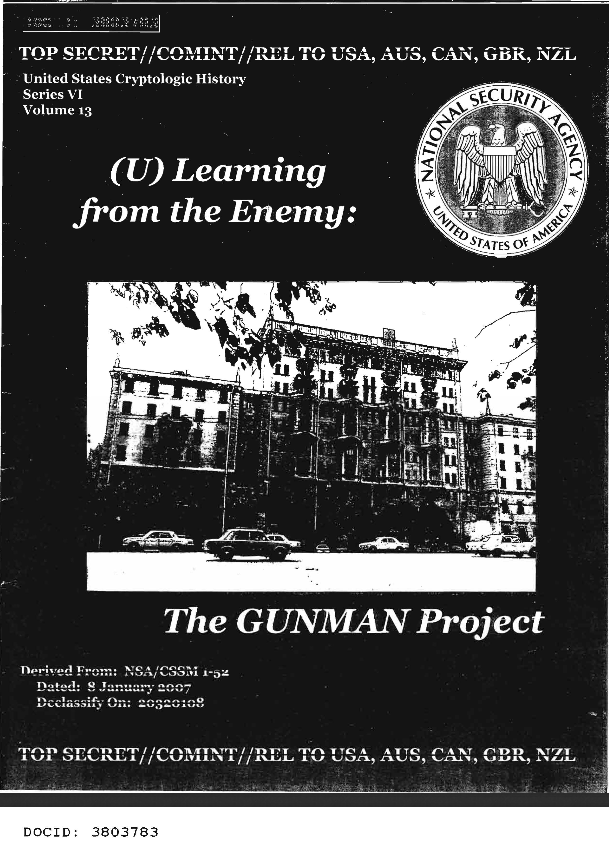


Ilustración 3. Portada del documento de la NSA, “The GUNMAN Project”.  
Desclasificado en Diciembre de 2011

El análisis complejo y exhaustivo de las máquinas de escribir no reveló la modificación de éstas hasta que se hicieron radiografías de toda la máquina y descubrieron la manipulación y los circuitos ocultos. La modificación de las máquinas era compleja, y virtualmente invisible e indetectable.

La modificación contenía circuitos integrados de última generación con una memoria central de 1 solo bit, ocultos dentro un soporte hueco en la parte inferior del teclado. Los datos se almacenaban en un buffer y posteriormente se enviaban en ráfagas cortas y de alta velocidad en las bandas de radio de 30, 60 y 90 Mhz. Estas frecuencias eran cercanas a la de estaciones de TV y ayudó a su ocultación y evitó la detección del envío de datos.

Para saber que teclas se estaban pulsando, disponía de un complejo sistema que detectaba la elevación, inclinación y rotación de la bola para saber qué carácter se estaba tecleando. La cantidad de caracteres existentes en la bola hacia que fueran necesarios 6 bits de datos, pero debido las limitaciones de la época[[8]](#footnote-8) los datos se comprimían en palabras de 4 bits. Se ignoraban teclas especiales como Mayúsculas, Espacio, Retroceso, Tabulador, Retorno de carro, Guiones y algunos caracteres podían ser ambiguos ya que se hacía una agrupación binaria de combinaciones de caracteres para poder tenerlo todo en 4 bits. Posteriormente, con los datos recuperados podían adivinar los caracteres ambiguos basándose en un análisis de frecuencias y con teorías de probabilidad.



Ilustración 4. Bola “pelota de golf” de la IBM Selectric, con los caracteres.

El 17 de noviembre de 1983, podemos situarlo como la fecha en la que se conoce el código fuente del primer keylogger creado por software por Perry Kivolowitz [5], el cual lo publicó [6] en los grupos de noticias de Usenet[[9]](#footnote-9) y dio motivo a una discusión entre usuarios para reforzar la seguridad del sistema operativo Unix.

Situándonos ya en el año 1999, se sabe que el FBI utilizó un keylogger [7] [8] para vigilar al jefe de La Cosa Nostra, Nicodemo Salvatore Scarfo, Jr., conocido como “Little Nicky”.

Este hecho es relevante por dos motivos. Fue el primer sospechoso criminal conocido en ser monitorizado con un keylogger. Y el keylogger, aparentemente, no era un desarrollo propio del FBI, sino que era un dispositivo comercial. Esto fue motivado porque Scarfo utilizaba un cifrado para proteger sus comunicaciones y el objetivo era conocer la clave de cifrado de estas. El FBI tuvo que entrar en la oficina de Scarfo 2 veces. Una para instalar el dispositivo, y otra para recuperar el contenido registrado.

En noviembre de 2001, el periodista Bob Sullivan de MSNBC reveló [9] que el FBI estaba desarrollando un software llamado “Magic Lantern” [10] cuya funcionalidad principal era la de keylogger y que además podían instalar y manejar en remoto. Un portavoz del FBI confirmó la existencia de este software en 2002, pero negó que se hubiera distribuido. En 2005 el FBI lo usó extensamente en muchos casos y en 2007 aparecieron los primeros documentos oficiales confirmando [11] su uso y ese mismo año pasó a formar parte de un nuevo software del FBI llamado CIPAV[[10]](#footnote-10). Magic Lantern se instalaba silenciosamente y no actuaba hasta que detectaba el uso de la herramienta PGP para poder capturar las claves de cifrado. Algunas compañías de antivirus cooperaron junto al FBI para que su herramienta no fuera detectada y se incluyó en listas blancas para no ser detectada por los antivirus. McAffee y Symantec fueron dos de estas empresas.

Pero no solo para cazar a mafiosos se usan los keyloggers. En este caso, fue el ladrón el que lo utilizó. En febrero de 2003, un estudiante Universitario, Douglas Boudreau, fue arrestado por haber instalado un keylogger en más 100 ordenadores del Boston College [12]. Recopiló información sobre unos 4800 profesores, empleados y estudiantes de dicha Universidad. Con los datos recopilados consiguió robar cerca de $2000.

En enero de 2004, un gusano, llamado MyDoom[[11]](#footnote-11) provocó una gran epidemia convirtiéndose en el gusano de correo electrónico que más rápido se propagó. El hecho es que contenía un keylogger para capturar los números de tarjetas de crédito.

Con la llegada del nuevo milenio parece ser que los keyloggers se empezaron a popularizar entre los delincuentes, y a finales de 2004, un grupo de ciberdelincuentes formado por jóvenes rusos y ucranianos, enviaron múltiples correos electrónicos a clientes de bancos en Francia. Estos correos electrónicos contenían un archivo adjunto con un keylogger o enlaces a sitios web que contenían dicho keylogger. Los usuarios eran engañados mediante métodos de ingeniería social, y una vez instalado el malware[[12]](#footnote-12), este quedaba oculto e inactivo y solo entraba en funcionamiento al detectar que el usuario accedía al sitio web de la entidad bancaria, para proceder en ese momento a registrar las contraseñas y los códigos bancarios.

En febrero de 2005, un empresario de Florida, Joe Lopez, presentó una demanda contra Bank of America por que le habían robado $90,000 de su cuenta y habían sido transferidos a una cuenta de Letonia. En realidad, el ordenador de Lopez había sido infectado por el virus “Backdoor Coreflood” que abría una puerta trasera al ordenador e incorporaba un keylogger. Los hackers consiguieron así los datos de acceso a las cuentas bancarias. Joe Lopez perdió el juicio porque consideraron que había sido negligente y no había tomado precauciones al manejar sus cuentas bancarias por internet.

En mayo de 2005 la policía israelí detuvo a un matrimonio en Londres que elaboraba programas maliciosos a medida para realizar espionaje industrial. Evidentemente, con la funcionalidad de keylogger. Estos programas eran vendidos a compañías israelíes que los utilizaban para espiar a sus competidores. El creador, Michael Haephrati [13], fue sentenciado a dos años de prisión, pero mantiene su compañía de seguridad, “Target Eye Limited”, donde anuncia sus productos de “vigilancia y monitorización para capturar la actividad de ordenadores remotos”. Lo curioso de esta historia, es que su mujer, Ruth, fue condenada a cuatro años de prisión (dos más que su marido) por la venta del software.



Ilustración 5. Muestra de una sección del perfil de LinkedIn de Michael Haephrati.   
Detenido en 2005 por la creación de un keylogger.

Con el auge de la telefonía móvil y los smartphones era inevitable que los keyloggers también evolucionaran y migraran a nuevas plataformas. Así pues, en 2011 ya aparecían noticias que evidenciaban que aparecían nuevas técnicas para espiar lo que los usuarios tecleaban en los smartphones. Hay que tener en cuenta que no existe un teclado físico como tal, pues son toques en la pantalla sobre un teclado virtual y el funcionamiento es muy diferente al de un teclado tradicional.

El profesor Hao Chen [14] de la Universidad de California, Davis, presentó un estudio [15] en la que una aplicación podía registrar las pulsaciones y saber que estaba tecleando el usuario a través de una técnica diferente a la que usan los keyloggers tradicionales.

El éxito de los keyloggers para llevar a cabo ciberdelitos estaba consolidado y con éxito asegurado. Tal es así que en el año 2013 se empezó a gestar el mayor ataque de robo financiero dirigido contra un banco. En este caso el keylogger era una pieza clave del malware pero que además estaba dotado de muchos más elementos y funcionalidades de malware, como captura de pantalla y comandos remotos. La operación, llamada “Carbanak”, culminó con un robo de 1000 millones de dólares.

Los estudiantes Universitarios tampoco han estado exentos del uso de keyloggers. En el año 2015, un estudiante de la Universidad de Birmingham utilizó un keylogger por hardware para robar las contraseñas de los profesores y cambiar sus calificaciones. El dispositivo no lo descubrieron hasta que no quitaron la carcasa protectora de la parte trasera del ordenador.

Y de nuevo, en octubre de 2017 un caso similar. Otro estudiante, esta vez de la Universidad de Kansas había sido expulsado por haber usado un keylogger, también de hardware, y haber usado los datos adquiridos para acceder al sistema de calificaciones y cambiarse las notas. Dos casos con exactamente el mismo objetivo, el mismo método, el mismo entorno, pero a una distancia de casi 7000 Kms y dos años de diferencia.

A finales del año 2017, el experto en seguridad Michael Myng detecto un problema en el software controlador del teclado del fabricante Synaptics instalado en algunos modelos de ordenadores portátiles de la marca HP. Dicho controlador tenía la capacidad de poder registrar los códigos de escaneo del teclado sobre un archivo de log mediante las herramientas WPP Tracing de Microsoft. Esto abría la posibilidad de que un malware pudiera activar un keylogger en dichas maquinas con solo cambiar unos valores en el registro de Windows. A pesar de que el keylogger no estaba activo, el riesgo existía y obligó al Synaptics y HP a actuar lo antes posible para corregir el driver y publicar una actualización de este para todos los modelos afectados.

En diciembre de 2017, la empresa de seguridad SUCURI, alertó en su blog [16] de la presencia de miles de sitios WordPress[[13]](#footnote-13) infectados con un keylogger desarrollado íntegramente en JavaScript[[14]](#footnote-14). En enero de 2018 apareció una evolución [17] de este.

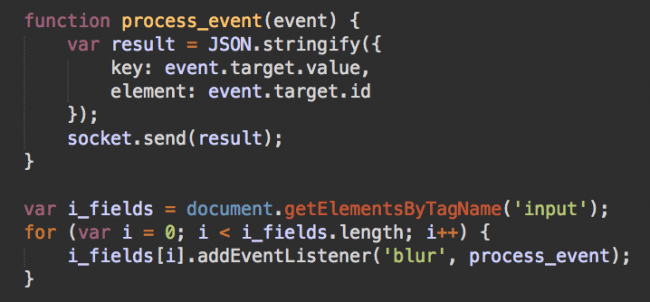


Ilustración 6. Código del keylogger inyectado después de su descifrado.

Un descubrimiento reciente, en Julio de 2019, del experto en seguridad Marcus Mengs, ha puesto al descubierto un fallo de seguridad en los dispositivos inalámbricos (teclados y ratones) del fabricante Logitech que utilizan la tecnología Unifying para usar un solo receptor para los mismos dispositivos. Este fallo de seguridad demuestra como un usuario que se encuentre en las inmediaciones, puede capturar todo lo que se teclea en el ordenador que usa los teclados de Logitech e incluso enviar comandos personalizados al equipo.

Hay muchos más casos, y los mostrados aquí son una pequeña selección, quizás los más relevantes, para podernos hacer una idea de los usos y evolución de los keyloggers.

De estos casos se puede vislumbrar una cierta evolución que empieza en el espionaje entre estados o poderes políticos, pasando por el uso para la vigilancia policial, llegando finalmente al uso extenso en ciberdelincuencia, para el robo de credenciales y datos bancarios y el espionaje industrial, y otros delitos menores como los casos de los estudiantes para modificar sus calificaciones académicas.

En cualquier caso, existe multitud de empresas que venden su producto de monitorización (keylogger y muchas más funcionalidades) como algo aparentemente legal y que se puede usar desde entornos laborales (para monitorizar el uso de los equipos por los trabajadores) hasta el control parental (padres que quieren tener un control de lo que escriben sus hijos en redes sociales).

Siendo estos dos, laboral y parental, entornos donde actualmente se usan los keyloggers, aunque en menor medida, y en el entorno de la ciberdelincuencia donde más se extiende como una característica de los complejos malware que se están distribuyendo.

Los keyloggers hardware son actualmente pequeños dispositivos usb que además disponen de conexión wifi y permiten monitorizar lo que se está escribiendo en tiempo real.

Los keyloggers software suelen formar parte de un paquete de malware, con un claro objetivo de robo de datos, pero que son solo una de las características de las que está compuesto dicho malware, que dispone de un panel de Mando y Control con opciones de captura de pantalla, ejecución remota de datos, envío de documentos, etc.

# Aspectos Legales

Como se ha comentado en el punto anterior, existen keyloggers (o software que lo incorporan) que se veden comercialmente y, por lo tanto, existen cierta incertidumbre sobre su legalidad.

Hemos querido conocer que dicen tanto la legislación Española como la Europea, para ver en qué situación se encuentra este tipo de dispositivos y/o programas. Y también hemos consultado a dos profesionales en materia jurídica. Un abogado generalista y otro especializado en derecho penal y delitos informáticos.

En Europa existe la **DIRECTIVA 2013/40/UE DEL PARLAMENTO EUROPEO Y DEL CONSEJO** [18], publicada en Agosto del 2013 relacionada con los ataques a los sistemas de información y por ende con los ciberdelitos. En su artículo 6 hace referencia a la “**Interceptación ilegal**” que es el objetivo directo de los keyloggers.

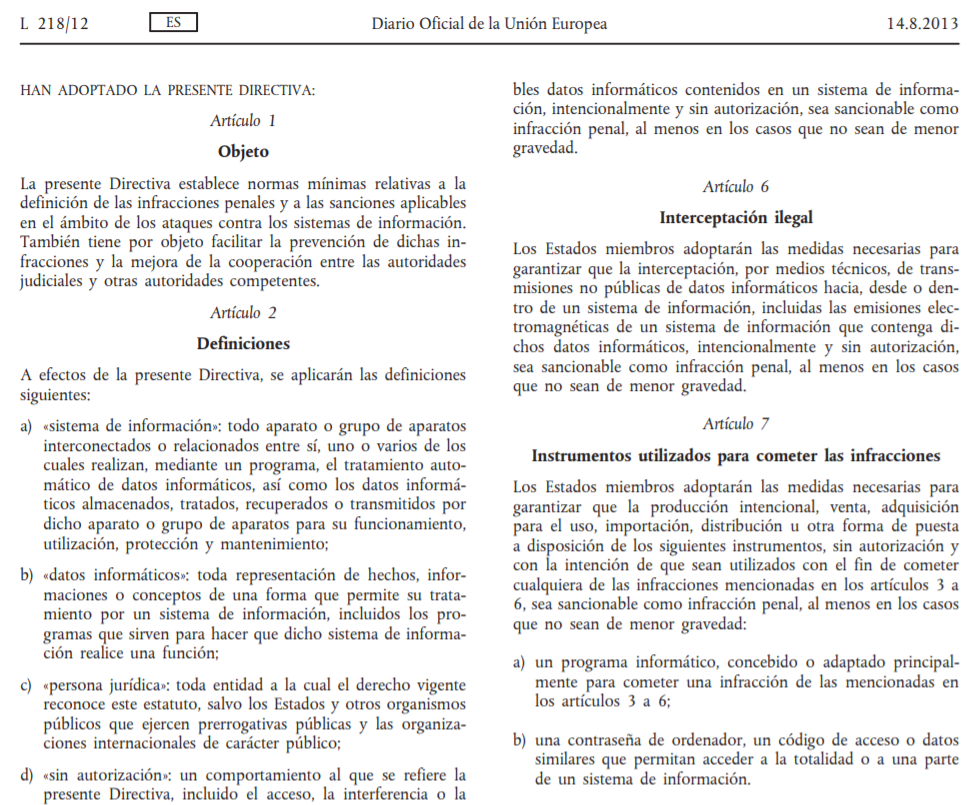


Ilustración 7. Diario Oficial de la Unión Europea.   
DIRECTIVA 2013/40/UE. 12 agosto 2013.   
Extracto del contenido relativo a los ataques contra los sistemas de información.

Por otro lado, la legislación española, en su Ley Orgánica 1/2015 [19], de 30 de marzo, por la que se modifica la Ley Orgánica 10/1995, de 23 de noviembre, del Código Penal, en su apartado ciento seis dice:

“El que, para descubrir los secretos o vulnerar la intimidad de otro, sin su consentimiento, se apodere de sus papeles, cartas, mensajes de correo electrónico o cualesquiera otros documentos o efectos personales, intercepte sus telecomunicaciones o utilice artificios técnicos de escucha, transmisión, grabación o reproducción del sonido o de la imagen, o de cualquier otra señal de comunicación, será castigado con las penas de prisión de uno a cuatro años y multa de doce a veinticuatro meses.”

Pero, las dudas respecto al uso de estos dispositivos, así como también las circunstancias en las que muchas veces se justifica su uso, hacen que el ciudadano no perciba la situación de estar vulnerando la ley o cometiendo un acto ilegal.

Solo hay que hacer una búsqueda en internet, en Google, o eBay o Amazon (por poner unos ejemplos) y escribir el término “*keylogger*”, y podremos ver como hay decenas de sitios que venden el producto (tanto en España, como en el resto de Europa) y parece ser que la ley hace caso omiso, o que estos negocios se encuentran en un estado de “vacío legal”.

Hay sitios web que lo venden a todo el mundo, y se escudan en que es el usuario el responsable de su uso y que debe consultar si las leyes locales de su país lo permiten. Aunque justifican que puede usarse para el control familiar.

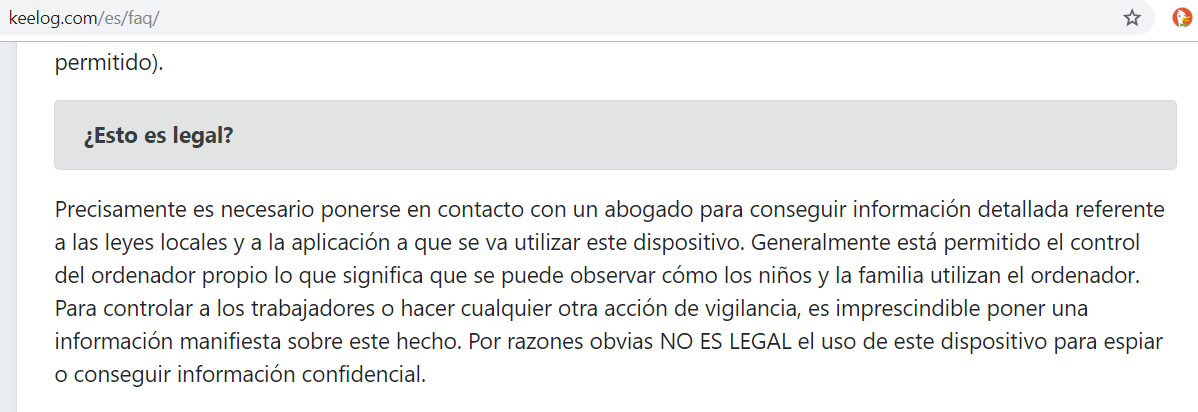


Ilustración 8. Ejemplo de aviso legal sobre el uso del Keylogger, de un vendedor internacional.

Pero, aun así, después de leer la directiva Europea [18] y la Legislación Española [19], a un ciudadano le pueden surgir dudas. ¿es siempre su uso ilegal? ¿existen excepciones? ¿si es ilegal, porque se permite su venta?

Para intentar despejar estas dudas y otras hemos contactado con dos abogados que nos ayuden a interpretar lo que dice la ley, y lo que es más importante, como la interpretan dependiendo de si están especializados en el tema o no.

Además de preguntar en que situación nos sitúa la ley, tanto en Europa como en España en relación a dicha directiva y a Ley Orgánica Española, también les hemos hecho unas preguntas con unos ciertos supuestos de situaciones de uso del keylogger.

Los supuestos en los que se instala el keylogger son los siguientes:

1. Un padre/madre de familia, que tiene un ordenador en casa y desea monitorizar que escriben sus hijos en internet, redes sociales, que buscan, etc.
2. Un marido/esposa, instala el dispositivo (o programa) en el teléfono u ordenador del conyugue para saber que escribe y con quien.
3. Un empresario lo instala en los ordenadores de su empresa porque tiene sospechas de que algún trabajador está filtrando información confidencial de la empresa.
4. La policía lo instala en el ordenador de un sospechoso para seguir sus actividades, supuestamente ilegales.
5. Un ciberdelincuente[[15]](#footnote-15) (consideramos que no es correcta la acepción «Hacker»[[16]](#footnote-16)) instala el keylogger en el ordenador de una víctima, empresa o particular, con el objetivo de obtener información privilegiada, privada o confidencial. Como, por ejemplo: nombres de usuario, contraseñas, datos bancarios o de tarjeta de crédito, etc.

Llegados a este punto nos pueden parecer evidentes y lógicas las respuestas a algunas de estas preguntas, pero veamos que nos responden los profesionales:

En el caso del abogado generalista, su respuesta es que, en el primer punto, si el hijo es menor de edad, no se está cometiendo ninguna ilegalidad ya que el control parental sobre los hijos está justificado. Sobre el segundo punto no tenemos respuesta pues no se incluyó en las preguntas formuladas.

En lo que respecta al uso empresarial, considera que se podría tipificar como una “intrusión ilegítima en el derecho de las comunicaciones y privacidad” siempre que el empresario no hubiera avisado previamente con normas en la empresa.

En el caso del uso Policial, considera que no hay delito si la policía tiene autorización judicial, salvo que esta persona tenga reconocido un derecho a la privacidad en su actividad, por ejemplo, los abogados con sus clientes. Y en esta afirmación hace referencia al “Caso Garzón” en que se imputó al juez Garzón por haber intervenido las comunicaciones de los Abogados con sus clientes sobre los que tenía una investigación en curso.

Sobre el caso del ciberdelincuente, su respuesta es muy ambigua, y considera la instalación del software como un medio y no se puede considerar como un delito en sí mismo y que el delito principal seria la revelación de secretos o la estafa en el caso de obtener números de cuenta o contraseña y, por lo tanto, si estos últimos no se han podido llevar a cabo, no existe delito.

Para el abogado Penalista y experto en Delitos Informáticos las respuestas versan algo diferentes y más extensas:

En el caso de los padres, incide en que no tiene por qué ser legal, ya que la Convención de los Derechos del Niño promulga los derechos de estos, y entre ellos se encuentra el derecho a la intimidad. Pero, por otro lado, hay que tener en cuenta que los padres están obligados por Ley a velar por sus hijos. Llegamos pues a una disyuntiva, en la que además hay que considerar que, en España, a partir de los 14 años de edad, aun siendo menores tienen derecho a hacer uso de sus datos personales y los padres no los pueden obligar en un sentido u otro.

Por lo tanto, desde el punto de vista jurídico, la instalación de un keylogger, se considera un programa espía que vulnera su intimidad. Y un procedimiento judicial que implicase la obtención de información del menor a través de estos instrumentos, se podría considerar perdido a menos que se pudiera justificar una situación de extrema necesidad en la que se prioriza la supervivencia del menor o la evitación de un peligro inminente.

Lo ideal es que los padres/madres pacten con los hijos acerca de la instalación de un software de estas características, y el argumento es que el adulto (padre o madre) es el titular del teléfono móvil (ordenador o línea de conexión a internet) y quien paga la factura, por tanto, es suyo y lo cede en uso a su hijo. Ya que cualquier delito cometido a través de ese dispositivo –ordenador o móvil- (ej. compartir pornografía infantil, bulling, descargas ilegales, etc.) la responsabilidad es del titular de la IP asociada al dispositivo.

En el caso del conyugue es claro y tajante, ya que tratándose de adultos e independientemente de que haya o no una situación afectiva, es considerado una “interceptación de las comunicaciones y violación de la intimidad”, y puntualiza que “*la intimidad, es un grado más que la privacidad*”.

En relación con el uso empresarial, nos hace hincapié en que el Estatuto de los Trabajadores, en su artículo 20, reconoce en el empresario su derecho de control de la actividad de los empleados. Pero, para que sea legal utilizar la información obtenida por un keylogger, el empleado debe ser advertido en el mismo contrato de cuáles son sus obligaciones respecto al uso de los dispositivos de la empresa y cuáles son los derechos del empleador / empresario respecto al control de su actividad.

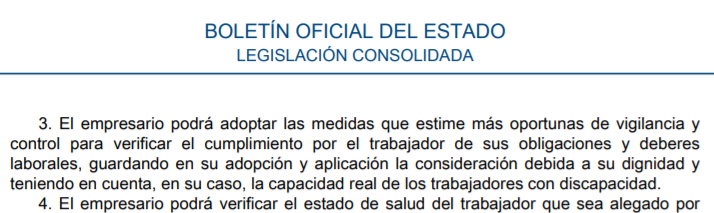


Ilustración 9. Extracto del Artículo 20 del Estatuto de los Trabajadores.

En el ámbito policial nos informa de que la ley de enjuiciamiento criminal desarrolla en el nuevo y extenso artículo 588 la justificación de los seguimientos de control remoto por parte de la policía. Tiene que estar autorizado por un Juez previa justificación de qué quiere obtenerse o, en su caso, por urgencia y necesidad, podría hacerse sin autorización, pero en 24 horas deberá confirmarse por el Juez la acción que se ha llevado a cabo. Los seguimientos tienen que estar limitados a un periodo de tiempo y, en el informe, debe descartarse cualquier información de terceros que no tenga nada que ver con el caso.

Y ya, por último, en el caso del uso del keylogger por parte de un ciberdelincuente, se hacen distinciones de 4 etapas:

1. Desarrollo del keylogger. Según el artículo 197 ter, tener o crear un software que su único destino es el de producir una acción ilícita es considerado delito y, por lo tanto, castigado con pena de prisión. La única justificación que puede tener una persona para tener en su poder un programa de estas características es que su actividad empresarial consista en el Hacking Ético y, por tanto, mediante contrato, se utilice mediante el **«Red Team»[[17]](#footnote-17)** este tipo de herramienta de intrusión.

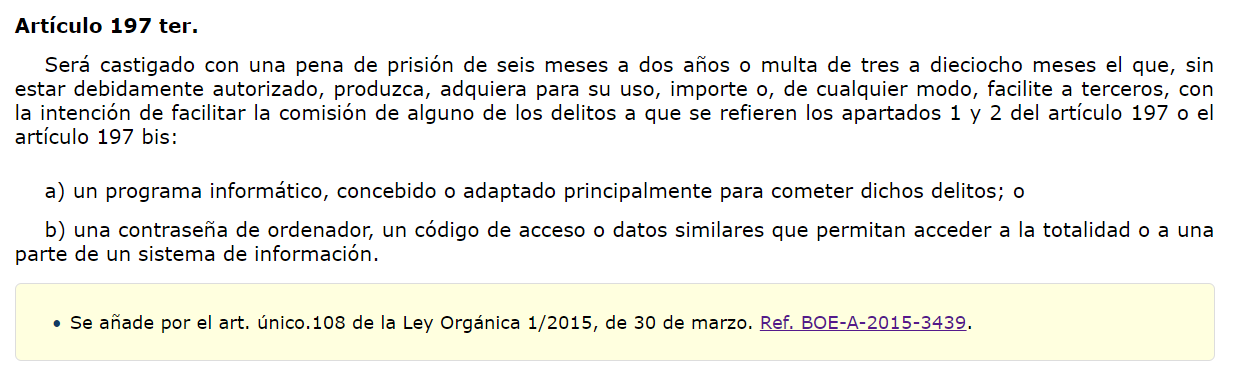


Ilustración 10. Extracto del artículo 197 ter, de la Ley Orgánica 1/2015, de 30 de marzo

1. Envío del keylogger. Es probable que el envío se haga mediante «**phishing**»[[18]](#footnote-18), es decir mediante engaño, para que el usuario se instale el programa. Por lo tanto, habría un agravante de engaño en caso de condena.
2. Obtención de credenciales de acceso bancarias. Indicado también en el artículo 197 ter, citado anteriormente.
3. Uso de las credenciales para lucrarse con ello. Sufrirá un agravante contemplado en el artículo 197.6 y se le añadirá un concurso de delitos al haber robo de dinero.

Llegados a este punto, después de las opiniones de los profesionales y tras la lectura de diversos documentos nos siguen quedando un par de dudas:

* Si desarrollar (y vender) un keylogger es ilegal, ¿por qué existen tantas empresas que los venden por internet?
* Por otro lado, la ley habla en todo momento de “interceptación de comunicaciones”, entendiendo esto como la comunicación entre dos sistemas o dos personas. Pero un keylogger intercepta pulsaciones de teclas del teclado que se guardan en el mismo ordenador. Si estas pulsaciones se guardan en bruto en un archivo de texto, sin tener en cuenta a quien van dirigidas o con que aplicación se escriben, y dado que solo se registra lo que escribe un usuario, ¿podría entenderse como una interceptación de comunicaciones?

# Tipología

Como se ha comentado anteriormente, podemos dividir los keyloggers en dos grandes grupos: Hardware y Software.

La funcionalidad principal, el registro de teclas, la cumplen a la perfección los dos tipos, pero cada uno tiene unas ventajas en inconvenientes que resumimos en la siguiente tabla:

|  |  |  |
| --- | --- | --- |
| Tipo | Ventajas | Inconvenientes |
| Hardware | * No se necesita acceso al sistema operativo, y por lo tanto no se requieren credenciales de usuario. * Los datos se almacenan dentro del propio dispositivo no dejando rastro en el equipo. * No se requiere drivers, ni software adicional que se tenga que instalar en el sistema. * Registra las pulsaciones de teclas para todos los usuarios que usen el equipo. * Indetectable para los antivirus o escáneres de seguridad. * Rapidez de instalación: Plug&Play. * Compatibilidad para diferentes sistemas: MS-DOS, Windows, Linux, Mac. | * Visible externamente por el usuario, por lo tanto, detectable si inspecciona las conexiones. * No tiene posibilidad de distinguir que usuario ha escrito el texto almacenado en caso de que el ordenador lo usen múltiples usuarios. * No tiene posibilidad de saber en qué ventana de aplicación se estaba escribiendo el texto almacenado. * Se requiere tener un dispositivo físico por cada máquina a controlar. * Para ver los datos hay que acceder físicamente al equipo para extraer el dispositivo, a no ser que este tenga conexión WiFi. |
| Software | * Permite identificar que usuario ha escrito el texto almacenado. * Se puede saber en qué ventana de aplicación se estaba escribiendo el texto almacenado. * Versátil y actualizable. Puede tener más funcionalidades que complementen su utilidad. * Difícilmente detectable por el usuario medio. | * Se requiere acceso a la cuenta de usuario para instalarlo. * Requiere versión específica para el sistema operativo que se vaya a usar. * La información se suele guardar en el propio equipo, por lo que pueden quedar rastros de su existencia. * Para acceder al contenido registrado se debe de enviar a un servidor externo o el usuario debe acceder a la máquina para ver el contenido. |

A modo de resumen, el keylogger hardware podríamos decir que es algo del estilo “*plug & play*”, sin necesidad de acceso al sistema, pero que no pasa desapercibido a no ser que el ordenador este dentro de un mueble y no tenga sus conexiones a la vista. Es un dispositivo que no va a instalarse nunca la propia víctima por error, ya que es imprescindible su colocación entre ordenador y teclado, y difícilmente un usuario lo va a poner voluntariamente.

Los keyloggers software suelen estar desarrollados en lenguajes de bajo nivel, o con un grado de interacción muy estrecha con las API[[19]](#footnote-19) del sistema operativo, por lo que el software está diseñado específicamente para una plataforma, requiere tener acceso al sistema para instalarlo, aunque aquí se pueden emplear métodos de engaño para que sea el propio usuario que instale el software o inicie otro proceso que haga la instalación de manera silenciosa o desatendida.

Hoy en día existen lenguajes de alto nivel que facilitan y simplifican esta comunicación con el sistema con lo que la necesidad en sus inicios de lenguajes de alto nivel se ha ido reduciendo. De hecho, en el apartado 7 (Variantes de Keyloggers) se hace mención del caso específico de los dispositivos móviles y de un parte de tipos de keyloggers hechos con lenguajes de alto nivel.

# Como captura los datos

Para entender la captura de datos de los keyloggers las diferenciaremos tratándose de un keylogger basado en Hardware, Software o con métodos mixtos.

## Captura en keylogger basado en Hardware

Los keyloggers hardware, son dispositivos que se conectan entre el teclado y el ordenador. Antiguamente usando el puerto PS/2[[20]](#footnote-20) y en la actualidad el puerto USB[[21]](#footnote-21).

Se trata de pequeños circuitos compuestos por un microcontrolador (en el esquema de ejemplo se muestra un Atmel AT89C2051) [16] que reciben el código de la tecla, lo traduce al carácter (o combinación de teclas correspondientes) y lo graban en un archivo de texto dentro de la memoria (una EEPROM AT24C512 en este esquema) del propio dispositivo. Al mismo tiempo, reenvía dicho código de tecla al ordenador, siendo una operación totalmente transparente e indetectable tanto por el sistema operativo como por cualquier otro programa que vele por la seguridad del sistema, como pueda ser un Antivirus.

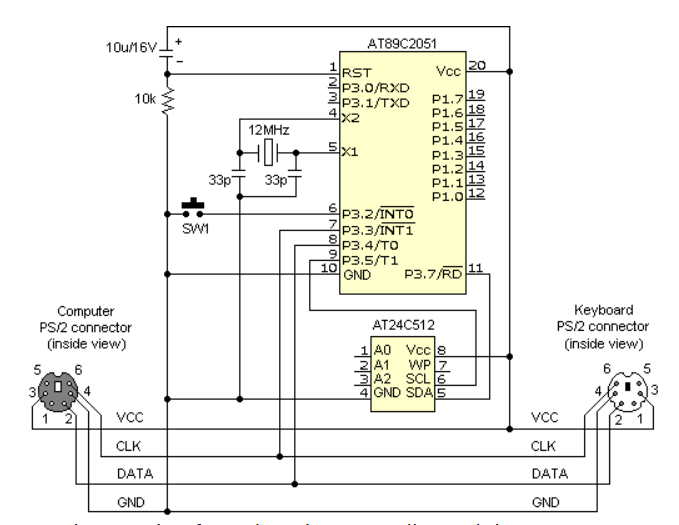


Ilustración 11. Esquema de circuito de keylogger básico usando la antigua conexión PS/2.

Este archivo de texto almacenado dentro del keylogger puede recuperarse posteriormente, desde el mismo ordenador, o desde otro (conectando el mismo keylogger) a través de una combinación secreta de teclas (normalmente personalizable) a modo de contraseña, que activaran una memoria de almacenamiento masivo (pendrive) en el sistema, que permitirá acceder a dicho archivo y ver su contenido.

Existen modelos más avanzados, que disponen de un chip de comunicaciones inalámbricas [17]. Podría ser Bluetooth o WiFi, y que permitirían captar los datos en tiempo real desde otro ordenador que estén dentro de las inmediaciones y rango inalámbrico permitan, o bien conectarlo a una red inalámbrica y ser accesible desde cualquier equipo que esté conectado en la red local.

## Captura en keylogger basado en Software

El keylogger basado en software se podría subdividir en diferentes categorías desde una perspectiva técnica pues no hay una manera única de hacer la captura de las pulsaciones, y a medida que evolucionan los sistemas o aparecen nuevas versiones pueden surgir nuevos métodos.

### En una Máquina Virtual

El keylogger podría residir dentro de una máquina virtual creada por el propio malware, de manera que permanezca oculto al sistema. Existe una prueba de concepto [22] llamada Blue Pill[[22]](#footnote-22) (píldora azul) que se presentó en la conferencia Black Hat del año 2006.

### Basado en la API.

En este caso se utiliza el mecanismo de “*hooks*” (ganchos) y es quizás el método más ampliamente usado, probablemente por su relativa facilidad de implementación. Los hooks permiten definir funciones que reciben las notificaciones de eventos concretos del sistema. En este caso, se reciben las notificaciones de los eventos de teclado, y dichas funciones, llamadas filtros, son ejecutadas tras cada evento y podrían comportarse de diferentes formas ante cada evento, añadiendo información, registrando datos o incluso eliminando dicho evento o ignorando la acción.

Las funciones filtro se vinculan a un tipo de notificación de evento, por ejemplo, KeyUp (cuando una tecla pasa del estado de pulsada a no pulsada). La vinculación se lleva a cabo mediante una llamada a la función **SetWindowsHookEx** [22] de la API de Windows y se puede desvincular o eliminar mediante la llamada a **UnhookWindowsHookEx** [23].

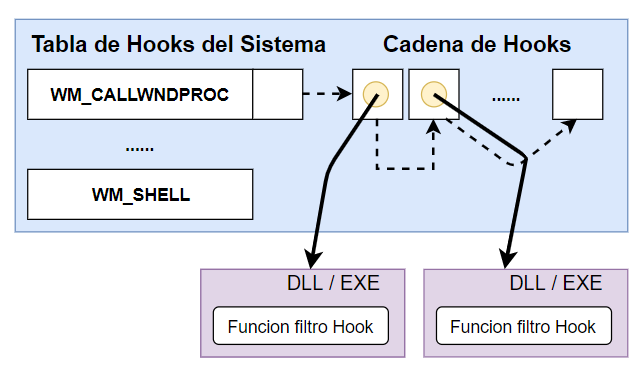


Ilustración 12. Esquema de funcionamiento del mecanismo de *hooks* en Windows.

Para evitar conflictos, si hay varias funciones filtro vinculadas a un mismo evento Windows establece una cola de funciones. Por lo que podría darse el caso de que existieran dos keyloggers diferentes en el sistema que recibieran los eventos del teclado.

El creador del keylogger solo debe de crear su función de filtro, y añadirla a la cadena de hooks mediante la función SetWindowsHookEx [22] de la API, y asegurarse de que su función se mantiene en memoria mediante un programa oculto que se cargue al inicio del sistema.

### Como rootkit

Implementado como un rootkit[[23]](#footnote-23) dentro del Kernel[[24]](#footnote-24), funcionando como si fuera un controlador de dispositivo del sistema [20]. Reside a nivel de núcleo, por lo que es muy difícil de detectar y además tiene acceso sin restricciones al sistema o al hardware. La creación de este keylogger implica que el software se aprovecha de otras vulnerabilidades del sistema (en el caso de un malware) para que con privilegios de administrador pueda instalar el software en el sistema que será reconocido como un controlador.

### Basado en Javascript

Similar al sistema del “hook” de la API, se utiliza sobre un sitio web vulnerable que ha sido comprometido, donde se inyecta el código JavaScript el cual escucha los eventos del teclado. La ventaja de este sistema es que el keylogger se ejecuta en una máquina cliente, pero la que se infecta con el código es el host que alberga la página web, con lo que si se consigue la infección de un sitio web popular se podrían captar las pulsaciones de teclado de cientos o miles de equipos.

En este caso, vemos un ejemplo [21] de keylogger en javascript, de manera que si se consigue inyectar este código en una página web se podrían captar las pulsaciones de las teclas del usuario que visite la página y enviarlas a un servidor externo, el cual las reciba y las registre en un archivo.

En la siguiente imagen se muestra el código encargado de ello.

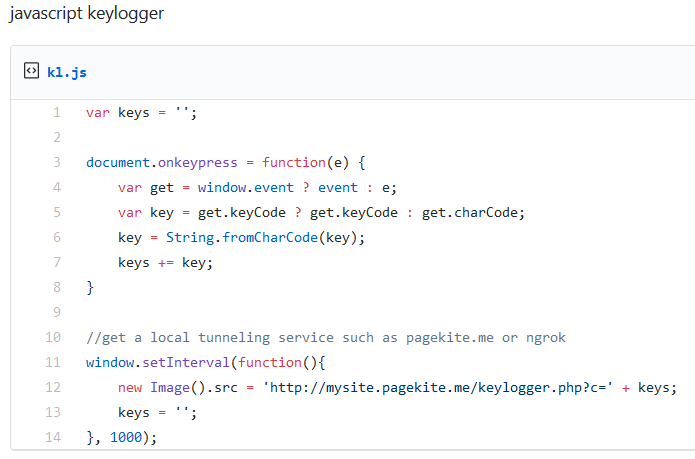


Ilustración 13. Código fuente de un keylogger en Javascript.

En el lado del servidor también existe un código que recibe dichas teclas y las registra en un archivo, el cual, si cabe, es mucho más simple y corto, y en este caso es en lenguaje PHP.



Ilustración 14. Código fuente en PHP del backend del keylogger que registra las pulsaciones.

### Inyección de memoria.

Un método poco común, es el basado en Inyección de memoria. Fue utilizado por los malware Zeus[[25]](#footnote-25) y SpyEye[[26]](#footnote-26). Esta técnica está dirigida exclusivamente a los navegadores Web y se basa en la alteración de unas tablas de memoria asociadas con el navegador en las que se inyecta el código. Esta técnica usada es conocida también como “Man in the browser” (MiTB[[27]](#footnote-27)). En este caso el método se aprovecha de la vulnerabilidad del navegador del usuario para infectarlo con el malware.

### Event Tracing for Windows

El Event Tracing for Windows (ETW) [22] es un mecanismo asíncrono de depuración del kernel integrado en Windows, que generalmente se utiliza para ayudar a los administradores y desarrolladores a solucionar problemas y medir el rendimiento del sistema y las aplicaciones. Está integrado y habilitado de manera predeterminada en Windows 2000, pero se mejoró sustancialmente su funcionalidad desde Windows 7. Con estas funciones de depuración también se podrían capturar pulsaciones del teclado, aunque estaría limitado a dispositivos USB, por lo que no funcionaría en equipos portátiles.

### Basados en CSS

Un desarrollador llamado Max Chehab, presentó una prueba de concepto de keylogger en CSS en su repositorio de GitHub [27] en el que utilizando selectores de atributos CSS podía cargar un recurso de un servidor externo, que de esta manera podría ser diferente para cada tecla, creando de esta manera un sofisticado keylogger.

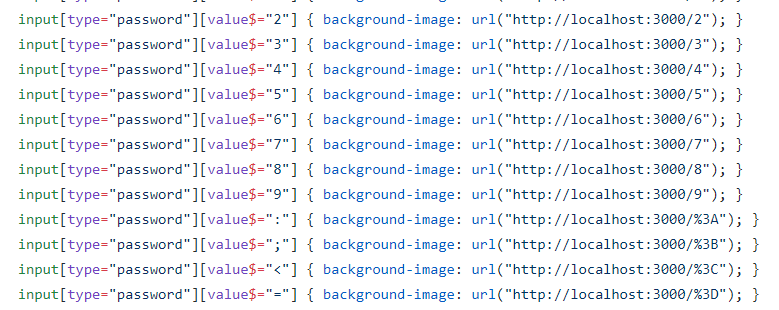


Ilustración 15. Muestra de parte de archivo CSS para crear un keylogger.

### Dispositivos móviles

El teclado físico no existe en los dispositivos móviles, smartphones, tabletas y otros elementos con pantalla táctil. Estos dispositivos que funcionan mayoritariamente en los entornos Android o IOS, tienen un teclado virtual en pantalla.

Se ha mencionado en el apartado de Historia y Estado del Arte, que en año 2011 un profesor de una Universidad de California [14] presentó un estudio [15] de cómo podía capturarse lo que escribían los usuarios. ¿Pero cómo lo hacían? En lugar de usar los métodos tradicionales a través de las funciones del teclado se basaban en los movimientos y vibraciones a detectados por los acelerómetros del dispositivo móvil. Con un complejo sistema matemático de cálculos y predicciones asociadas a cada tecla la aplicación desarrollada, llamada TouchLogger, llegaba a una precisión por encima del 70% en pruebas usando solo el teclado numérico.

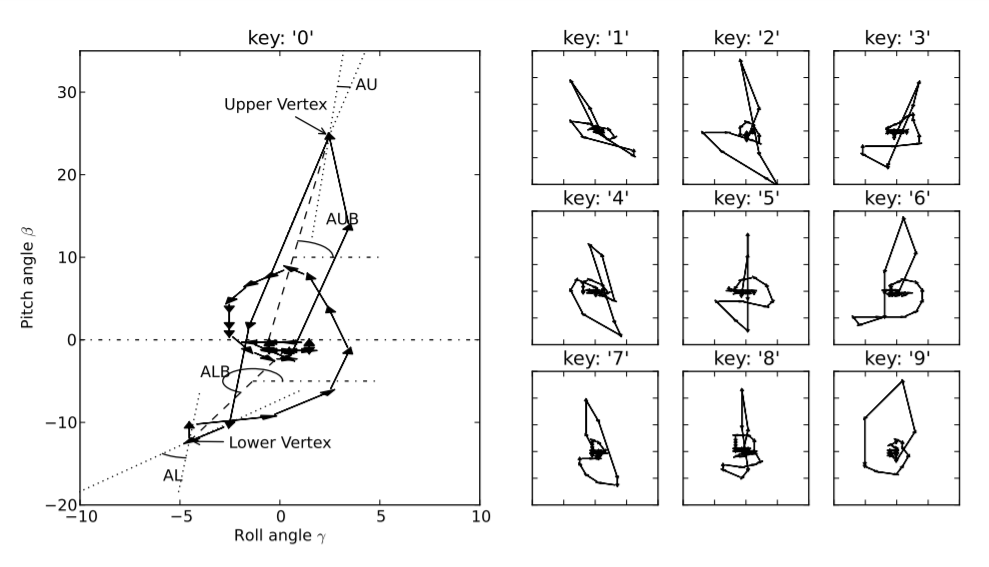


Ilustración 16. Patrones típicos de ángulos y balanceo cuando se presionan diferentes teclas de dígitos.

Esto es especialmente relevante porque un programa que pueda buscar un malware de keylogging no supondrá que el malware esté en la detección de movimientos de los acelerómetros que, por otro lado, es usado por muchas aplicaciones legítimas en los dispositivos móviles.

No obstante, también existe la propia API del sistema que permite gestionar eventos de control de teclado. Reescribiendo [28] el método ***onKey()*** de la clase ***Softkeyboard*** incluido en el Android SDK se le puede añadir la capacidad de keylogger a cualquier aplicación desarrollada para esta plataforma.

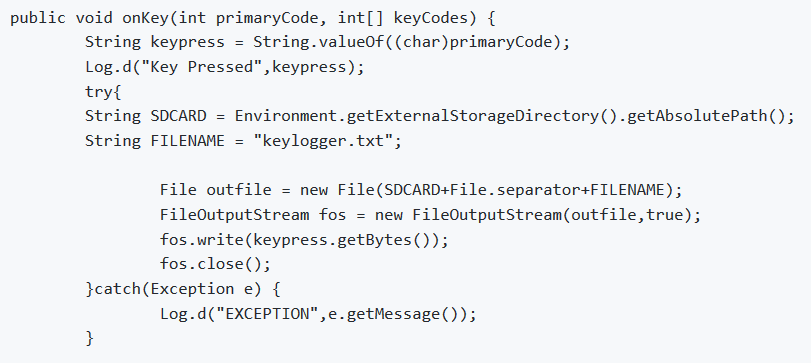


Ilustración 17. Ejemplo de método modificado en Android SDK para crear un keylogger.

## Otras técnicas usando canales laterales

Existen otras técnicas, no muy extendidas, pero que conviene mencionar. Estas técnicas son las que utilizan lo que podríamos denominar «Canales Laterales», y son las que obtiene información por señales generadas a partir de efectos secundarios que no se suelen tener en cuenta.

Estos sistemas, suelen usar una combinación de Hardware y Software para poder realizar la captura de datos.

### Criptoanálisis acústico.

Es una técnica que aprovecha los sonidos generados por el teclado del ordenador y los sonidos de los componentes internos para deducir que teclas se están tecleando. Existen dos *papers,* uno del «IBM Almaden Research Center», titulado “Keyboard Acoustic Emanations” [23] y publicado el año 2004, y otro publicado en el año 2005, por la Universidad de Berkeley, titulado “Keyboard Acoustic Emanations Revisited” [24] que hacen referencia a estas técnicas.

Cada tecla del teclado crea una firma acústica sutilmente diferente cuando se pulsa. Entonces es posible identificar qué firma de pulsación de tecla se relaciona con cada carácter del teclado a través de métodos estadísticos como el análisis de frecuencia. La frecuencia de repetición de firmas de pulsaciones de teclas acústicas similares, los tiempos entre diferentes pulsaciones de teclado y otra información de contexto, como el idioma probable en el que el usuario está escribiendo. Para que este método sea efectivo se requiere previamente una muestra amplia de datos, más de 1000 pulsaciones de teclas, de la que el análisis posterior podrás asignar los sonidos correspondientes a cada tecla.

### Emisiones electromagnéticas.

También a distancias cortas, una distancia máxima de 20 metros, se pueden captar las emisiones electromagnéticas de un teclado sin tener que estar conectados físicamente a él. Esto es debido a que habitualmente, para reducir costes, no se pone un blindaje electromagnético en estos dispositivos por su baja emisión y los costes que comporta.

Existe un paper titulado «Compromising Electromagnetic Emanations of Wired and Wireless Keyboards» [25], presentado en el «18th USENIX Security Symposium 2009» en el que midiendo las emisiones electromagnéticas en un espectro amplio de frecuencias y con un analizador de espectral se puede llegar a una efectividad del 95% en la detección de pulsaciones del teclado.

### Captura inalámbrica.

Saliéndonos un poco del concepto tradicional del dispositivo hardware que se debe de conectar al equipo monitorizado, y adentrándonos en otras técnicas como la captura de datos a través de conexiones inalámbricas, existen también ciertos prototipos de keyloggers muy específicos para capturar los datos de teclados inalámbricos de algunos fabricantes. Tal es el caso de un dispositivo llamado KeySweeper [22], un dispositivo basado en la plataforma Arduino[[28]](#footnote-28) que se presenta como un cargador USB de pared (realmente realiza dicha función), que captura todas las pulsaciones de teclas de cualquier teclado inalámbrico de Microsoft que usa un protocolo propietario en RF 2.4Ghz. De hecho, el dispositivo captura la señal inalámbrica, la decodifica, registra los datos, y puede reenviarlos a través de una conexión GSM[[29]](#footnote-29).

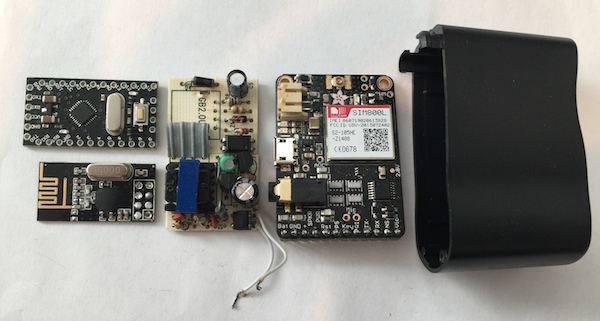


Ilustración 18. Muestra de los circuitos incorporados en el keylogger KeySweeper.

Evidentemente el dispositivo tiene que estar en el área de alcance de la señal inalámbrica que puede variar dependiendo de las características arquitectónicas de cada edificio, pero no es necesario que dicho dispositivo se encuentre conectado al ordenador, y en buenas condiciones ni siquiera en la misma estancia.

# Técnicas de distribución

Las técnicas de distribución en el caso de los keyloggers hardware son claras; se debe de tener acceso físico al equipo del que se quiere registrar la información. Conectar el dispositivo entre el teclado y el ordenador (normalmente por el puerto usb) y ya está listo.

En el caso de los keyloggers por software la distribución varía en función de los intereses o motivos de quien lo distribuya o instale.

En el caso de Padres cuyo objetivo es tener un control sobre el uso que hacen los hijos del ordenador y saber a qué lugares acceden, la técnica de distribución es la instalación directa en el equipo a monitorizar, pues son también propietarios de ellos.

En el caso de empresas, la distribución es la misma. La instalación del keylogger suele ser directa en el equipo, o incluso usan un keylogger hardware.

Cuando se trata de ciberdelincuentes no es habitual el uso de un keylogger hardware por el hecho que implica acceder físicamente al equipo de la víctima y por lo evidente que sería al ver el dispositivo. Así que el uso es mayoritariamente un keylogger software y su distribución puede ser por diferentes medios, aunque la técnica común es usar la ingeniería social con el objeto de hacer creer al usuario de que está descargando o recibiendo un documento o una aplicación legítima.

Pudiera ser que el mismo programa que se oculta al usuario contenga el keylogger, o podría ser, como ocurre muchas veces, que el programa que se instala es simplemente un pequeño código que es el que se encarga de descargar posteriormente el keylogger desde internet. Este último, es lo que se suele llamar un «*Dropper*»[[30]](#footnote-30).

* Memoria USB con software que induce al usuario, mediante ingeniería social, a ejecutarlo: demostración de un juego, software comercial falso, software pirata, etc, el cual tras ejecutarse descarga posteriormente y de manera silenciosa el keylogger en el sistema.
* Web que ofrece al usuario la descarga de un programa que “promete hacer algo” y en realidad, aunque podría hacer lo que dice sigilosamente instala también el keylogger. Muchas veces estas webs tienen detrás una red de anuncios publicitarios para la distribución (Malvertising). En muchos casos dicho anuncio falso se muestra en sitios legítimos los cuales incitan al usuario a hacer ‘click’ en el anuncio y en ese momento se descarga un troyano que instala el keylogger en el equipo.
* Envío de email phishing o spread phishing. Dependiendo de si es un intento al azar o con una víctima concreta (spread phishing), se hace un envío de un email falso, con la intención de que el usuario acceda a una web, o se descargue una aplicación la cual instalará el keylogger en su equipo.
* A través de redes P2P o descargas de sitios poco confiables. Muchos usuarios, con afán de buscar algún programa pirata, o un generador de claves, se descargan aplicaciones que en muchos casos pueden incorporar aplicaciones poco confiables como un keylogger.
* Payload[[31]](#footnote-31) de otros malwares. En otros casos puede darse que el keylogger venga como carga dentro virus o gusanos que se propaguen por la red, y como carga adicional a su propagación incorporen la función de instalar adicionalmente un keylogger.
* Instalación post-exploit[[32]](#footnote-32). Es menos habitual, pero también podría darse el caso de un ciberdelincuente que haya accedido a un sistema aprovechando alguna vulnerabilidad, y una vez explotada esta y habiendo accedido al sistema, instale un keylogger que le permita más adelante conocer otras credenciales que le puedan dar acceso a otros equipos.

## Ataques de incrustación de PDF

Dentro de cualquier método de distribución, no solo para los keyloggers, sino para cualquier tipo de malware en general, hay detrás una técnica para ocultar el malware, tanto para engañar al usuario, como para los diferentes programas de seguridad, ya sean antivirus, firewalls u otros sistemas.

Una técnica habitual, y extensamente conocida, es crear un documento PDF, aparentemente legítimo y con información real, el cual contiene en su interior el código malicioso, ya sea un troyano, virus, dropper u otro código que realice acciones en el equipo receptor. Suele ser una pieza de código pequeño para pasar desapercibido, que descarga posteriormente el malware que realiza la verdadera acción buscada, por lo que es habitual que se trate de un dropper.

Los ficheros PDF[[33]](#footnote-33) están ampliamente extendidos y ya considerados como un estándar de distribución de documentos en general y documentos de operaciones comerciales. Es por esto que a cualquier usuario no le extrañara recibir en su correo tras una compra en internet, un email con un documento adjunto que es un pdf que indica ser un comprobante de compra. Es aquí cuando el ciberdelincuente puede usar mediante ingeniera social, y teniendo en cuenta que los usuarios son curiosos (y confiados) por naturaleza, el hecho de realizar un ataque de incrustación de PDF [32].

El formato PDF ha ido creciendo con el tiempo y se le acabó dotando con un motor que le permite ejecutar lenguaje JavaScript. Esto lo convierte en un formato de documento dinámico y con gran capacidad para distribuir malware, ya que el usuario no es consciente de estar ejecutando un programa, sino que su sensación es la de abrir un documento que contiene información.

Para ello se puede crear un documento PDF el cual contenga el código malicioso dentro de la propia información del texto, y añadir un script para que ejecute este código malicioso en el momento de abrir el documento PDF. Suponiendo que el código malicioso incrustado dentro del PDF se llame “EsteEsUnMalware.exe” el script que insertaríamos para que se ejecute en cuanto se abra el PDF es el que se ve en la siguiente ilustración:

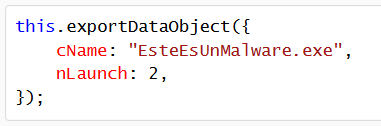


Ilustración 19. Script que ejecutará el malware incrustado una vez abierto el PDF.

Pero existe un problema adicional, pues, Adobe Reader o cualquier otro lector de PDF moderno, incluirá medidas de seguridad para evitar el lanzamiento de un archivo ejecutable como el que se ve en el ejemplo.

Aun así existen métodos en los que se usa un archivo XML embebido dentro el PDF en el que se inserta la etiqueta «.SettingContent-ms» [33] que es una función soportada por Windows 10 con la que se podría llegar a ejecutar el malware sin que el antivirus o el sistema operativo se percate de ello[[34]](#footnote-34).

Independientemente del método anterior, el formato PDF sigue permitiendo la ejecución de código, y siempre cabe la posibilidad de que se encuentren métodos adicionales para la descarga y ejecución de malware.

## Incrustación en documentos de Word

De la misma manera que en el formato PDF, pasa algo similar con los documentos de Microsoft Office[[35]](#footnote-35), y habitualmente usados en Word y Excel. Esta suite ofimática incorpora un lenguaje basado en Visual Basic, llamado VBA [34], o Visual Basic para Aplicaciones, que brinda las mismas posibilidades de ejecución de código que los archivos PDF.

Así pues, se pueden usar técnicas similares a las descritas en los documentos PDF, pero con un lenguaje de scripting diferente, y se podría programar un documento de Word el cual ejecute automáticamente [35] una macro de código malicioso [36] en el sistema.

## **Incrustación dentro de Scripts PowerShell**

No es común, pero si posible. Si el usuario es engañado y es consciente de que ejecuta un script de PowerShell[[36]](#footnote-36) del sistema, es posible insertar código oculto [37] dentro de este script que pase desapercibido para el usuario y ejecute otras acciones que no espera. El código se suele codificar en Base64[[37]](#footnote-37) para insertarlo como una cadena de texto, hecho que facilita la inclusión de datos binarios, por un lado, y oculta el código a la vista del usuario.

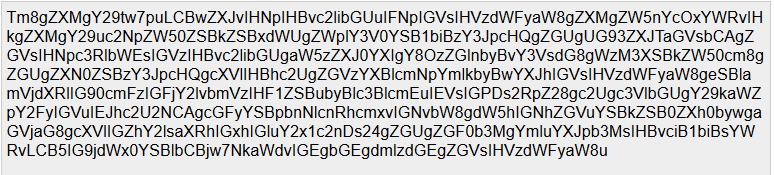


Ilustración 20. Muestra del texto del párrafo anterior codificado en Base64.

# Técnicas de ocultación y evasión

Cuando hablamos de técnicas de ocultación y evasión nos referiremos a los keyloggers software, ya que en el caso de un keylogger hardware es evidente que su ocultación es prácticamente imposible. No obstante, haremos un inciso, pues existen algunos fabricantes que tienen circuitos de dimensiones muy reducidas [26] (cercanas al tamaño de una tecla o menores) con función de keylogger preparados para ser instalados en el interior de teclados normales. Hecho que implica tener que desmontar el teclado, hacer el montaje, probablemente con necesidad de hacer soldaduras, y volver a cerrar el teclado.

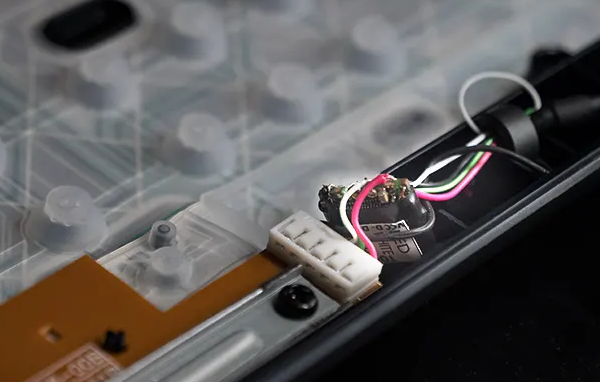


Ilustración 21. Muestra de keylogger hardware instalado en el interior de un teclado.

Se da el caso también de algunos fabricantes que ofrecen un teclado con apariencia totalmente normal a cualquier otro teclado del mercado, pero que incluye un keylogger incorporado internamente [27].

En el caso de los keyloggers basados en software, la ocultación del código pasa por cifrar los datos del código ejecutable, de manera que, si el programa es analizado por un antivirus o con cualquier herramienta de análisis estático, no se encontraran evidencias del código que realice la lectura del teclado. Esto es así porque el código está cifrado y/o comprimido en una zona de datos del ejecutable. Es en el modo de ejecución cuando este código cifrado que está en una zona de memoria, se descifra sobre otra zona y pasa a ser el código ejecutable real del keylogger. Este método es común a la mayoría de malware que este desarrollado para evitar su detección. De hecho, nada impide que puedan tener un doble cifrado para dificultar más su detección. Normalmente en este cifrado se suelen usar operaciones XOR [42], porque son sencillas de codificar y rápidas en su ejecución.

# Análisis de Keylogger Hardware

En este apartado se han adquirido, y estudiado, dos keyloggers de diferentes características de dos fabricantes diferentes. Se ha evaluado su facilidad o dificultad en la instalación y puesta en marcha, así como la eficacia, seguridad y utilidad de los resultados obtenidos. La diferencia más notable es que uno de los modelos tiene conexión WiFi, y esto también se evidencia en su coste, aunque curiosamente tienen exactamente las mismas dimensiones y factor de forma por lo que puestos uno al lado del otro no se puede distinguir el uno del otro.



Ilustración 22. Foto de los keyloggers hardware analizados



Ilustración 23. Vista en detalle de los dos keyloggers

A continuación, se muestra una tabla comparativa, a modo de resumen de las características de cada uno de los keyloggers.

|  |  |  |
| --- | --- | --- |
|  | **Keelog KeyGrabber Pico** | **Maltronics WiFi KeyLogger Pro** |
| Tipo | **USB** | **USB** |
| Precio | **29 €** | **44 £** |
| Dimensiones | **20mm x 18mm x 12mm** | **20mm x 18mm x 12mm** |
| Memoria | **16Mb** | **16Mb** |
| Protección por Contraseña |  |  |
| Plug & Play |  |  |
| Configurable |  |  |
| Modo Flash Drive |  |  |
| Indetectable por Antivirus |  |  |
| Ajuste layout/idioma teclado |  |  |
| Borrado remoto Log |  |  |
| Guarda pulsaciones en un fichero Log |  |  |
| Registro de fecha y hora |  |  |
| Envio Log por email |  |  |
| Conexión Wifi |  |  |
| Modo Access Point |  |  |
| Data Streaming |  |  |
| Monitorización remota a través de servicio web |  |  |

## KeeLog KeyGrabber Pico 16Mb

Este keylogger [33] solo requiere la única acción de conectarse al puerto USB entre el ordenador y el teclado. Si se conecta a un equipo portátil, pero no se conecta un teclado externo no tiene ningún efecto pues es imprescindible que los datos pasen a través del keylogger para capturar las pulsaciones de las teclas.



Ilustración 24.  
KeyGrabber en su empaquetado original.



Ilustración 25. KeyGrabber conectado al teclado.

Una vez conectado no se requiere de ninguna acción más y a partir de ese momento cualquier tecla pulsada en el teclado en el que está conectado será capturada y guardada en la memoria interna del dispositivo para poder ser revisada posteriormente.

Para ver el contenido capturado se debe de tener acceso físico al equipo, y desde el mismo equipo o extrayendo el keylogger y conectándolo a otro ordenador de la misma manera que en el equipo monitorizado, se podrá acceder al contenido de la memoria para ver el contenido de las teclas capturadas.

Para ello, el keylogger espera una combinación de teclas a modo de contraseña que deben de ser pulsadas al mismo tiempo. Por defecto la combinación de teclas es **K-B-S**, la cual activará el modo Pendrive que habilitará una letra de unidad en el sistema y nos permitirá acceder al fichero de log llamado LOG.TXT

También se podrá observar un manual en PDF y un archivo CONFIG.TXT que permite ajustar algunos parámetros en caso de ser necesarios (pero no imprescindibles) como la configuración de idioma del teclado conectado, cambiar el password por defecto, o si se guardan las teclas especiales.

El contenido del fichero se puede examinar con la misma aplicación de “Bloc de Notas” de Windows o cualquier otra aplicación, en incluso sistema, que pueda abrir un fichero de texto. Las teclas especiales se indican como palabras clave encerradas entre corchetes “[ ]” de manera que la tecla de borrado hacia atrás, conocida como “Backspace” queda registrada como “[Bck]”.

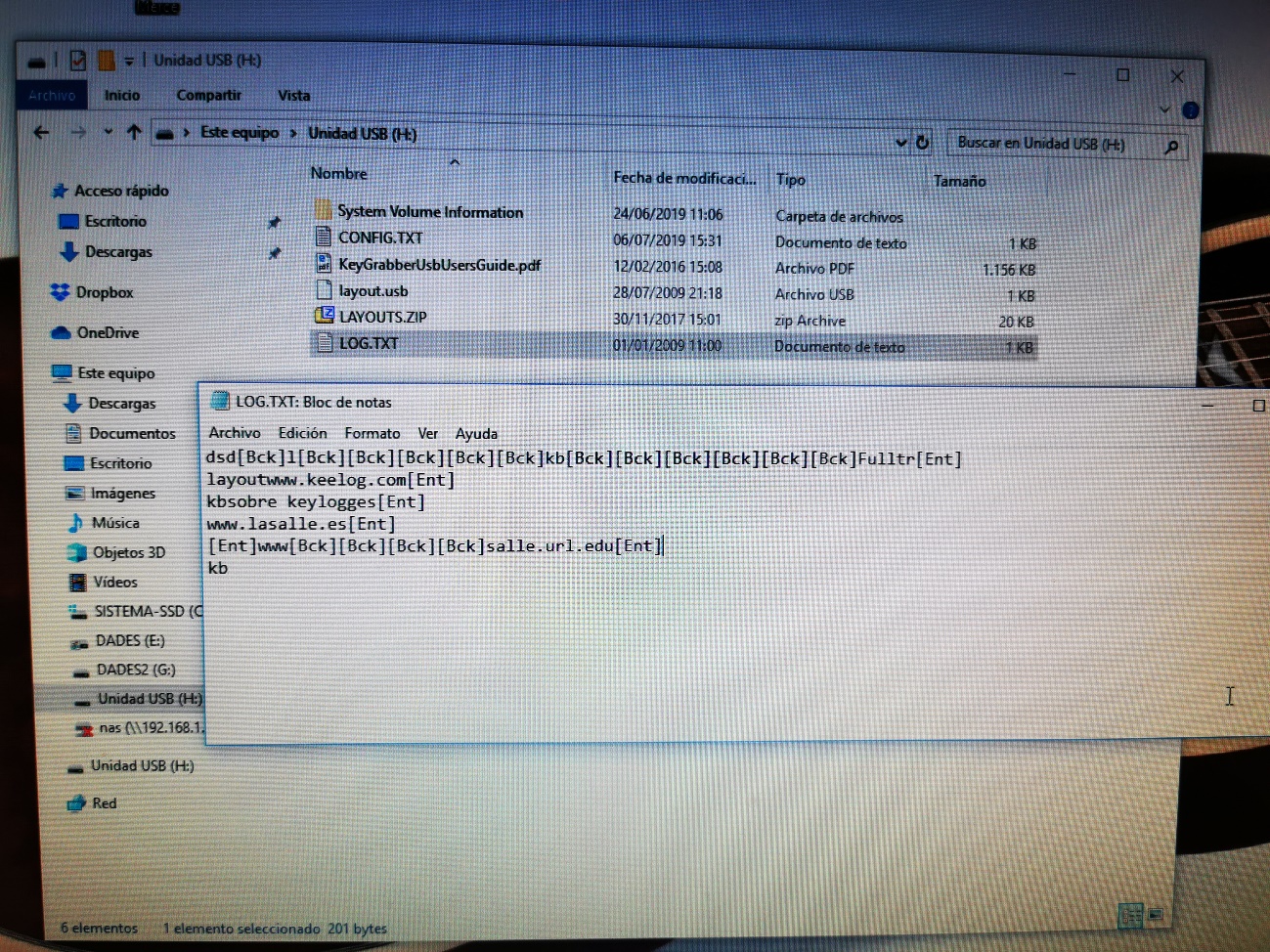


Ilustración 26. Ejemplo de fichero de LOG con el contenido capturado por el keylogger.

Como se puede ver, el contenido del fichero de texto es legible y registra todas las pulsaciones de teclado, incluidas teclas especiales. Aunque no es posible saber en ningún momento en que contexto han sido pulsadas las teclas. Si era un sitio web, un documento de Word, o Hoja de Cálculo o cualquier otra aplicación. Aunque siempre cabe la posibilidad de poder deducir cierta información. Tampoco es posible, en el caso de que el ordenador lo use más de una persona, saber que usuario ha tecleado el texto.

## Maltronics WiFi KeyLogger Pro

Este keylogger [34], al igual que el analizado en el punto anterior, también se puede usar directamente conectándolo al equipo del que se desean registrar las pulsaciones del teclado. No obstante, tiene una serie de características adicionales que se pueden configurar, y que hacen aconsejable hacer unos ajustes previos desde otro equipo para aprovechar al máximo todas sus posibilidades.

Al conectarlo al puerto USB de un equipo activa un punto de acceso WiFi con el SSID[[38]](#footnote-38) “AIR\_xxxxxx” donde “xxxxxx” es una combinación de números hexadecimales que varía con cada dispositivo. Esta conexión es inicialmente “abierta”, lo que significa que cualquier equipo que pueda detectar dicha conexión en su ordenador podrá conectarse al keylogger, de ahí la importancia de configurar el keylogger inicialmente.

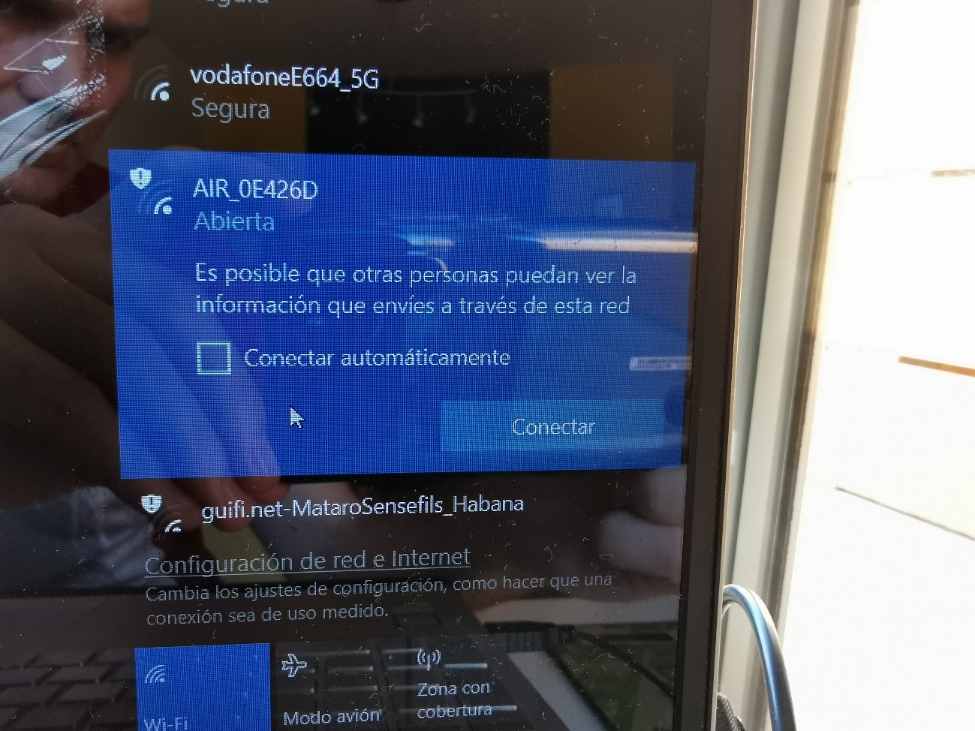


Ilustración 27. Vista del SSID activo del keylogger analizado.

Nos podremos conectar con un navegador a la administración del keylogger en la dirección IP (por defecto) 192.168.4.1, y el primer paso a efectuar será configurar el punto de acceso activando un cifrado de la comunicación y estableciendo una contraseña. Opcionalmente podemos cambiar el nombre del punto de acceso (SSID) y también establecer que sea oculto.

El siguiente punto a configurar es el nivel de detalle de las teclas especiales a registrar, el lenguaje del teclado utilizado o el nivel de filtrado del teclado que vendría a ser la sensibilidad que se puede ajustar en caso de no registrarse todas las teclas, aunque con el valor por defecto que trae parece ser suficiente.

Tras configurar estos valores, ya se puede conectar el keylogger en el equipo a monitorizar, y desde otro equipo (dentro de un radio de alcance WiFi) nos podríamos conectar al keylogger y ver el contenido de las pulsaciones de teclas que va registrando.

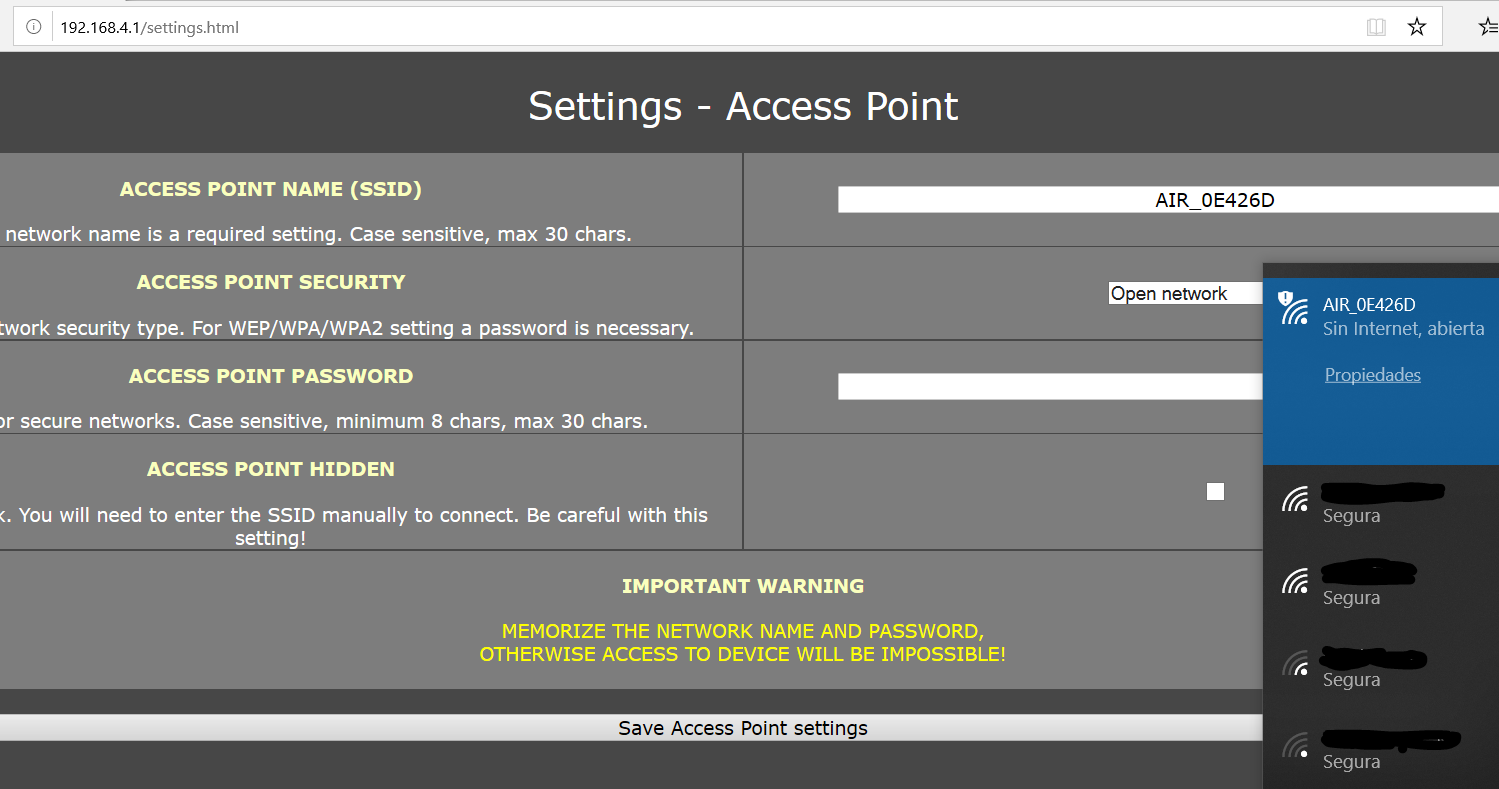


Ilustración 28. Pantalla de configuración del punto de acceso Wifi del keylogger.

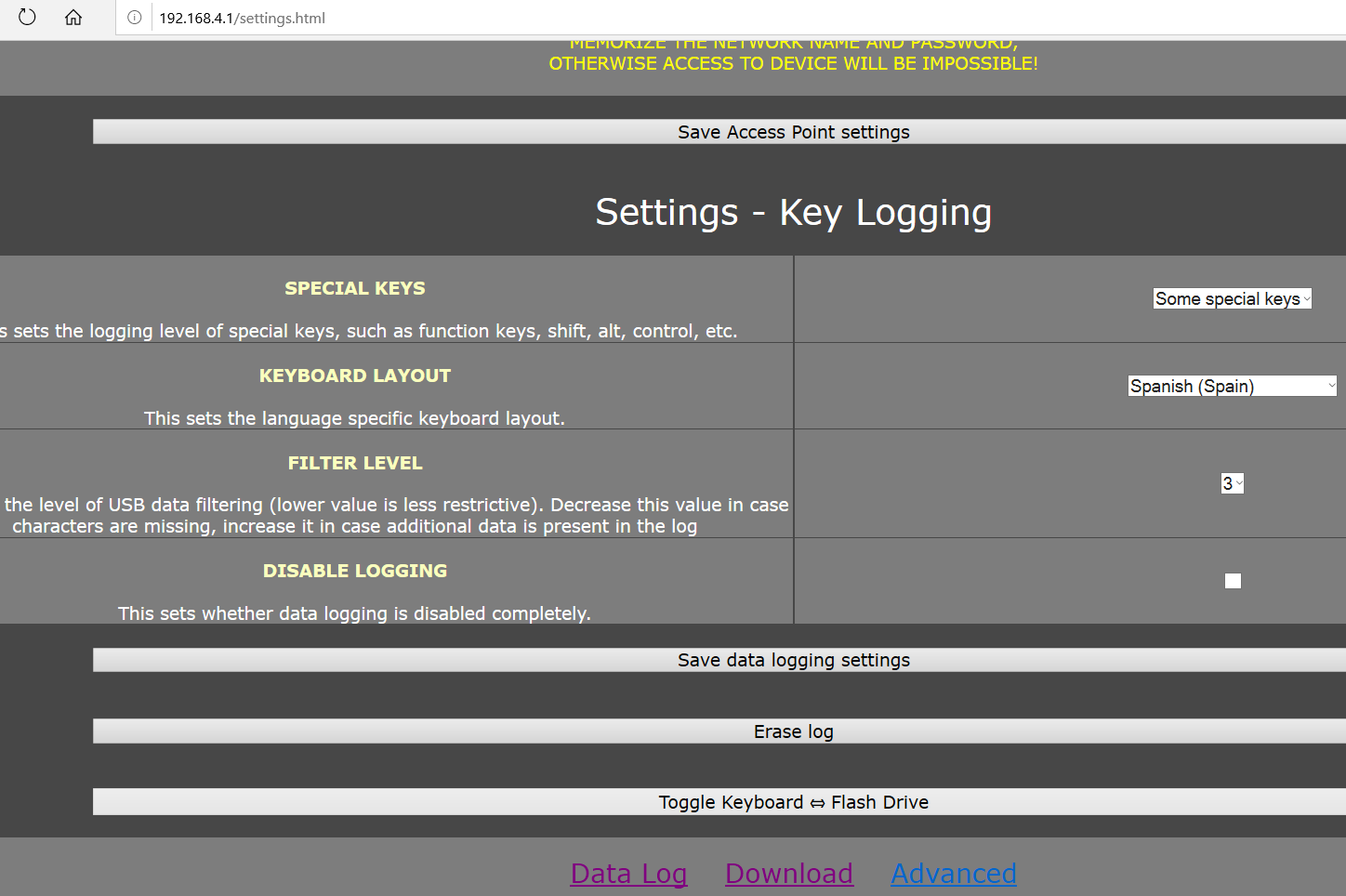


Ilustración 29. Pantalla de configuración del registro de teclas.

Al conectarnos al punto de acceso del keylogger podremos observar el contenido que va registrando. En esta pantalla el contenido no se muestra en tiempo real, sino que hay que acceder a un apartado donde se muestra el contenido y hay que recargar la página en caso de querer ver el contenido más reciente.

Dicho contenido también se puede descargar como un archivo de texto en nuestro equipo para poder ser examinado posteriormente con cualquier editor de texto, como el bloc de notas.

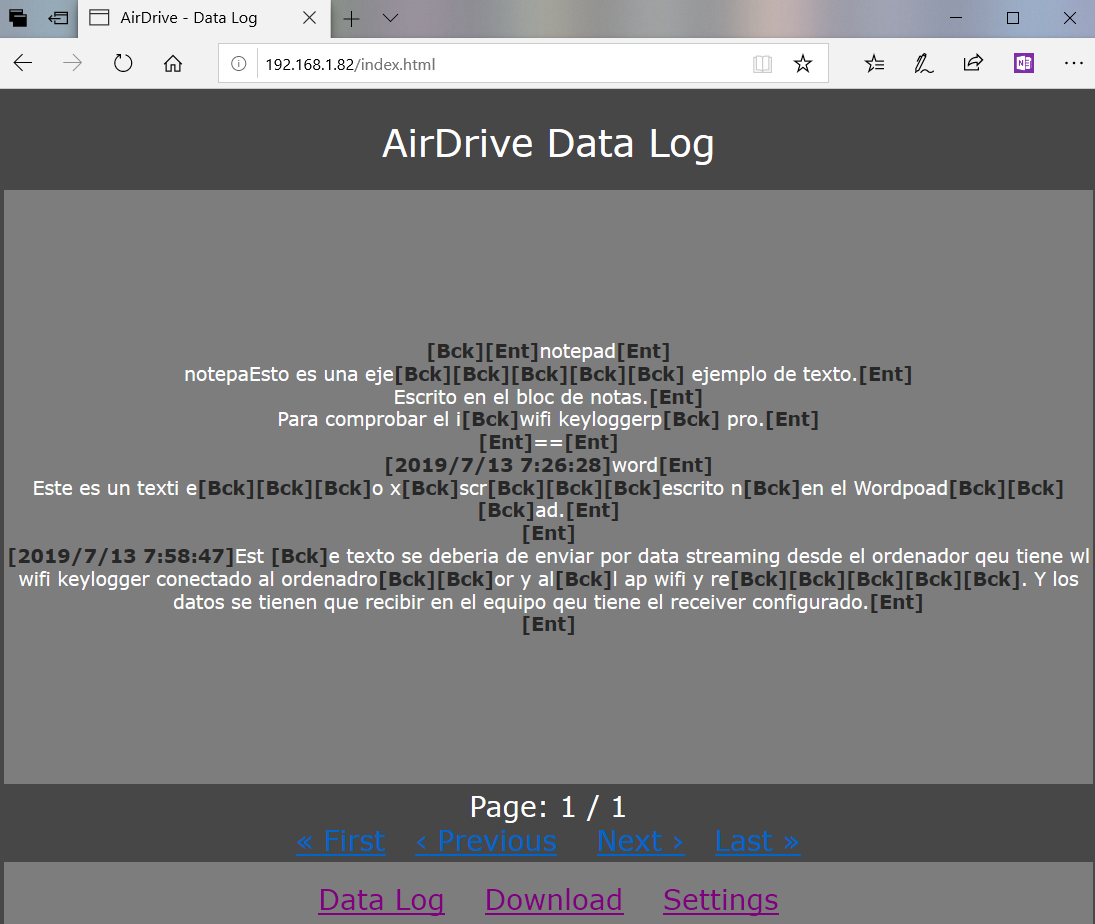


Ilustración 30. Muestra del contenido registrado por el keylogger accedido a través de la conexión WiFi.

Existe la posibilidad de configurar el keylogger para que envíe en tiempo real el contenido a otro equipo. Para ello el equipo receptor debe de tener una aplicación que esté a la escucha en un puerto concreto para recibir las teclas. Se adjunta en la misma página de control del keylogger un enlace para descargar un fichero ejecutable de ejemplo para comprobar el funcionamiento.

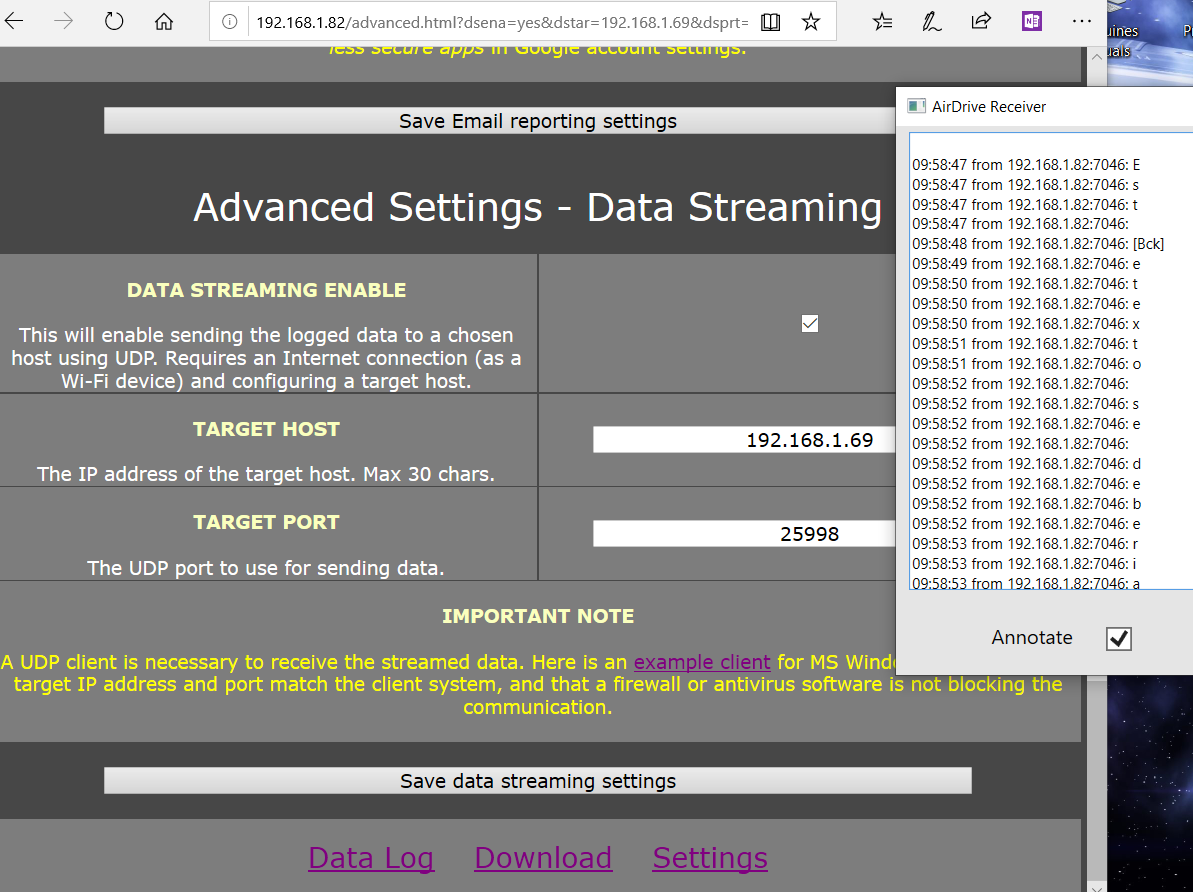


Ilustración 31. Ejemplo de recepción de las teclas del keylogger en tiempo real mediante data streaming.

Existen otras opciones avanzadas de configuración, como por ejemplo que el mismo keylogger se conecte como cliente a la propia red WiFi del usuario y posteriormente el acceso a este se pueda hacerse a través de la red local. Esto facilita el acceso al keylogger, pero adolece de un grave problema de seguridad, pues cualquier usuario conectado a la misma red que conozca la ip del keylogger, tendrá la posibilidad de acceder al panel de control de dicho keylogger sin necesidad de autenticarse, ya que no dispone de ninguna opción para activar un control de acceso mediante contraseña.

# Análisis de Keylogger Software

En el caso de los keyloggers por software se ha hecho una selección previa de ocho keyloggers, algunos “free” y otros comerciales con versión “demo” o “free”. De esta lista hemos confeccionado una tabla de características para ver que ofrece cada uno.

Cuadro comparativo (provisional)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Windows Spy  Keylogger 3.0** | **Revealer  Keylogger  Free** | **Spyrix  Keylogger  Gratis** | **Heavenward KeyLogger** | **Best Free  Keylogger Lite** | **Windows  Keylogger Free** | **KidLogger  Standart** | **IwantSoft  FreeKeylogger** |
| Tipo | **Free** | **Free** | **Free** | **Free** | **Free** | **Free** | **Free** | **Free** |
| Protección Contraseña | 🗶 | ✓ | ❓ | ✓ |  | ✓ | 🗶 |  |
| Inicio  automático | ✓ |  |  |  |  |  | ✓ |  |
| No requiere  Administrador | ✓ |  |  |  |  |  |  |  |
| Indetectable por Antivirus | ❓ | ❓ | ✓ |  |  |  |  |  |
| Modo Invisible | ✓ | 🗶 | 🗶 | ✓ | ✓ | ✓ | ✓ | ✓ |
| Path fichero  Log configurable | ✓ |  |  |  |  |  |  |  |
| Guarda pulsaciones  en un solo  fichero Log | ✓ |  |  |  |  |  |  |  |
| Nombre fichero Log Configurable | ✓ |  |  |  |  |  |  |  |
| Autoborrado Log |  |  |  |  | ✓ |  |  |  |
| Cifrado de Log | 🗶 | 🗶 | 🗶 | 🗶 | ✓ |  |  |  |
| Envio Log por email | 🗶 | 🗶 | 🗶 | ✓ | 🗶 |  |  |  |
| Log comprimido con contraseña | 🗶 | 🗶 | 🗶 | ✓ | 🗶 |  |  |  |
| Captura de  Pantalla | 🗶 | 🗶 | ✓ |  | ✓ | ✓ | ✓ | 🗶 |
| Registra Eventos  Sistema |  |  |  |  |  |  | ✓ |  |
| Monitoriza Sitios  Web Visitados |  |  |  |  |  |  | ✓ |  |
| Monitorización Remota a través de una web | 🗶 | 🗶 | ✓ |  |  |  |  |  |
| Monitoriza aplicaciones | 🗶 | 🗶 | ✓ |  |  | ✓ | ✓ | ✓ |
| Registro Fecha y Hora | 🗶 | 🗶 | ✓ |  |  | ✓ |  |  |
| Control Portapapeles | 🗶 | 🗶 | ✓ |  |  | ✓ | ✓ |  |
| Control Unidades  USB | 🗶 | 🗶 | ✓ |  |  |  |  |  |
| Comtrol Impresoras | 🗶 | 🗶 | ✓ |  |  |  |  |  |
| Registra Audio |  |  |  |  |  |  | ✓ |  |
| Foto Webcam Selfie |  |  |  |  |  |  | ✓ |  |
| Notas | Se debe de configurar  por separado en  cada sesión de usuario. Crea carpeta oculta en el sistema donde se instala la aplicación. |  |  |  |  |  |  | Bloqueo  de aplicaciones |

Posteriormente analizamos un par de estos keyloggers.

## Análisis de keylogger ….

Pendiente --- pendiente --- pendiente --- pendiente ---

-------------------------- [APARTADO PENDIENTE DE COMPLETAR] -------------------

¿??????????????????????

## Caracteristica 1

Pendiente --- pendiente --- pendiente --- pendiente ---

## Caracteristica 2

Pendiente --- pendiente --- pendiente --- pendiente ---

## Caracteristica 3

Pendiente --- pendiente --- pendiente --- pendiente ---

# Anti-Keyloggers

De la misma manera que hay software keylogger, también existe software anti-keylogger. Su función, tal y como se define en su nombre, es la detectar, eliminar o neutralizar el funcionamiento de los keyloggers.

Evidentemente, un buen antivirus también es imprescindible y también debería de detectar este tipo de malware, pero en algunos casos, un software específico que complemente este análisis también es recomendable.

-

-------------------------- [APARTADO PENDIENTE DE COMPLETAR] -------------------

<https://windowsreport.com/free-antikeylogging-software/>

<https://freeware.fandom.com/wiki/Lists_of_freeware_antikeyloggers>

<https://www.thewindowsclub.com/keylogger-detector-freeware-windows>

# Reversing de un Keylogger

Obtenemos una muestra de un binario que corresponde a un *malware* que incorpora un *keylogger*, y vamos a proceder a hacer un análisis de ingeniería inversa (*reversing*) en este apartado para conocer los puntos más relevantes que incorpora.

## Análisis preliminar del binario

Ante cualquier fichero binario (ejecutable) del que no se conoce que hace exactamente, ni su procedencia, empezaremos por llevar a cabo un análisis estático para obtener la máxima información posible sobre las acciones que puede ejecutar y todo esto dentro de un entorno virtual para minimizar el impacto en caso de producirse una ejecución involuntaria de dicho programa.

Para este análisis se ha utilizado una máquina virtual dentro un VMWare Workstation con un Windows 10 Professional de 64 bits, con 5 Gb de memoria RAM.

El binario que hemos obtenido tiene un tamaño de 61Kb, con el nombre “***sample.bin***”, no tiene la extensión .exe para evitar una ejecución accidental, y procedemos a extraer la información básica sobre el archivo.

El primer paso es obtener el hash MD5 del archivo para poder buscar información de este hash en otras fuentes. Para ello utilizamos la herramienta **PE-Bear** que nos indica este hash, además de otra información útil.

**HASH MD5 59b57bdabee2ce1fb566de51dd92ec94**

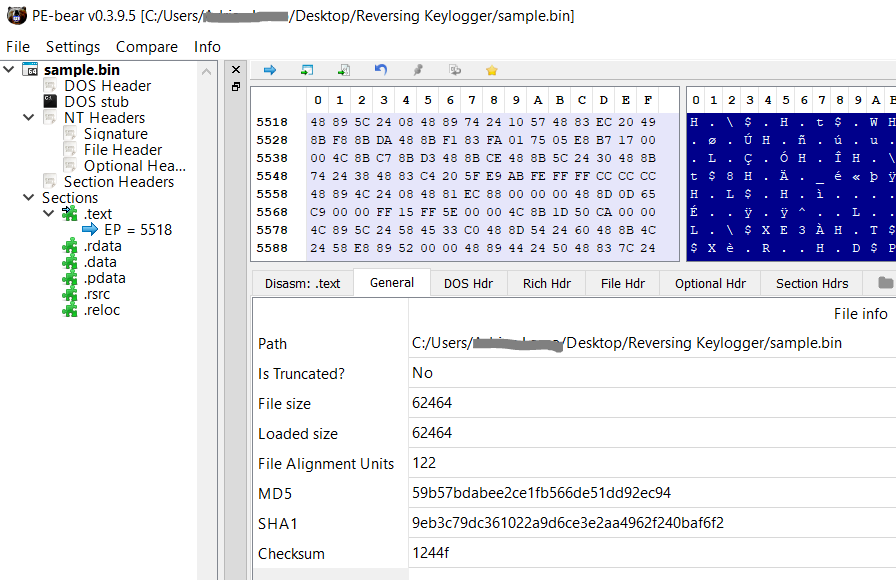


Ilustración 32. Muestra de PE-Bear. Herramienta para analizar cabeceras PE de ejecutables.

También podemos hacer un primer análisis estático del binario, ya que permite desensamblar el código y mostrarlo para hacer un primer seguimiento visual. De esta primera lectura del código desensamblado observamos que hay algunas llamadas a la API de Windows que nos pueden ser de interés o revelar información interesante.

Por ejemplo, **IsDebuggerPresent** [35], que se utiliza para detectar si está corriendo el código dentro de un depurador y podría llevar a que el código se comportara de manera diferente con esta detección. O una llamada a **WriteFile**, lo que nos hace notar de manera inequívoca que el binario realiza escrituras directas sobre un archivo. O **GetProcAddress**, que se utiliza para recuperar la dirección de una función en un archivo DLL cargado en la memoria.

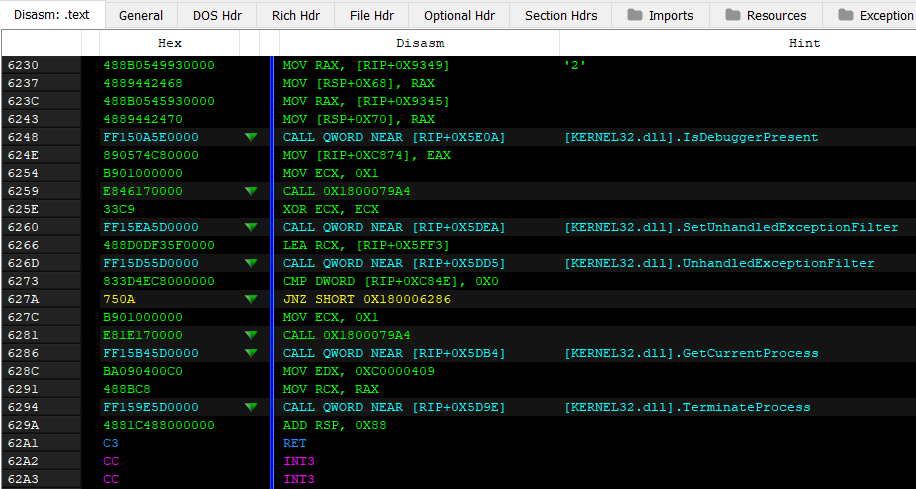


Ilustración 33. Muestra parcial del desensamblado del binario con la herramienta PE-Bear.

Complementamos la lista de funciones utilizando la herramienta ***010 Editor***, que nos permite buscar todas las cadenas de texto en el binario que no estén cifradas. De esta manera tendremos la lista completa de las llamadas a la API de Windows, así como otras cadenas de texto que pudieran ser relevantes.

De las diferentes informaciones extraídas podemos constatar los siguientes datos:

* Está desarrollado / compilado usando Visual Studio para Windows 64 bits y, por lo tanto, no funciona en 32 bits.
* Que es un falso ejecutable ya que el tipo de fichero es una DLL
* Que utiliza la librería kernel32.dll
* Que la fecha de compilación es 31/marzo/2015.
* Que tiene llamadas a algunas funciones de la API que suelen ser habitualmente usadas por malware: GetProcessWindowStation, GetUserObjectInformation, GetModuleFilename, TerminateProcess, GetCurrentProcess, y otras más.

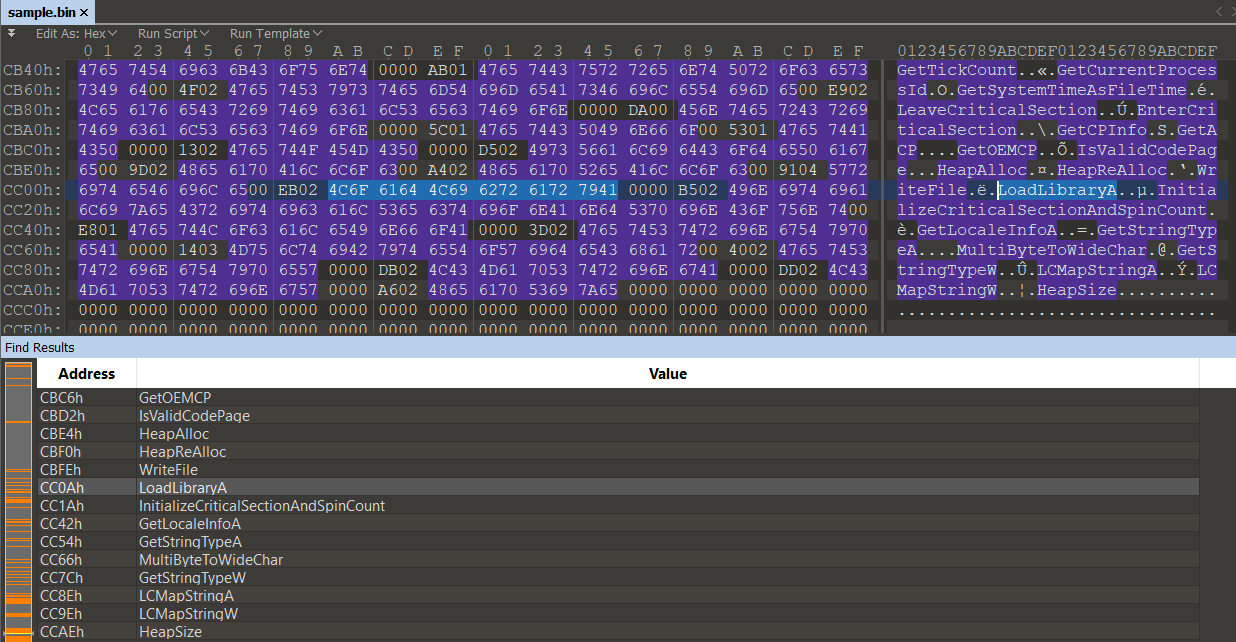


Ilustración 34. Muestra de cadenas de texto encontradas en el binario.

Utilizaremos también algunas herramientas on-line, como el sitio web **Virus Total** [35] donde subimos el binario para que nos indique si algunos motores de Antivirus lo detectan como malicioso y si ya es un malware conocido.

En este caso, efectivamente es reconocido como malware por 45 de los 67 motores que utiliza Virus Total. Es un ejecutable de 64 bits conocido como **TurlaKeylogger.bin**, aunque es probable que el nombre pueda cambiar dependiendo del fabricante del antivirus, y en el caso del antivirus de Microsoft es detectado como “**Tigggre!rfn**”.

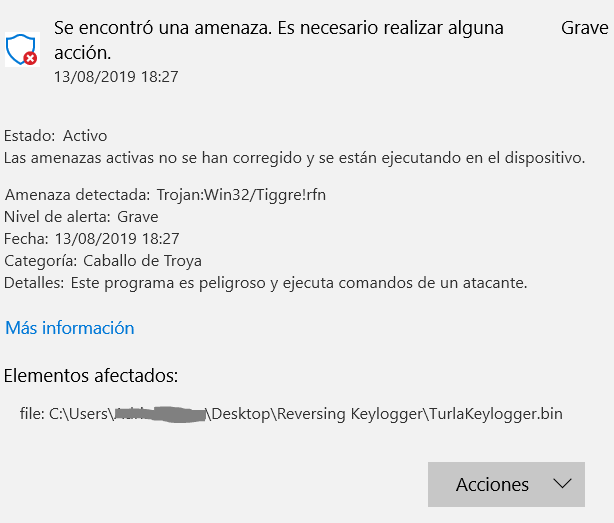


Ilustración 35. Detección de Turla Keylogger por Antivirus de Microsoft.

De este sitio web se puede obtener más información que complemente nuestro análisis como por ejemplo el comportamiento analizado en el código: que ficheros abre o modifica, y también que procesos finaliza o crea en el sistema.

También podemos obtener información adicional aportada por la comunidad, como país de origen, si pertenece a una APT[[39]](#footnote-39) conocida o fecha de la primera detección.

El enlace de referencia de este archivo en Virus Total es:

<https://www.virustotal.com/gui/file/b7b5d28be983c774ef83a8960a68134732a79818c572e8800cea6428f27fb114/detection>

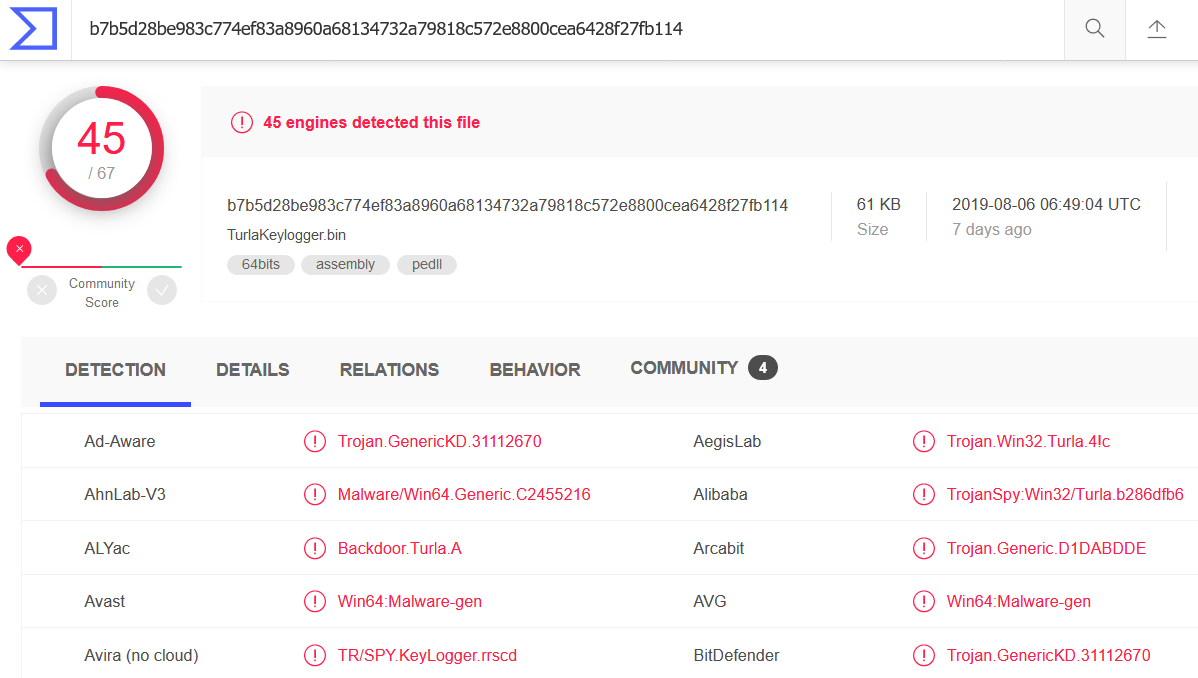


Ilustración 36. Muestra de la información ofrecida en Virus Total.

Su fecha de creación es del 31 de marzo de 2015, aunque el primer envío a Virus Total (y detección) es del 12 de marzo de 2018, tres años después de su creación, y sorprende que en la fecha en que se ha enviado para su último análisis, 6 de agosto de 2019 aún haya 22 motores de antivirus que no lo reconozcan.

Para completar este análisis preliminar nos ayudamos de dos sitios web adicionales que utilizan la ejecución dinámica del binario en entornos virtuales para ofrecernos, si cabe, información adicional o ampliada.

El primero es **Hybrid Analysis** [36] el cual nos informa de que el binario no se ejecuta en entornos de 32 bits, que es una DLL, que los lenguajes usados son Ruso e Inglés, y también nos ofrece información de las cadenas de texto que ya habíamos averiguado anteriormente así como las que son indicadoras de que el binario es sospechoso de ser malicioso.

El enlace de referencia del archivo en Hybrid Analysis es:

<https://www.hybrid-analysis.com/sample/b7b5d28be983c774ef83a8960a68134732a79818c572e8800cea6428f27fb114/5ba8e05b7ca3e129745e6813>

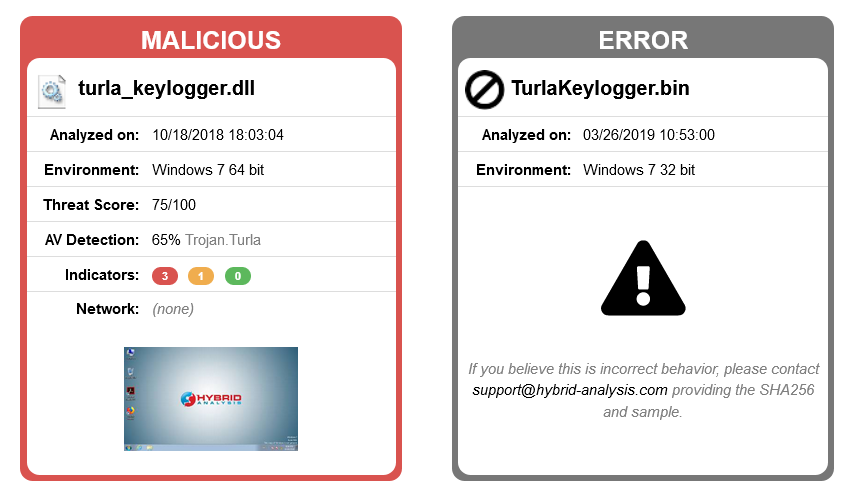


Ilustración 37. Muestra ejecución del binario en entorno virtual de Hybrid Analysis.

El siguiente sitio web es INTEZER Analyze [37] que nos sirve para confirmar gran parte la información anteriormente obtenida, y añadir algunas informaciones interesantes.

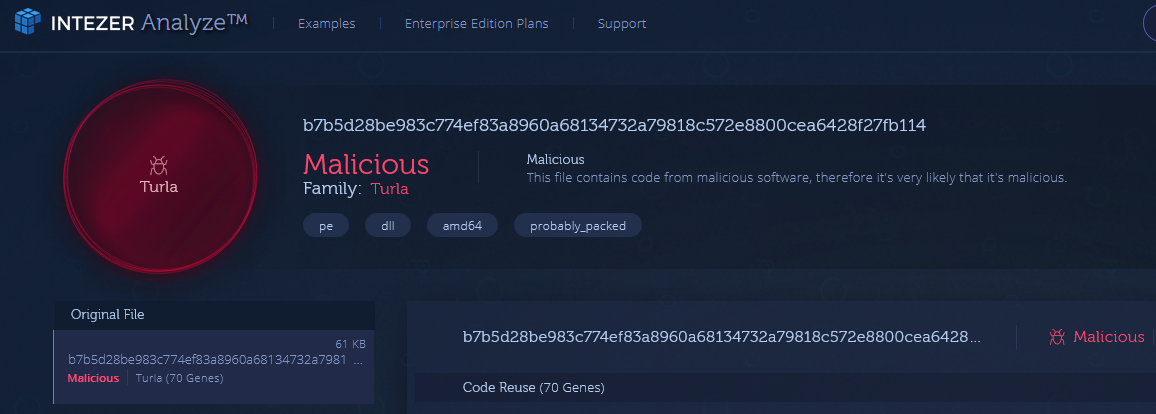


Ilustración 38. Muestra de la información ofrecida por INTEZER Analyze.

Nos confirma que el binario es malicioso, y vemos además que nos ofrece información de otras muestras de binarios analizados que pudieran tener relación por similitudes en el código o bloques de este que son reutilizados. Muchos programas maliciosos están desarrollados con lenguajes de alto nivel y entornos de desarrollo comerciales (por ejemplo, Visual Studio [40]) que generan siempre un patrón concreto en algunas operaciones, y por otro lado es probable que utilicen librerías de desarrollo de malware proporcionadas por terceros y, por lo tanto, muchos programas maliciosos tienen partes de código similares si su objetivo es realizar las mismas operaciones.



Ilustración 39. Muestra de binarios que tienen partes de coincidencia en el código.

Si observamos algunos bloques de código podemos ver una parte donde se cargan punteros de datos a memoria para realizar ciertas operaciones (que veremos más adelante) y se puede comprobar que estas zonas de memoria están rellenas con la palabra “PADDING” (“RELLENO” en Castellano) repetidas veces.

Esto puede significar que el programa se reserva un área de memoria vacía, o sin código efectivo en este momento, porque realizará algún tipo de descifrado de datos des de otra zona de memoria para dejar los datos descifrados aquí. Es un método común de ocultación de muchos programas maliciosos, en los que el código está oculto (cifrado) en una zona de memoria y en el momento de la ejecución hay una pequeña parte de código que se encarga de descifrar estos datos y colocarlos en otra zona de memoria donde finalmente estará ubicado el verdadero programa malicioso. Esta información se confirma por el hecho de que INTERZER nos informa inicialmente con “*probably packed*”.

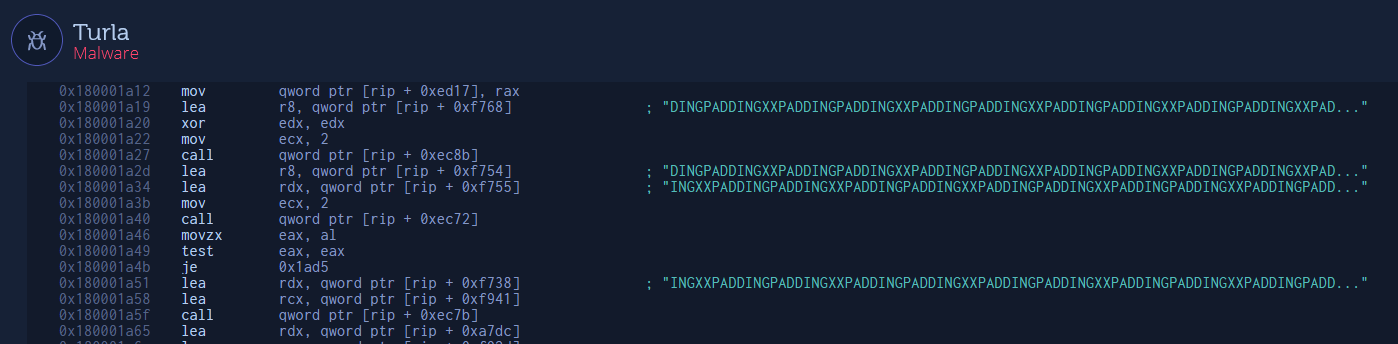


Ilustración 40. Ejemplo de punteros a zonas de memoria vacías para rellenar con datos.

También podemos observar partes de código donde existen repetidas operaciones de XOR [41] (OR exclusivo) que son habitualmente utilizadas en el cifrado y descifrado de la información.

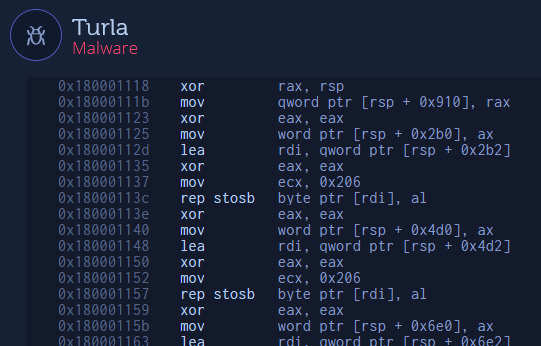


Ilustración 41. Muestra de bloque de código donde hay repetidas operaciones XOR.

Y ya para finalizar, podemos forzar una ejecución dinámica del binario sobre un entorno virtual, el cual nos mostrará la información obtenida tras ejecutar dicho binario. Aquí podemos comprobar que el binario tiene el nombre “loaddll64.exe” y que ahora tiene un tamaño de 92Kbs.



El enlace de referencia al análisis de nuestro binario es el siguiente:

<https://analyze.intezer.com/#/analyses/690085a2-ed14-4673-b9ec-1afdc4139d55>

## Análisis estático

Aunque las acciones realizadas en el punto anterior se pueden considerar también parte del análisis estático, las ponemos por separado por considerar que el análisis estático supone algo más profundo y más centrado en la ejecución del binario.

Para llevar a cabo este análisis se han utilizado dos herramientas diferentes. Por un lado, por sus diferentes capacidades de análisis y herramientas que ofrecen cada una, y por otra para ayudarnos en la comprensión del código desensamblado y contrastar las diferentes informaciones que nos ofrece cada herramienta.

Estas herramientas constan principalmente de un desensamblador, pero que se complementan con módulos de análisis, generación de código, depuración y otros complementos con plugins externos.

La primera herramienta es Ghidra [38], creada por la NSA (Agencia de Seguridad Nacional del Gobierno de los Estados Unidos) y que liberó con todo su código fuente en marzo de 2019.



Ilustración 42. Muestra parcial de la pantalla de Ghidra una vez cargado el binario a analizar.

La segunda herramienta que vamos a utilizar es Radare [39] (conocido también como **radare2**), desarrollada en España y también de código abierto, y la complementaremos con Cutter [40], que es un GUI para usar con radare2.

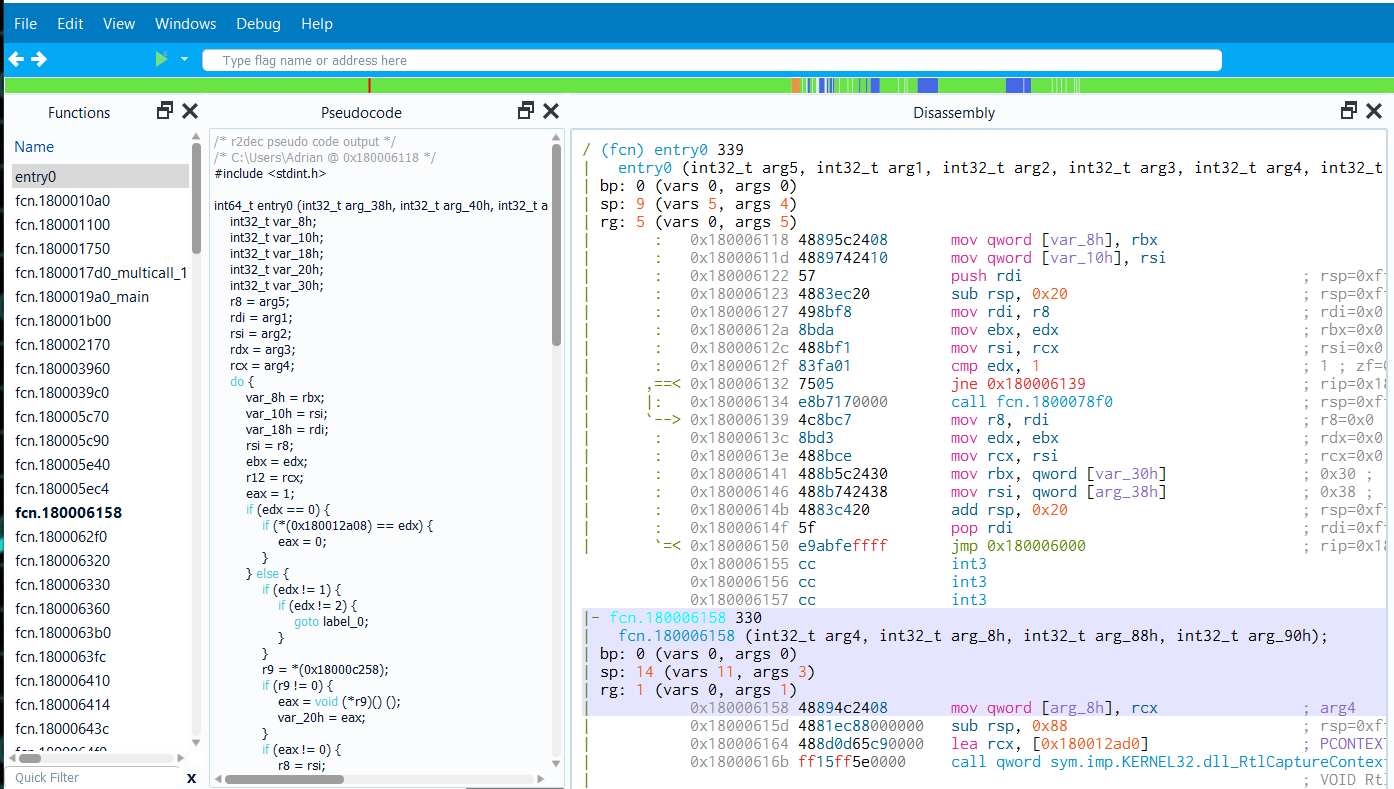
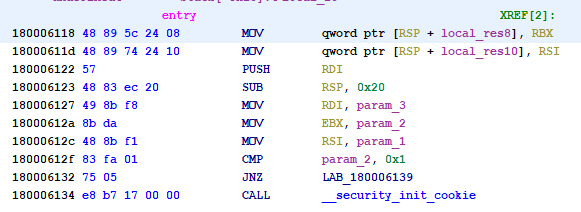


Ilustración 43. Muestra parcial de radare2 junto con Cutter una vez cargado el binario a analizar.

Una vez cargado el binario podemos ver el punto de entrada de la dll que está en la dirección 0x180006118. A partir de este punto hacemos una primera observación del código intentando seguir las diferentes comparaciones y saltos para intentar reconocer algunas llamadas a la API del sistema e ir reconociendo módulos y funciones.



Una observación en unas posiciones de memoria más adelante en el código nos permite confirmar lo que habíamos observado en el punto del análisis preliminar efectuado anteriormente, y comprobamos que existen llamadas a la función *IsDebuggerPresent* [41] de la API de Windows, que es una de las medidas de protección para evitar que el programa sea depurado.

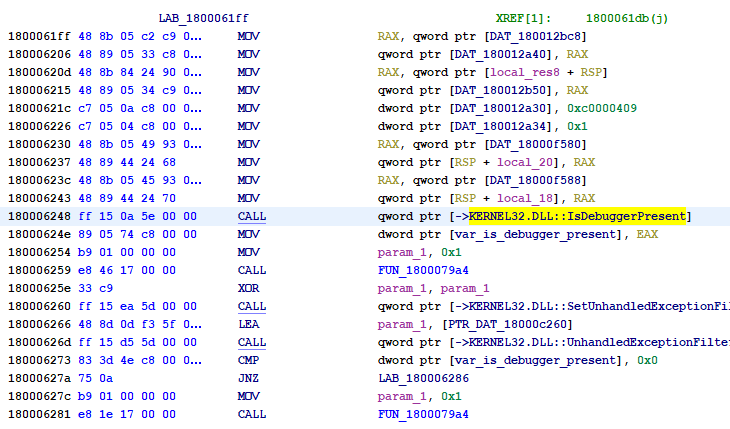


Ilustración 44. Muestra donde se observa una de las medidas de protección usadas.

Tras cargar el binario en las dos herramientas y observar el código en el punto de entrada podemos ver que hace una llamada a una subrutina, la cual tiene múltiples llamadas a otra subrutina en la dirección 0x180001750 con diferentes parámetros en cada una de las llamadas. A medida que vayamos detectando las funciones de parte del código renombraremos las funciones con nombres más entendibles y poder seguir mejor el flujo posteriormente.



Ilustración 45. Función que realiza repetidas llamadas a la dirección 0x180001750

Estas llamadas repetitivas a dicha subrutina, cargan previamente 3 valores diferentes en los registros *r8d*, *rdx* y *ecx* lo que hace sospechar que pueda tratarse de alguna rutina de cifrado o descifrado, aunque lo podremos constatar más adelante.

Dicho código en ensamblador, traducido a código descompilado en C quedaría de la siguiente manera.



Ilustración 46. Código en C resultante del código en ensamblador anterior.

El valor cargado en rdx es una dirección de memoria, por lo que es muy probable que sea donde están ubicados los datos a tratar. El valor establecido en r8d parece ser un contador después de analizar su comportamiento dentro de la subrutina.

A medida que vamos descubriendo el uso de cada registro o dirección de memoria dentro del código analizado sustituimos los valores con nombres de variables más comprensibles en el pseudocódigo generado.

Deducimos pues que los datos a descifrar se obtienen de diferentes partes de memoria, empezando por la dirección 0x18000f2f0 y que tras realizar las operaciones de descifrado se van colocando a partir de 0x18000f010.

Con Ghidra mostramos el código a modo de gráfico con cada una de las subrutinas y la parte del descifrado quedaría tal y como se muestra en la siguiente ilustración.

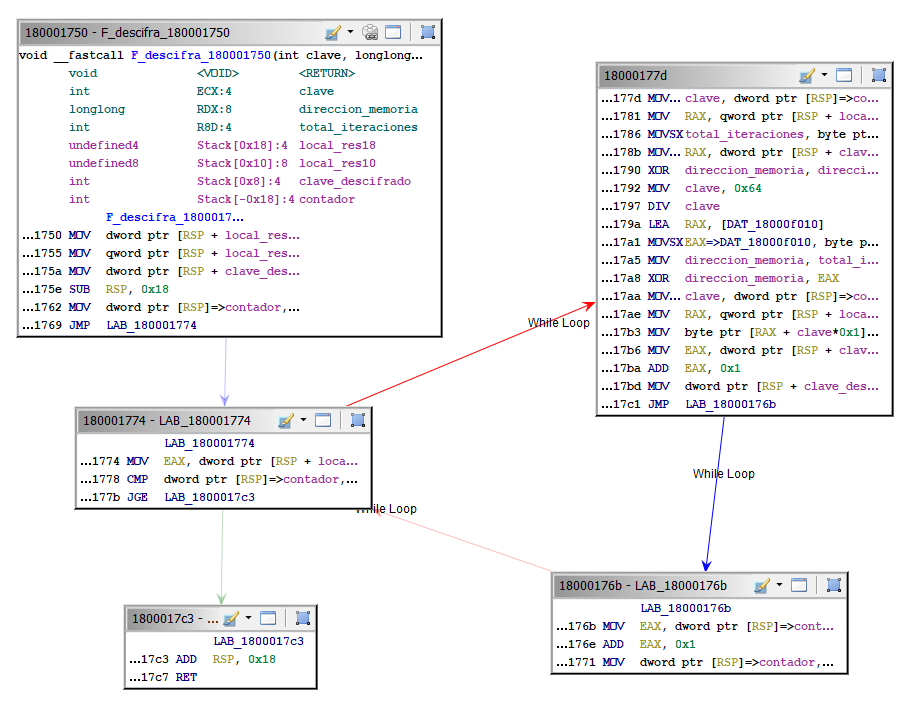


Ilustración 47. Visualización en modo gráfico de la subrutina de descifrado.

Haciendo un seguimiento del código, visualizando en modo gráfico, y tras sustituir los valores de los registros y direcciones de memoria por nombres de variables adecuados podemos obtener el siguiente pseudocódigo de descifrado con Ghidra que es más optimizado que el pseudocódigo que nos ofrece radare2, aunque para llegar hasta este punto se ha realizado el análisis con los dos desensambladores.



Ilustración 48. Código en C resultante de la función de descifrado.

Realizar el descifrado de todos los datos es bastante complicado en modo estático pues implica de alguna manera ejecutar el código para que recorra las diferentes posiciones de memoria y haga las operaciones pertinentes para obtener los datos. Porque en este punto necesitaríamos un script que nos decodificara estos datos.

Llegados a este punto tendríamos un gráfico de llamadas entre funciones que quedaría de la siguiente manera.

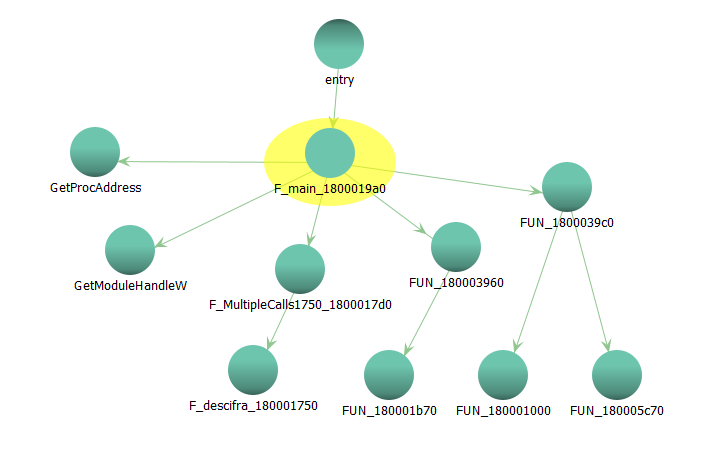


Ilustración 49. Muestra de gráfico de llamadas entre funciones.

Si volvemos a la función principal podremos ver como después de llamar a la función que realiza el supuesto descifrado sobre unas posiciones de memoria, posteriormente llama a GetModuleHandleW y GetProcAddress pasándole como parámetro las direcciones de los datos descifrados que contienen los nombres de las funciones a ejecutar y obtiene una dirección de memoria a la que posteriormente llama.



Ilustración 50. Sección de la función principal del binario.

A continuación, la función ubicada en 0x1800039c0 vemos que realiza un largo proceso de movimiento de valores de memoria que se descifran también durante la ejecución. Es de nuevo, una función añadida para dificultar el depurado del malware evitar, si cabe, saber qué hace exactamente.



Ilustración 51. Sección de la segunda función de descifrado.

Dejamos en este punto el análisis estático y con la información obtenida pasamos al análisis dinámico.

## Análisis dinámico

Vista la dificultad y tedioso trabajo del análisis estático, procedemos a hacer un análisis dinámico a partir de los datos que ya conocemos del punto anterior, que nos ayudan a definir puntos de parada en la depuración, para ver qué datos hay en memoria y seguir mejor el proceso. No obstante, tratándose de un malware, cabe insistir que no solo hay que llevar a cabo esta acción en una máquina virtual, sino que es aconsejable la toma de continuos *snapshots[[40]](#footnote-40)* y, sobre todo, uno inicial antes de empezar el depurado. También es recomendable desactivar cualquier comunicación de red para aislar el malware lo máximo posible en la máquina y evitar su propagación por error o descuido.

Para llevar a cabo la depuración del binario hemos usado el debugger x64dbg [42], y lo primero que hemos hecho es indicarle que se pare en la entrada de cada librería DLL[[41]](#footnote-41) para evitar que el código se siga ejecutando e iremos monitorizando hasta llegar el inicio de nuestro binario. Una vez encontrado el inicio de nuestro binario (observando el mapa de memoria veremos el nombre del binario junto a las diferentes secciones que tiene cargadas) podremos establecer un punto de parada en este punto para las siguientes veces que reiniciemos el análisis.

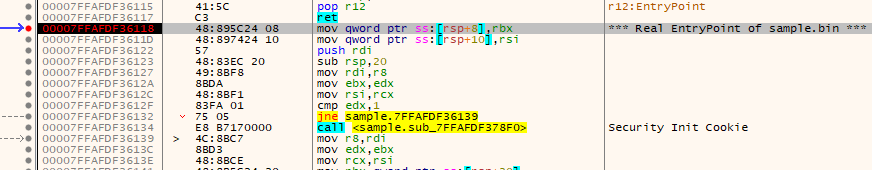


Ilustración 52. Fragmento de código donde se muestra el punto de entrada del binario a analizar.

Tal y como se muestra en la ilustración anterior, podemos observar como el inicio del binario está en la dirección 0x7FFAFDF36118[[42]](#footnote-42), aunque esta dirección no tiene por qué coincidir con el inicio del bloque de código tal y como se muestra en el mapa de memoria de la siguiente ilustración.



Ilustración 53. Parte del mapa de memoria del binario cargado en x64dbg.

Seguimos ejecutando paso a paso hasta llegar a la verdadera función principal de la aplicación. Ahí identificamos la llamada a la función que realiza múltiples llamadas a otra función donde lee datos de una zona de memoria, los decodifica y los coloca en otra área de memoria. A medida que identificamos las funciones cambiamos las direcciones de memoria en el depurador con un nombre descriptivo y añadimos un comentario que nos ayude en el seguimiento.

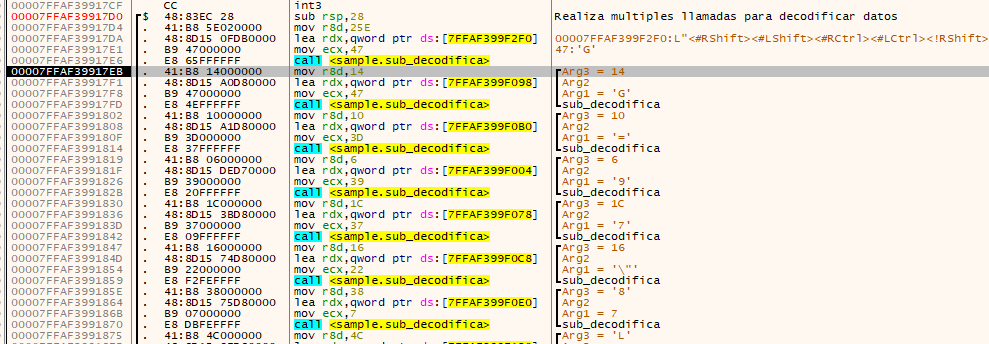


Ilustración 54. Función que realiza múltiples llamadas para decodificar datos de memoria.

Si observamos parte de la zona de memoria donde se han ido dejando los datos podremos ver los nombres de las teclas como «RShift» «LShift» «RCtrl» «PageUp» etc.

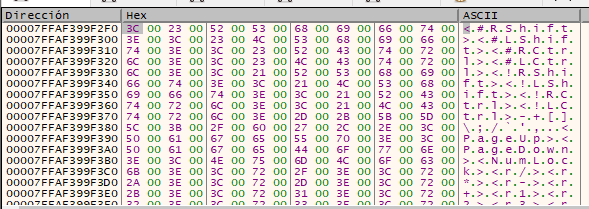


Ilustración 55. Muestra de zona de memoria donde se dejan los datos una vez decodificados.

Seguidamente nos encontramos con un segundo proceso de decodificación de datos similar al anterior, aunque las operaciones son diferentes. Y tras finalizar el proceso podemos observar en memoria como el resultado de este descifrado nos revela nombres de funciones de la API de Windows que nos pueden ayudar a saber qué hace el código. De entre ellas, una muy reveladora como la función SetWindowsHookExw [23].

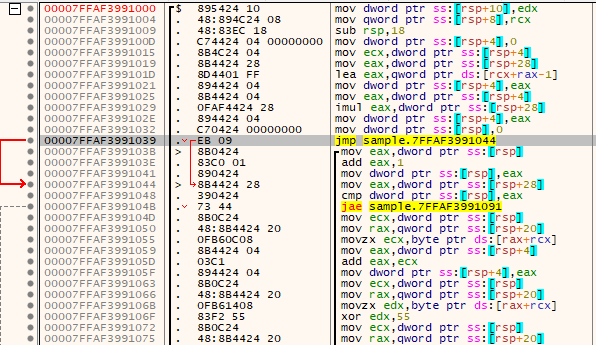


Ilustración 56. Muestra de parte del código del segundo proceso de descifrado.

Llegados a este punto, ya sabemos cuál es el resultado de las funciones que habíamos analizado en el apartado del análisis estático, pero de las que no podíamos obtener los datos resultantes, ya que en ese punto no ejecutábamos el código.

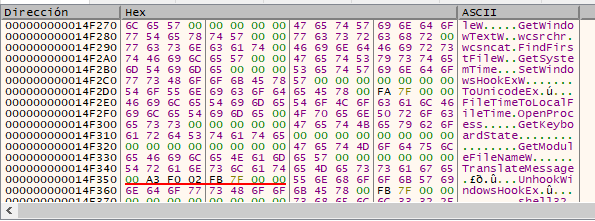


Ilustración 57. Datos descifrados con los nombres de las funciones de la API de Windows.

-------------------------- [APARTADO PENDIENTE DE COMPLETAR] -------------------

# Desarrollo de un Keylogger

En este apartado llevaremos a cabo el desarrollo de un keylogger, con el único objetivo de aprender su funcionamiento y ver que el núcleo de la funcionalidad básica es un código muy corto, simple y sencillo.

## Lenguaje elegido para el desarrollo.

El desarrollo de un keylogger suele necesitar de un lenguaje que nos acerque a funciones de más bajo nivel puesto que necesita “hookear” una función esencial del sistema. Los primeros keyloggers estaban escritos en lenguaje C, o incluso en Ensamblador por su cercanía a las funciones del sistema y su mayor control.

Hoy en día es hacerlo posible con otros lenguajes de mayor nivel que facilitan la creación del código más comprensible sin renunciar a la potencia del acceso a funciones de más bajo nivel.

El lenguaje elegido finalmente para el desarrollo ha sido Python versión 3.7 por su gran versatilidad, es simple, rápido, ordenado y portable. Y sobre todo tiene una gran aceptación dentro de la comunidad relacionada con la Ciberseguridad.

Los lenguajes que se han tenido en cuenta para el desarrollo han sido:

* + Delphi (Object Pascal)
  + C
  + Go
  + Python

## El Keylogger más corto

¿Cuál sería el código del keylogger más simple?

Pues realmente se puede escribir un keylogger en Python [35] en once líneas de código. Con sus limitaciones, pero totalmente funcional. El código sería el siguiente:

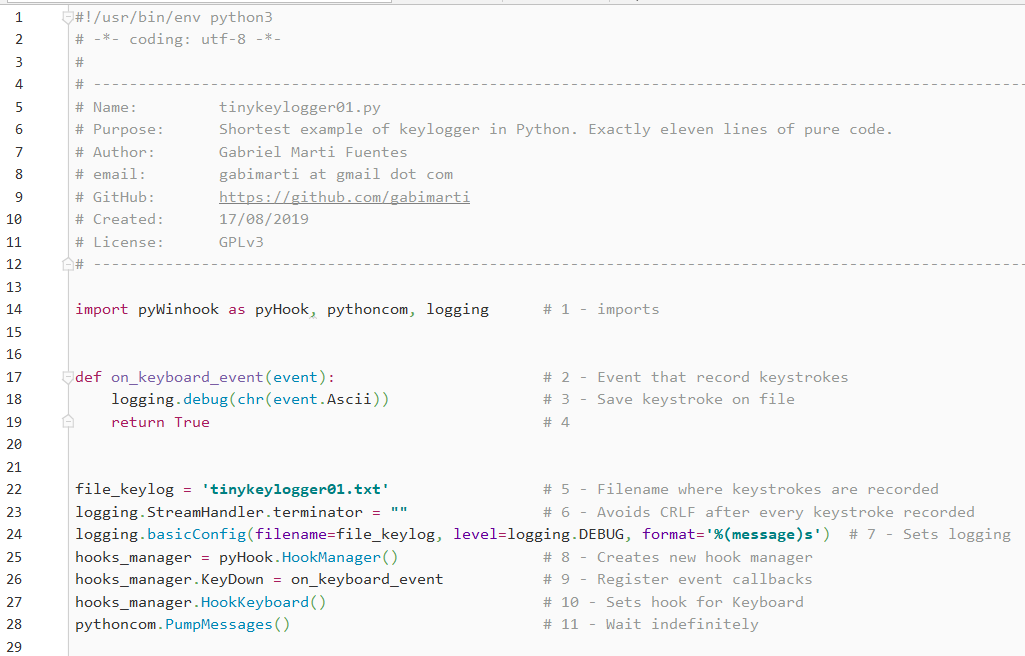


Ilustración 58. Ejemplo de código más corto de keylogger en Python.

Para ello utilizamos tres librerías de Python:

* pyWinHook (anteriormente llamada pyHook), que nos permitirá hacer el ***hook[[43]](#footnote-43)*** con la API del teclado.
* pythoncom, que nos permitirá dejar el programa en espera y procesar los mensajes del sistema y peticiones a la API.
* logging, que provee funciones para interaccionar con el sistema de registro (log) de mensajes del sistema, que suele ser usado para depuración, pero en este caso lo utilizaremos para registrar las teclas pulsadas.

Con esto se genera un archivo de texto planto, con todas las pulsaciones de teclado estándar, pero con la limitación de que caracteres especiales como (borrado hacia atrás) quedan guardados con su código binario y se muestran “extraños” en el archivo de texto, y otros caracteres como letras acentuadas no se registran correctamente, al mismo tiempo que tampoco se reproducen correctamente en pantalla, evidenciando al usuario de que algo mal funciona en el sistema.

El código es casi auto explicativo en sí mismo, aunque se ha comentado cada línea para saber que función hace. Haremos hincapié en tres puntos.

1. La línea 17 (línea 2 de código real). Donde se define la función que procesara cada petición de tecla, donde lo único que hace es coger el carácter correspondiente al código de tecla recibido y guardarlo al fichero de log.
2. La línea 23 (línea 6 de código real) donde establecemos que al final de cada grabación de mensaje de log no se añada el salto de línea (cadena vacía) para poder hacer una mejor lectura del fichero donde se guardan las pulsaciones de teclas.
3. La línea 26 (línea 9 de código real) donde, después de definir un hookManager, le indicamos que para cada estado del teclado con la tecla pulsada (KeyDown) llame a la función on\_keyboard\_event().

De hecho, todas las líneas son importantes e imprescindibles pero estas tres que hemos nombrado son claves a nuestra manera de verlo.

**¿Puede hacerse con un código más corto?**

Si, puede hacerse utilizando una librería diferente. En este caso, usando **pynput**, podemos crear una keylogger básico [36] con 8 líneas de código.



Ilustración 59. Ejemplo de keylogger en ocho líneas de código usando librería pynput.

Con esta librería se soluciona parcialmente un problema que existe en el primer código. El uso de teclas acentuadas, que en el primer caso no es soportado, y aquí quedan registradas correctamente en el log, aunque no funciona correctamente con todos los caracteres especiales con este código resumido, y tenemos el problema añadido que el fichero de texto generado no es tan claro (o legible) como en el primer caso, pues cada tecla queda registrada entrecomillada y sí que aparecen registradas las teclas especiales como “Enter”, “Control”, flechas de desplazamiento, etc.

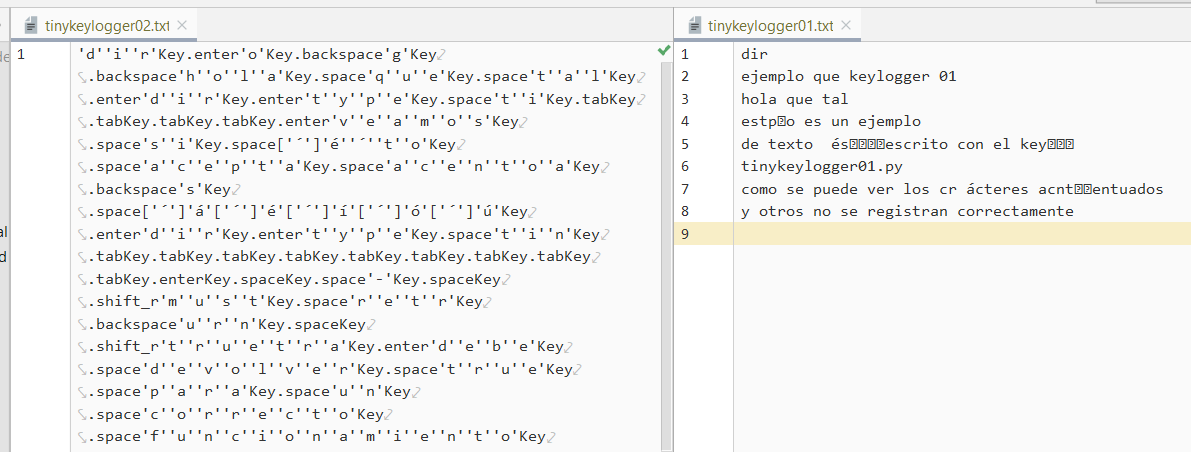


Ilustración 60. Muestra ficheros de texto generados por los dos ejemplos de keylogger.

Cualquiera de los dos ejemplos puede servir de punto de partida para la creación de nuestro keylogger.

## Características del Keylogger

Vamos a basar el desarrollo del keylogger tomando de partida las pruebas iniciales que hemos puesto de ejemplo con los códigos cortos. Este keylogger es basado en la API de Windows, y aunque Python es un lenguaje multiplataforma, el código aquí expuesto está pensado y desarrollado para Windows y para otros entornos es muy probable que necesite cambios o modificaciones.

No vamos a desarrollar un keylogger perfecto, ni siquiera optimizado al límite en cuanto tamaño de código. El objetivo es mostrar su desarrollo y funcionamiento y dotarlo de una serie de características que serán las siguientes:

* Registro de teclas básico (evidentemente, porque es un keylogger)
* Parámetros de configuración en argumentos al cargar el keylogger
* Captura de pantalla periódica cada X pulsaciones (configurable)
* Envío de pulsaciones de teclas a un programa monitor.
* Envío de datos a una cuenta de email
* Envío de fichero con registro de teclas a servicios de paste (pastebin y pastecode)
* Envío de imágenes de capturas de pantalla a canal privado de Telegram usando un chatbot de Telegram, y enlaces de las publicaciones en los servicios de paste.

Para ello desarrollamos dos programas:

* El keylogger en sí mismo.
* El programa para monitorizar el keylogger en una red local.

## Partes importantes del desarrollo

Pendiente --- pendiente --- pendiente --- pendiente ---

## Problemas y soluciones

Pendiente --- pendiente --- pendiente --- pendiente ---

## Implantación

Pendiente --- pendiente --- pendiente --- pendiente ---

## Mejoras posibles

Pendiente --- pendiente --- pendiente --- pendiente ---

## Notas finales

Todo el código fuente y documentación de dicho keylogger puede encontrarse en el siguiente repositorio de GitHub.

<https://github.com/gabimarti/Demons-eye-keylogger>

Pendiente --- pendiente --- pendiente --- pendiente ---

-------------------------- [APARTADO PENDIENTE DE COMPLETAR] -------------------

# Resultados finales

Pendiente --- pendiente --- pendiente --- pendiente ---

…

-------------------------- [APARTADO PENDIENTE DE COMPLETAR] -------------------

# Coste del proyecto

Para evaluar el coste del proyecto se pueden (o deberían) tener en cuenta tres puntos de vista en cuanto a la retribución económica.

* Retribución neta de una persona como trabajador de una empresa.
* Coste para la empresa de la persona que lleva a cabo dicho proyecto.
* Coste para un cliente final que ha encargado dicho proyecto a una empresa.

Para ello nos basaremos en el último supuesto, que sería el coste para una empresa que ha encargado dicho proyecto a una empresa especializada, la cual dispone de trabajadores que han ejecutado este proyecto. Evidentemente es el coste más alto, ya que aquí hay que tener en cuenta los sueldos de las personas, los costes de seguridad social, retenciones irpf, costes de empresa y sumar el beneficio que la cargue cualquier empresa a su trabajo.

Pendiente --- pendiente --- pendiente --- pendiente ---

## Coste temporal

El volumen de trabajo que ha comportado la realización del proyecto se puede clasificar en los siguientes apartados:

* Investigación y Documentación
* Estudio de diferentes keyloggers Hardware y Software
* Reversing keylogger
* Pruebas programación Python pre-desarrollo
* Desarrollo del keylogger
* Redacción de la memoria

A continuación, explicamos detalladamente en qué ha consistido y cuánto tiempo ha requerido cada uno de los apartados mencionados anteriormente:

…

Figura 3. Coste temporal

## Coste económico

El coste económico necesario para el desarrollo del proyecto …

-------------------------- [APARTADO PENDIENTE DE COMPLETAR] -------------------

# Conclusiones

Pendiente --- pendiente --- pendiente --- pendiente ---

Que he visto del *turla*, y que capacidades explotan los ciberdelincuentes

* Que se ha conseguido, y que no se ha conseguido de todo el proyecto.

-------------------------- [APARTADO PENDIENTE DE COMPLETAR] -------------------

# Líneas de futuro

Donde van a ir los keyloggers …..

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber hendrerit an. Qui ut wisi vocibus suscipiantur, quo dicit ridens inciderint id. Quo mundi lobortis reformidans eu, legimus senserit definiebas an eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert percipit cu, eirmod consectetuer signiferumque eu per. In usu latine equidem dolores. Quo no falli viris intellegam, ut fugit veritus placerat per:

Enlaces útiles para las líneas de futuro:

<https://blog.trendmicro.com/trendlabs-security-intelligence/keeping-a-hidden-identity-mirai-ccs-in-tor-network/>

…

-------------------------- [APARTADO PENDIENTE DE COMPLETAR] -------------------

# Referencias

1. LA HISTÓRIA (QUE SE CONOCE) DE LOS ATAQUES INFORMÁTICOS REALIZADOS POR EL FBI.

<https://r3d.mx/2016/05/18/la-historia-que-se-conoce-de-los-ataques-informaticos-realizados-por-el-fbi/>

1. SPYING ON THE MOB: UNITED STATES V. SCARFO – A CONSTITUTIONAL ANALYSIS. Nathan E. Carrell.

<http://illinoisjltp.com/journal/wp-content/uploads/2013/10/Carrell.pdf>

1. THE ULTIMATE KEYLOGGER: FBI’S MAGIC LANTERN.

<https://www.mobistealth.com/blog/ultimate-keylogger-magic-lantern/>

1. FBI SOFTWARE CRACKS ENCRYPTION WALL.

<http://www.nbcnews.com/id/3341694/ns/technology_and_science-security/t/fbi-software-cracks-encryption-wall/>

1. THE CASE FOR MAGIC LANTERN: SEPTEMBER 11 HIGHLIGHTS THE NEED FOR INCREASED SURVEILLANCE. Christopher Woo & Miranda So. Harvard Journal of Law & Technology.

<http://jolt.law.harvard.edu/articles/pdf/v15/15HarvJLTech521.pdf>

1. NEW FBI DOCUMENTS PROVIDE DETAILS ON GOVERNMENT’S SURVEILLANCE SPYWARE.

<https://www.eff.org/es/deeplinks/2011/04/new-fbi-documents-show-depth-government>

1. Creador y Código fuente del primer keylogger por software. Perry Kivolowitz.

<http://pages.cs.wisc.edu/~perryk/>  
<http://securitydigest.org/unix/archive/006>

1. Student Expelled for Using Hardware Keylogger to Hack School, Change Grades. <https://www.bleepingcomputer.com/news/security/student-expelled-for-using-hardware-keylogger-to-hack-school-change-grades/>
2. Boston College Student Steals Around $2,000 with Keylogger. <https://www.neowin.net/news/boston-college-student-steals-around-2000-with-keylogger>
3. NOTICIAS Y ENLACES RELACIONADOS CON EL KEYLOGGER EN LOS PORTATILES HP.

<https://www.techrepublic.com/article/built-in-keylogger-found-in-hp-laptops-again/>  
<https://mashable.com/2017/12/11/keylogger-found-on-hundreds-of-hp-computer-models/>  
<https://zwclose.github.io/HP-keylogger/>  
<https://support.hp.com/us-en/document/c05827409>

1. Keyloggers: How they work and how to detect them (Part 1).

<https://securelist.com/keyloggers-how-they-work-and-how-to-detect-them-part-1/36138/>

1. Keyloggers: Implementing keyloggers in Windows. Part Two.   
   <https://securelist.com/keyloggers-implementing-keyloggers-in-windows-part-two/36358/>
2. 'Sleeper bugs' used to steal 1m in France

<https://www.theguardian.com/technology/2006/feb/07/news.france>

1. TouchLogger: Inferring Keystrokes On Touch Screen From Smartphone Motion. Hao Chen. <https://web.cs.ucdavis.edu/~hchen/paper/hotsec2011.pdf>
2. Escándalo en Israel por varios casos de espionaje industrial mediante virus.  
   <https://www.elmundo.es/navegante/2005/05/30/empresas/1117448921.html>
3. Police: Hackers stole Israeli defense data.  
   <http://www.chinadaily.com.cn/english/doc/2005-06/04/content_448599.htm>
4. El mayor atraco del siglo: los ciberdelincuentes roban mil millones de dólares. <https://www.kaspersky.es/blog/el-mayor-atraco-del-siglo-los-hackers-roban-mil-millones-de-dolares/5370/>
5. The Great Bank Robbery: the Carbanak APT.   
   <https://securelist.com/the-great-bank-robbery-the-carbanak-apt/68732/>
6. Análisis del código fuente de Carbanak.   
   <https://unaaldia.hispasec.com/2019/04/analisis-del-codigo-fuente-de-carbanak.html>
7. Exam cheat jailed for hacking into university computer system.   
   <https://www.telegraph.co.uk/news/uknews/law-and-order/11560173/Exam-cheat-jailed-for-hacking-into-university-computer-system.html>
8. SUCURI. Cloudflare[.]Solutions Keylogger on Thousands of Infected WordPress Sites. <https://blog.sucuri.net/2017/12/cloudflare-solutions-keylogger-on-thousands-of-infected-wordpress-sites.html>
9. SUCURI. Cloudflare[.]solutions Keylogger Returns on New Domains. <https://blog.sucuri.net/2018/01/cloudflare-solutions-keylogger-returns-on-new-domains.html>
10. DIRECTIVA 2013/40/UE DEL PARLAMENTO EUROPEO Y DEL CONSEJO.

<https://eur-lex.europa.eu/legal-content/ES/TXT/PDF/?uri=CELEX:32013L0040&qid=1442911632730&from=EN>

1. Agencia Estatal Boletín Oficial del Estado. Ley Orgánica 1/2015, de 30 de marzo, por la que se modifica la Ley Orgánica 10/1995, de 23 de noviembre, del Código Penal.  
   <https://www.boe.es/diario_boe/txt.php?id=BOE-A-2015-3439>
2. Agencia Estatal Boletín Oficial del Estado. Código de Derecho de la Ciberseguridad.  
   <https://www.boe.es/legislacion/codigos/codigo.php?id=173_Codigo_de_Derecho__de_la_Ciberseguridad>
3. BOE. Real Decreto Legislativo 2/2015, de 23 de octubre, por el que se aprueba el texto refundido de la Ley del Estatuto de los Trabajadores.

<https://www.boe.es/buscar/pdf/2015/BOE-A-2015-11430-consolidado.pdf>

1. Agencia Estatal Boletín Oficial del Estado. Real Decreto de 14 de septiembre de 1882 por el que se aprueba la Ley de Enjuiciamiento Criminal.

<https://www.boe.es/buscar/act.php?id=BOE-A-1882-6036&tn=2>

1. …
2. …
3. …
4. Información y esquemas para realizar un keylogger hardware básico.   
   <http://www.keelog.com/diy-hardware-keylogger/>
5. …
6. Keystroke Logging. Wikipedia. <https://en.wikipedia.org/wiki/Keystroke_logging>
7. The Blue Pill virtualization attack.   
   <https://searchservervirtualization.techtarget.com/tip/The-Blue-Pill-virtualization-attack-Virtual-machine-malware>
8. Información sobre binario de Turla Keylogger en Virus Total.  
   <https://www.virustotal.com/gui/file/b7b5d28be983c774ef83a8960a68134732a79818c572e8800cea6428f27fb114/community>
9. The Epic Turla Operation. <https://securelist.com/the-epic-turla-operation/65545/>
10. Descarga de muestra binario de Turla Keylogger.

<https://github.com/mstfknn/malware-sample-library/blob/master/Turla/TurlaKeylogger.bin>

1. Descarga de Ghidra. <https://github.com/NationalSecurityAgency/ghidra>
2. Descarga de Radare. <https://rada.re/r/>
3. Descarga de Cutter. <https://cutter.re/>
4. Descarga de x64dbg. <https://x64dbg.com/#start>
5. Documentación de la librería pythoncom para Python.  
   <http://timgolden.me.uk/pywin32-docs/pythoncom.html>
6. …
7. Repositorio de GitHub con el código fuente del keylogger desarrollado. <https://github.com/gabimarti/Demons-eye-keylogger>
8. Repositorio de GitHub con el documento de este TFM.   
   <https://github.com/gabimarti/TFM-Analisis-Keyloggers-Ingenieria-Inversa>
9. …

# Referencias

|  |  |
| --- | --- |
| [1] | K. Collins, «Today we’re worried about smart TVs, but in the 1980s Russian spies were hacking typewriters,» 16 Marzo 2017. [En línea]. Available: https://qz.com/932448/forget-smart-tvs-in-the-1980s-spies-were-hacking-typewriters/. |
| [2] | Crypto Museum, «IBM Selectric Bug. Operation GUNMAN - how the Soviets bugged IBM typewriters,» [En línea]. Available: https://www.cryptomuseum.com/covert/bugs/selectric/. |
| [3] | Center for Cryptologic History, National Security Agency, «Learning from the Enemy: The GUNMAN Project,» 2012. [En línea]. Available: http://www.foo.be/docs/intelligence/Learning\_From\_the\_Enemy\_The\_GUNMAN\_Project.pdf. |
| [4] | S. A. Maneki, «(U) Learning from the Enemy: The GUNMAN Project.,» 8 Enero 2007. [En línea]. Available: https://www.nsa.gov/Portals/70/documents/news-features/declassified-documents/cryptologic-histories/Learning\_from\_the\_Enemy.pdf. |
| [5] | «Perry Kivolowitz,» [En línea]. Available: http://pages.cs.wisc.edu/~perryk/. |
| [6] | «The 'Security Digest' Archives (TM). DOCUMENT: Unix 'Security Mailing List',» [En línea]. Available: http://securitydigest.org/unix/archive/006. |
| [7] | «La historia (que se conoce) de los ataques informáticos realizados por el FBI,» 18 Mayo 2016. [En línea]. Available: https://r3d.mx/2016/05/18/la-historia-que-se-conoce-de-los-ataques-informaticos-realizados-por-el-fbi/. |
| [8] | N. E. Carrell, «SPYING ON THE MOB: UNITED STA TES V. SCARFO - A CONSTITUTIONAL ANALYSIS,» 10 2013. [En línea]. Available: http://illinoisjltp.com/journal/wp-content/uploads/2013/10/Carrell.pdf. |
| [9] | B. Sullivan, «FBI software cracks encryption wall,» 2001 Noviembre 2001. [En línea]. Available: www.nbcnews.com/id/3341694/ns/technology\_and\_science-security/t/fbi-software-cracks-encryption-wall/. |
| [10] | C. W. &. M. So, «THE CASE FOR MAGIC LANTERN: SEPTEMBER 11 HIGHLIGHTS THE NEED FOR INCREASED SURVEILLANCE,» 2002. [En línea]. Available: http://jolt.law.harvard.edu/articles/pdf/v15/15HarvJLTech521.pdf. |
| [11] | J. Lynch, «New FBI Documents Provide Details on Government’s Surveillance Spyware,» 29 Abril 2011. [En línea]. Available: https://www.eff.org/es/deeplinks/2011/04/new-fbi-documents-show-depth-government. |
| [12] | «Boston College Student Steals Around $2,000 with Keylogger,» 7 Febrero 2003. [En línea]. Available: https://www.neowin.net/news/boston-college-student-steals-around-2000-with-keylogger. |
| [13] | «Perfil de LinkedIn de Michael Haephrati. Target Eye Limited.,» [En línea]. Available: https://www.linkedin.com/in/haephrati/. |
| [14] | «Hao Chen, Professor, Department of Computer Science, University of California, Davis,» [En línea]. Available: https://web.cs.ucdavis.edu/~hchen/. |
| [15] | H. Chen, «TouchLogger: Inferring Keystrokes On Touch Screen From Smartphone Motion,» 2011. [En línea]. Available: https://web.cs.ucdavis.edu/~hchen/paper/hotsec2011.pdf. |
| [16] | SUCURI, «Cloudflare[.]Solutions Keylogger on Thousands of Infected WordPress Sites,» 6 Diciembre 2017. [En línea]. Available: https://blog.sucuri.net/2017/12/cloudflare-solutions-keylogger-on-thousands-of-infected-wordpress-sites.html. |
| [17] | SUCURI, «Cloudflare[.]solutions Keylogger Returns on New Domains,» 24 Enero 2018. [En línea]. Available: https://blog.sucuri.net/2018/01/cloudflare-solutions-keylogger-returns-on-new-domains.html. |
| [18] | Diario Oficial de la Unión Europea, «DIRECTIVA 2013/40/UE DEL PARLAMENTO EUROPEO Y DEL CONSEJO,» 12 Agosto 2013. [En línea]. Available: https://eur-lex.europa.eu/legal-content/ES/TXT/PDF/?uri=CELEX:32013L0040&qid=1442911632730&from=EN. |
| [19] | Jefatura del Estado. Gobierno de España, «Ley Orgánica 1/2015, de 30 de marzo, por la que se modifica la Ley Orgánica 10/1995, de 23 de noviembre, del Código Penal.,» 31 Marzo 2015. [En línea]. Available: https://www.boe.es/diario\_boe/txt.php?id=BOE-A-2015-3439. |
| [20] | KeeLog, «Keylogger por Hardware Open Source DIY,» [En línea]. Available: https://www.keelog.com/es/diy-hardware-keylogger/. |
| [21] | KeeLog, «Keylogger Inalámbrico - ¡Hazlo tú mismo!,» [En línea]. Available: http://www.keelog.com/es/wireless-keylogger/. |
| [22] | J. Rutkowska, «Introducing Blue Pill,» 22 Junio 2006. [En línea]. Available: https://blog.invisiblethings.org/2006/06/22/introducing-blue-pill.html. |
| [23] | Microsoft Windows Dev Center, «Documentación SetWindowsHookEx,» 12 Mayo 2018. [En línea]. Available: https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-setwindowshookexa. |
| [24] | Microsoft Windows Dev Center, «Documentación UnhookWindowsHookEx,» 12 Mayo 2018. [En línea]. Available: https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-unhookwindowshookex. |
| [25] | Adlice Software, «KernelMode Rootkits: Part 3, kernel filters,» 10 Julio 2014. [En línea]. Available: https://www.adlice.com/kernelmode-rootkits-part-3-kernel-filters/. |
| [26] | atorralb, «javascript keylogger,» [En línea]. Available: https://gist.github.com/atorralb/fbc82f70e1e2c1ebc67f. |
| [27] | CyberPoint, «Logging Keystrokes with Event Tracing for Windows (ETW),» 22 Octubre 2016. [En línea]. Available: https://www.cyberpointllc.com/srt/posts/srt-logging-keystrokes-with-event-tracing-for-windows-etw.html. |
| [28] | M. Chehab, «CSS Keylogger,» 21 Febrero 2018. [En línea]. Available: https://github.com/maxchehab/CSS-Keylogging. |
| [29] | T. Gersic, «tomgersic/AndroidKeyLogger,» 1 Agosto 2012. [En línea]. Available: https://github.com/tomgersic/AndroidKeyLogger. |
| [30] | Dmitri Asonov, Rakesh Agrawal, IBM Almaden Research Center, «Keyboard Acoustic Emanations,» 2004. [En línea]. Available: https://www.davidsalomon.name/CompSec/auxiliary/KybdEmanation.pdf. |
| [31] | Li Zhuang, Feng Zhou, J. D. Tygar. University of California, Berkeley, «Keyboard Acoustic Emanations Revisited,» 2005. [En línea]. Available: https://people.eecs.berkeley.edu/~tygar/papers/Keyboard\_Acoustic\_Emanations\_Revisited/tiss.preprint.pdf. |
| [32] | Martin Vuagnoux, Sylvain Pasini. LASEC/EPFL, «Compromising Electromagnetic Emanations of Wired and Wireless Keyboards,» Agosto 2009. [En línea]. Available: https://www.usenix.org/legacy/events/sec09/tech/full\_papers/vuagnoux.pdf. |
| [33] | S. Kamkar, «KeySweeper, a stealthy Arduino-based device,» 12 Enero 2015. [En línea]. Available: https://samy.pl/keysweeper/. |
| [34] | Nora Codes, «PDF Embedding Attacks,» 04 Agosto 2018. [En línea]. Available: https://nora.codes/post/pdf-embedding-attacks/. |
| [35] | M. Nelson, «The Tale of SettingContent-ms Files,» 11 Junio 2018. [En línea]. Available: https://posts.specterops.io/the-tale-of-settingcontent-ms-files-f1ea253e4d39. |
| [36] | Microsoft. Centro para desarrolladores de Office., «Introducción a VBA en Office,» 14 Agosto 2019. [En línea]. Available: https://docs.microsoft.com/es-es/office/vba/library-reference/concepts/getting-started-with-vba-in-office. |
| [37] | Microsoft, «Descripción de los comportamientos de macros AutoExec y AutoOpen en Word,» [En línea]. Available: https://support.microsoft.com/es-es/help/286310/description-of-behaviors-of-autoexec-and-autoopen-macros-in-word. |
| [38] | Microsoft, «Macro malware,» 10 Agosto 2019. [En línea]. Available: https://docs.microsoft.com/en-us/windows/security/threat-protection/intelligence/macro-malware. |
| [39] | F. Viggiani, «Embedding EXE files into PowerShell scripts,» 15 03 2016. [En línea]. Available: https://truesecdev.wordpress.com/2016/03/15/embedding-exe-files-into-powershell-scripts/. |
| [40] | KeeLog, «AirDrive Forensic Keylogger Module,» [En línea]. Available: http://www.keelog.com/airdrive-forensic-keylogger-module-usb-hardware-keylogger-module-with-wifi-and-flash/. |
| [41] | KeeLog, «Forensic Keylogger Keyboard,» [En línea]. Available: http://www.keelog.com/keylogger-keyboard/. |
| [42] | Wikipedia, «Cifrado XOR,» [En línea]. Available: https://es.wikipedia.org/wiki/Cifrado\_XOR. |
| [43] | KeeLog, «KeeLog KeyGrabber Pico 16Mb,» [En línea]. Available: http://www.keelog.com/keygrabber-pico-usb-16mb-tiny-usb-hardware-keylogger-with-16mb-flash-drive/. |
| [44] | Maltronics, «Maltronics WiFi KeyLogger PRO,» [En línea]. Available: https://maltronics.com/collections/wifi-keyloggers/products/wifi-keylogger-pro. |
| [45] | Microsoft Windows Dev Center, «IsDebuggerPresent function,» [En línea]. Available: https://docs.microsoft.com/en-us/windows/win32/api/debugapi/nf-debugapi-isdebuggerpresent. |
| [46] | «Virus Total,» [En línea]. Available: https://www.virustotal.com/. |
| [47] | «Hybrid Analysis,» [En línea]. Available: https://www.hybrid-analysis.com/. |
| [48] | «INTEZER Analyze,» [En línea]. Available: https://analyze.intezer.com. |
| [49] | Microsoft, «Visual Studio,» [En línea]. Available: https://visualstudio.microsoft.com/es/. |
| [50] | «Ghidra,» [En línea]. Available: https://ghidra-sre.org/. |
| [51] | «Radare,» [En línea]. Available: https://www.radare.org/r/. |
| [52] | «Cutter - the QT Gui for r2,» [En línea]. Available: https://radare.org/cutter/. |
| [53] | «x64dbg An open-source x64/x32 debugger for windows.,» [En línea]. Available: https://x64dbg.com/#start. |
| [54] | G. Martí, «tinykeylogger01.py,» 17 08 2019. [En línea]. Available: https://github.com/gabimarti/Demons-eye-keylogger/blob/master/test/tinykeylogger01.py. |
| [55] | G. Martí, «tinykeylogger02.py,» 17 08 2019. [En línea]. Available: https://github.com/gabimarti/Demons-eye-keylogger/blob/master/test/tinykeylogger02.py. |

Índice de Ilustraciones

[Ilustración 1. Dos modelos de keylogger hardware. Para teclado ps/2 y USB. 4](#_Toc17660111)

[Ilustración 2. IBM Selectric II (1971-1981) 7](#_Toc17660112)

[Ilustración 3. Portada del documento de la NSA, “The GUNMAN Project”. Desclasificado en Diciembre de 2011 8](#_Toc17660113)

[Ilustración 4. Bola “pelota de golf” de la IBM Selectric, con los caracteres. 9](#_Toc17660114)

[Ilustración 5. Muestra de una sección del perfil de LinkedIn de Michael Haephrati. Detenido en 2005 por la creación de un keylogger. 11](#_Toc17660115)

[Ilustración 6. Código del keylogger inyectado después de su descifrado. 12](#_Toc17660116)

[Ilustración 7. Diario Oficial de la Unión Europea. DIRECTIVA 2013/40/UE. 12 agosto 2013. Extracto del contenido relativo a los ataques contra los sistemas de información. 14](#_Toc17660117)

[Ilustración 8. Ejemplo de aviso legal sobre el uso del Keylogger, de un vendedor internacional. 15](#_Toc17660118)

[Ilustración 9. Extracto del Artículo 20 del Estatuto de los Trabajadores. 17](#_Toc17660119)

[Ilustración 10. Extracto del artículo 197 ter, de la Ley Orgánica 1/2015, de 30 de marzo 18](#_Toc17660120)

[Ilustración 11. Esquema de circuito de keylogger básico usando la antigua conexión PS/2. 22](#_Toc17660121)

[Ilustración 12. Esquema de funcionamiento del mecanismo de *hooks* en Windows. 24](#_Toc17660122)

[Ilustración 13. Código fuente de un keylogger en Javascript. 25](#_Toc17660123)

[Ilustración 14. Código fuente en PHP del backend del keylogger que registra las pulsaciones. 25](#_Toc17660124)

[Ilustración 15. Muestra de parte de archivo CSS para crear un keylogger. 26](#_Toc17660125)

[Ilustración 16. Patrones típicos de ángulos y balanceo cuando se presionan diferentes teclas de dígitos. 27](#_Toc17660126)

[Ilustración 17. Ejemplo de método modificado en Android SDK para crear un keylogger. 28](#_Toc17660127)

[Ilustración 18. Muestra de los circuitos incorporados en el keylogger KeySweeper. 30](#_Toc17660128)

[Ilustración 19. Script que ejecutará el malware incrustado una vez abierto el PDF. 33](#_Toc17660129)

[Ilustración 20. Muestra del texto del párrafo anterior codificado en Base64. 34](#_Toc17660130)

[Ilustración 21. Muestra de keylogger hardware instalado en el interior de un teclado. 35](#_Toc17660131)

[Ilustración 22. Foto de los keyloggers hardware analizados 36](#_Toc17660132)

[Ilustración 23. Vista en detalle de los dos keyloggers 36](#_Toc17660133)

[Ilustración 24. KeyGrabber en su empaquetado original. 38](file:///D:\DATA\Cloud\Dropbox\La%20Salle%20-%20Master%20Ciberseguretat%20-%202018-2019\MCS%2099%20-%20TFM%20(Treball%20Fi%20Master)\TFM%20Keyloggers\TFM_Análisis%20de%20Keyloggers%20mediante%20Ingenieria%20Inversa.docx#_Toc17660134)

[Ilustración 25. KeyGrabber conectado al teclado. 38](file:///D:\DATA\Cloud\Dropbox\La%20Salle%20-%20Master%20Ciberseguretat%20-%202018-2019\MCS%2099%20-%20TFM%20(Treball%20Fi%20Master)\TFM%20Keyloggers\TFM_Análisis%20de%20Keyloggers%20mediante%20Ingenieria%20Inversa.docx#_Toc17660135)

[Ilustración 26. Ejemplo de fichero de LOG con el contenido capturado por el keylogger. 39](#_Toc17660136)

[Ilustración 27. Vista del SSID activo del keylogger analizado. 40](#_Toc17660137)

[Ilustración 28. Pantalla de configuración del punto de acceso Wifi del keylogger. 41](#_Toc17660138)

[Ilustración 29. Pantalla de configuración del registro de teclas. 41](#_Toc17660139)

[Ilustración 30. Muestra del contenido registrado por el keylogger accedido a través de la conexión WiFi. 42](#_Toc17660140)

[Ilustración 31. Ejemplo de recepción de las teclas del keylogger en tiempo real mediante data streaming. 43](#_Toc17660141)

[Ilustración 32. Muestra de PE-Bear. Herramienta para analizar cabeceras PE de ejecutables. 50](#_Toc17660142)

[Ilustración 33. Muestra parcial del desensamblado del binario con la herramienta PE-Bear. 51](#_Toc17660143)

[Ilustración 34. Muestra de cadenas de texto encontradas en el binario. 52](#_Toc17660144)

[Ilustración 35. Detección de Turla Keylogger por Antivirus de Microsoft. 52](#_Toc17660145)

[Ilustración 36. Muestra de la información ofrecida en Virus Total. 53](#_Toc17660146)

[Ilustración 37. Muestra ejecución del binario en entorno virtual de Hybrid Analysis. 54](#_Toc17660147)

[Ilustración 38. Muestra de la información ofrecida por INTEZER Analyze. 54](#_Toc17660148)

[Ilustración 39. Muestra de binarios que tienen partes de coincidencia en el código. 55](#_Toc17660149)

[Ilustración 40. Ejemplo de punteros a zonas de memoria vacías para rellenar con datos. 55](#_Toc17660150)

[Ilustración 41. Muestra de bloque de código donde hay repetidas operaciones XOR. 56](#_Toc17660151)

[Ilustración 42. Muestra parcial de la pantalla de Ghidra una vez cargado el binario a analizar. 57](#_Toc17660152)

[Ilustración 43. Muestra parcial de radare2 junto con Cutter una vez cargado el binario a analizar. 58](#_Toc17660153)

[Ilustración 44. Muestra donde se observa una de las medidas de protección usadas. 59](#_Toc17660154)

[Ilustración 45. Función que realiza repetidas llamadas a la dirección 0x180001750 59](#_Toc17660155)

[Ilustración 46. Código en C resultante del código en ensamblador anterior. 60](#_Toc17660156)

[Ilustración 47. Visualización en modo gráfico de la subrutina de descifrado. 61](#_Toc17660157)

[Ilustración 48. Código en C resultante de la función de descifrado. 62](#_Toc17660158)

[Ilustración 49. Muestra de gráfico de llamadas entre funciones. 62](#_Toc17660159)

[Ilustración 50. Sección de la función principal del binario. 63](#_Toc17660160)

[Ilustración 51. Sección de la segunda función de descifrado. 63](#_Toc17660161)

[Ilustración 52. Fragmento de código donde se muestra el punto de entrada del binario a analizar. 64](#_Toc17660162)

[Ilustración 53. Parte del mapa de memoria del binario cargado en x64dbg. 65](#_Toc17660163)

[Ilustración 54. Función que realiza múltiples llamadas para decodificar datos de memoria. 65](#_Toc17660164)

[Ilustración 55. Muestra de zona de memoria donde se dejan los datos una vez decodificados. 65](#_Toc17660165)

[Ilustración 56. Muestra de parte del código del segundo proceso de descifrado. 66](#_Toc17660166)

[Ilustración 57. Datos descifrados con los nombres de las funciones de la API de Windows. 66](#_Toc17660167)

[Ilustración 58. Ejemplo de código más corto de keylogger en Python. 69](#_Toc17660168)

[Ilustración 59. Ejemplo de keylogger en ocho líneas de código usando librería pynput. 70](#_Toc17660169)

[Ilustración 60. Muestra ficheros de texto generados por los dos ejemplos de keylogger. 71](#_Toc17660170)

Índice de tablas

1. Un *keylogger* es un programa (o dispositivo hardware) que se encarga de registrar todas las pulsaciones de teclas que realiza el usuario y las almacena en un fichero de datos. Este fichero se almacena oculto en el propio equipo (o en el interior del dispositivo), y habitualmente suele ser enviado posteriormente por internet a un servidor. [↑](#footnote-ref-1)
2. En el campo de seguridad informática, la infraestructura mando y control (Command and Control en inglés, usualmente abreviado C&C o C2) consta de servidores y otros elementos que son usados para controlar remotamente el malware. [↑](#footnote-ref-2)
3. La Guerra Fría fue un enfrentamiento político, económico, social, militar, informativo y científico iniciado tras finalizar la Segunda Guerra Mundial entre el bloque Occidental (occidental-capitalista) liderado por Estados Unidos, y el bloque del Este (oriental-comunista) liderado por la Unión Soviética. [↑](#footnote-ref-3)
4. La máquina de escribir IBM Selectric (conocida también como la IBM de bola) fue un influyente diseño de máquina de escribir eléctrica de IBM, cuyo primer modelo salió al mercado en 1961. [↑](#footnote-ref-4)
5. Leningrado, conocida hoy en día como San Petersburgo. [↑](#footnote-ref-5)
6. Lincoln Faurer, Director de la NSA, creó el “Proyecto Gunman” porque no confiaba que, ni en el Departamento de estado, ni la CIA, manejaran correctamente el asunto. Fue aprobado por Ronald Reagan en febrero de 1984. [↑](#footnote-ref-6)
7. Los pequeños dispositivos que son escondidos y utilizados para grabar o interceptar conversaciones o comunicaciones son conocidos como “bugs” (bichos o insectos, si traducimos literalmente). [↑](#footnote-ref-7)
8. No se sabe exactamente, pero algunas conjeturas apuntan a que los rusos solo tenían acceso a tecnología digital de 4 bits en ese momento, y por ese motivo comprimían los datos de 6 bits a 4 bits. [↑](#footnote-ref-8)
9. Usenet es el acrónimo de Users Network (Red de usuarios), consistente en un sistema global de discusión en Internet, que evolucionó de las redes UUCP. [↑](#footnote-ref-9)
10. CIPAV (Computer and Internet Protocol Address Verifier), es un software desarrollado por el FBI en el año 2007, con el objetivo de interceptar y detener hackers, extorsionadores, usuarios de pornografía infantil y delincuentes en general. En 2013 se consideró ilegal el uso de CIPAV. [↑](#footnote-ref-10)
11. Mydoom, también conocido como W32.MyDoom@mm, Novarg, MiMail.R y "Shimgapi", fue un gusano informático que afecta a sistemas Windows. Su velocidad de propagación superó los registros anteriores establecidos por el gusano Sobig y ILoveYou. [↑](#footnote-ref-11)
12. El concepto de malware procede de la expresión inglesa “malicious software”. Se trata de un software malicioso: es decir, de un programa informático cuya finalidad es provocar un daño en un sistema. [↑](#footnote-ref-12)
13. WordPress es un sistema de gestión de contenidos, enfocado a la creación de cualquier tipo de página web. Originalmente alcanzó una gran popularidad en la creación de blogs, para convertirse con el tiempo en una de las principales herramientas para la creación de páginas web comerciales. [↑](#footnote-ref-13)
14. JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, ​ basado en prototipos, imperativo, débilmente tipado y dinámico. [↑](#footnote-ref-14)
15. Ciberdelincuente. Persona que realiza actividades delictivas en internet como robar información, acceder a redes privadas, estafas, y todo lo que tiene que ver con los delitos e ilegalidad. [↑](#footnote-ref-15)
16. Hacker es un término ya aceptado por la RAE. <https://dej.rae.es/lema/hacker>

    Persona experta en alguna rama tecnológica que accede a un sistema informático o a informaciones ubicadas en dicho sistema o en la red de comunicaciones (bases de datos, programas informáticos, etc.) sin permiso del titular y sin necesidad de móvil o acción posterior alguna. [↑](#footnote-ref-16)
17. Red Team es un grupo independiente de ciberatacantes que reta a una organización para que pueda mejorar su efectividad. Actúa de la misma forma que lo haría un enemigo real y utiliza capacidades similares con el objetivo de encontrar las puertas de entrada vulnerables y de mejorar su seguridad. [↑](#footnote-ref-17)
18. Phishing, conocido como suplantación de identidad, es un término informático que denomina un modelo de abuso informático y que se comete mediante el uso de un tipo de ingeniería social, con la intención de engañar al usuario para que introduzca datos personales, contraseñas o ejecute una aplicación maliciosa. El cibercriminal, conocido como phisher, se hace pasar por una persona o empresa de confianza en una aparente comunicación oficial electrónica, por lo común un correo electrónico, o algún sistema de mensajería instantánea o incluso utilizando también llamadas telefónicas. [↑](#footnote-ref-18)
19. La interfaz de programación de aplicaciones, conocida también por la sigla API, en inglés, application programming interface,​ es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. [↑](#footnote-ref-19)
20. El puerto PS/2 toma su nombre de la serie de computadoras IBM Personal System/2 que es creada por IBM en 1987, y empleada para conectar teclados y ratones. Posteriormente fue adoptado como estándar para todo el mercado PC. [↑](#footnote-ref-20)
21. El Bus Universal en Serie (en inglés: Universal Serial Bus), conocido por la sigla USB, es un bus de comunicaciones que sigue un estándar que define los cables, conectores y protocolos para conectar, comunicar y proveer de alimentación eléctrica entre computadoras, periféricos y dispositivos electrónicos. [↑](#footnote-ref-21)
22. Blue Pill es el nombre en clave para un rootkit basado en la virtualización x86. Soporta virtualización AMD-V (Pacifica), y también Intel VT-x (Vanderpool). Fue diseñado por Joanna Rutkowska y se demostró en Black Hat Briefings el 3 de agosto de 2006, con una implementación de referencia para el kernel de Microsoft Windows Vista. El nombre es una referencia al concepto de píldora roja y píldora azul de la película de 1999, The Matrix. [↑](#footnote-ref-22)
23. Un rootkit es un conjunto de software que permite un acceso de privilegio continuo a un ordenador y que mantiene su presencia activamente oculta al control de los administradores al corromper el funcionamiento normal del sistema operativo. [↑](#footnote-ref-23)
24. El Kernel es el módulo central, o núcleo, del Sistema Operativo. Es la parte del sistema operativo que se carga primero, y permanece en la memoria principal. Es el responsable de la administración de la memoria, la administración de procesos y tareas, y la administración del disco. El código del kernel generalmente se carga en un área protegida de la memoria para evitar que los programas u otras partes del sistema operativo lo sobrescriban. [↑](#footnote-ref-24)
25. Zeus, ZeuS o Zbot es un malware, troyano, que se ejecuta en versiones de Microsoft Windows. Se ha utilizado principalmente para robar información bancaria mediante el registro de teclas del navegador y la captura de formularios. También se ha usado para instalar el ransomware CryptoLocker. [↑](#footnote-ref-25)
26. SpyEye es un malware que ataca a los usuarios de los navegadores web Google Chrome, Firefox, Internet Explorer y Opera en el sistema operativo Microsoft Windows. Este malware utiliza el registro de pulsaciones de teclas y la captura de formularios para obtener credenciales con el objetivo de robar dinero de cuentas bancarias. [↑](#footnote-ref-26)
27. Man-in-the-browser (MITB, MitB, MIB, MiB), es una técnica relacionada con *man-in-the-middle* (MITM). Es un troyano proxy que infecta un navegador web aprovechando alguna vulnerabilidad del navegador, que le permite modificar páginas web, modificar el contenido de la transacción o insertar transacciones adicionales, todo de forma completamente oculta, e invisible tanto para el usuario como para la aplicación web del host. [↑](#footnote-ref-27)
28. Arduino es una compañía de desarrollo de software y hardware de fuente abierta, así como una comunidad internacional que diseña y manufactura placas de desarrollo de hardware para construir dispositivos digitales y dispositivos interactivos que puedan detectar y controlar objetos del mundo real. [↑](#footnote-ref-28)
29. La nomenclatura GSM proviene de las palabras Global System for Mobile comunications. Es un estándar de comunicación para la telefonía móvil, implementado mediante la combinación de satélites y antenas terrestres. A los móviles que usan la tecnología GSM también se les conoce por móviles 2g o de segunda generación. [↑](#footnote-ref-29)
30. Un Dropper es un software diseñado para instalar algún tipo de malware (virus, puertas traseras, etc.) en el sistema operativo donde ha sido ejecutado de manera oculta y sin que el usuario, ni los sistemas de protección del sistema, se aperciban de ello. [↑](#footnote-ref-30)
31. La carga útil (payload en inglés) es el conjunto de datos transmitidos que es en realidad el mensaje enviado. La carga útil excluye las cabeceras o metadatos, que son enviados simplemente para facilitar la entrega del mensaje. En el ámbito del malware, la carga útil es el código malicioso. [↑](#footnote-ref-31)
32. Exploit es una palabra inglesa que significa explotar o aprovechar, y que en el ámbito de la informática es un fragmento de software, fragmento de datos o secuencia de comandos o acciones, utilizada con el fin de aprovechar una vulnerabilidad de seguridad de un sistema de información para conseguir un comportamiento no deseado del mismo. [↑](#footnote-ref-32)
33. PDF (sigla del inglés Portable Document Format, «formato de documento portátil») es un formato de almacenamiento para documentos digitales independiente de plataformas de software o hardware. Este formato es de tipo compuesto (imagen vectorial, mapa de bits y texto). Fue inicialmente desarrollado por la empresa Adobe Systems, oficialmente lanzado como un estándar abierto el 1 de julio de 2008 y publicado por la Organización Internacional de Estandarización (ISO) como ISO 32000-1. [↑](#footnote-ref-33)
34. Actualmente, si un sistema Windows 10 esta actualizado con todos los parches de seguridad estará protegido contra este método. [↑](#footnote-ref-34)
35. Microsoft Office es una suite ofimática que abarca el mercado completo en Internet e interrelaciona aplicaciones de escritorio, servidores y servicios para los sistemas operativos Microsoft Windows, Mac OS X, iOS y Android. La primera versión de Office contenía los programas Word, Excel y PowerPoint. [↑](#footnote-ref-35)
36. Powershell es una consola de sistema, un terminal o «CLI» bastante más avanzado y completo que MS-DOS o CMD desde el que podremos configurar completamente un equipo informático basado en Windows sin tener que depender de un escritorio para ello. [↑](#footnote-ref-36)
37. Base64 es un sistema de codificación de información binaria a texto. Cada dígito Base64 representa exactamente 6 bits de datos. Por lo tanto, tres bytes de 8 bits (es decir, un total de 24 bits) se pueden representar con cuatro dígitos Base64 de 6 bits. [↑](#footnote-ref-37)
38. El SSID (Service Set Identifier) es una secuencia de 0-32 octetos incluida en todos los paquetes de una red inalámbrica para identificarlos como parte de esa red. El código consiste en un máximo de 32 caracteres, normalmente alfanuméricos. Todos los dispositivos inalámbricos que intentan comunicarse entre sí deben compartir el mismo SSID. Es lo que un usuario común llamaría como “el nombre de la red inalámbrica”. [↑](#footnote-ref-38)
39. APT significa “Advance Persistent Threat”, o en español: Amenaza Avanzada Persistente. El término se volvió famoso debido a una nota del New York Times en donde se detallaba la campaña de un ataque de un mes, en donde una unidad militar de China, conocida ahora como “APT 1”, logró penetrar las redes de organización de los medios de comunicación con una serie de correos electrónicos de phishing y un diluvio de muestras de malware personalizadas. Desde ese momento APT se atribuye a un ciberataque muy preciso y de gran intensidad o a un grupo que perpetra un conjunto de ciberataques y que están normalmente financiados o apoyados por estados u organizaciones. [↑](#footnote-ref-39)
40. Un “snapshot” es una copia instantánea de volumen. Vendría a ser el símil a realizar una fotografía del estado de un sistema en un momento determinado, para que en caso de necesidad deshacer todos los cambios realizados posteriormente y volver a este punto. [↑](#footnote-ref-40)
41. Una biblioteca de enlace dinámico o más comúnmente DLL (en inglés dynamic-link library) es el término con el que se refiere a los archivos con código ejecutable que se cargan bajo demanda de un programa por parte del sistema operativo. [↑](#footnote-ref-41)
42. Las direcciones de memoria pueden variar entre diferentes cargas del binario y en diferentes equipos. Esto es debido al sistema de protección ASLR (Address Space Layout Randomization) que se encarga de cargar los programas y librerías en direcciones de memoria aleatorias para dificultar el uso de funciones del sistema por parte de programas maliciosos. [↑](#footnote-ref-42)
43. Hook, proveniente del inglés, significa **Gancho**. En programación, el término *hooking* abarca una gama de técnicas utilizadas para alterar o aumentar el comportamiento de un sistema operativo, de aplicaciones o de otros componentes de software interceptando llamadas de función o mensajes o eventos pasados entre componentes de software. El código que maneja tales llamadas de función, eventos o mensajes interceptados se llama un *hook*. [↑](#footnote-ref-43)