Ransomware que se propaga a través de Telegram

line 1: Christian Pérez
line 2: *Facultad de Ingeniería de Sistemas, EPN.*
line 3: *Escuela Politécnica Nacional*
line 4: Quito, Ecuador
line 5: christian.perez01@epn.edu.ec

line 1: Jostin Vega
line 2: *Facultad de Ingeniería de Sistemas*
line 3: *Escuela Politécnica Nacional*
line 4: Quito, Ecuador
line 5: jostin.vega@epn.edu.ec

line 1: Michael Perugachi
line 2: *Facultad de Ingeniería de Sistemas*
line 3: *Escuela Politécnica Nacional*
line 4: Quito, Ecuador
line 5: michael.perugachi@epn.edu.ec

line 1: Dylan Ordoñez
line 2: Facultad de ingeniería en Sistemas
line 3: Escuela Politécnica Nacional
line 4: Quito, Ecuador
line 5: dylan.ordonez@epn.edu.ec

Resumen*— Se llevó a cabo el diseño de un software malicioso de tipo Ransomware el cual se esparce mediante la aplicación de mensajería Telegram, tanto la metodología para encriptar los datos y parte de la interfaz gráfica de este diseño fue inspirada en dos de los ransomware más populares, tales como Petya y Wanna Cry, el método que se llevó a cabo para esparcir este software malicioso entre las posibles víctimas fue disfrazarlo como un archivo zip el cual va a contener el instalador de office 2022 para que el usuario lo ejecute con toda confianza. El lenguaje de programación que se utilizó es Python por la facilidad de su sintaxis y el gran número de librerías dedicadas a encriptación de datos y creación de keys para llevar la desencriptación a cabo, estas mismas fueron de gran ayuda al momento de desarrollar el ransomware con una estructura más sencilla.

Keywords*—encriptación, víctima, librerías, keys, datos, rescate, ataque, cifrado, acceso, malware, troyano, ransomware.

I. INTRODUCCIÓN

El presente proyecto tiene como finalidad el diseño y desarrollo de un software malicioso de secuestro de datos o ransomware, el cual se va a esparcir a los dispositivos de las víctimas mediante la aplicación de mensajería Telegram, haciéndose pasar por un ejecutable para "instalar" Microsoft Office de manera gratuita, en otras palabras el método a aplicar para engañar a la víctima va a ser escondiendo el ransomware como un troyano el cual se va a encontrar en un canal de Telegram.

El método de ataque va a ser de tipo cifrado y este se va a ejecutar cuando la victima ejecute el supuesto instalador de un programa, en ese momento el ransomware va a cifrar todos los archivos que incluyan una extensión que le hayamos especificado y en la ruta que se haya definido para realizar este ataque, para finalmente dejar unas instrucciones con los pasos para que la víctima pueda recuperar el acceso a sus datos, si se solicita y se cancela el servicio de resacate se va a enviar una key con la cual la víctima va a poder obtener acceso nuevamente a sus archivos.

Este proyecto se realizó completamente en Python 3 por la facilidad de acceso a todas las librerías que permiten el cifrado y descifrado de archivos. Este software malicioso se realizó únicamente con fines académicos.

II. MARCO TEÓRICO

El ransomware es un tipo de software malicioso que cifra los archivos personales de la computadora de algún usuario o empresa, cuando el proceso de encriptación de archivos finaliza, el atacante exige un rescate a la víctima para restaurar el acceso a todos sus datos luego del pago, cabe recalcar que en la mayoría de ataques de este tipo se impone por parte del desarrollador un tiempo límite para realizar el pago, si no se realiza el dicho pago en ese tiempo el malware va a amenazar al usuario con borrar todos los datos de su computador.

Los ataques de ransomware generalmente se llevan a cabo utilizando un troyano que se disfraza como un archivo legítimo que engaña al usuario para que lo descargue o

ejecute. Una vez activado, el ransomware encriptará los archivos de la víctima y mostrará un mensaje exigiendo el pago para restaurar el acceso. El pago generalmente se exige en forma de criptomoneda como Bitcoin, porque permite transacciones anónimas y no rastreables.

Los ataques de ransomware pueden tener graves consecuencias tanto para las personas como para las organizaciones. Pueden resultar en pérdida de productividad, pérdida de ingresos y costos financieros significativos debido a la demanda de rescate y el costo de remediación.

Tipos de Ransomware

- **Ransomware de cifrado:** este tipo de ransomware cifra los archivos de la víctima y exige un rescate para descifrarlos.

- **Locker ransomware:** este tipo de ransomware bloquea a la víctima de su dispositivo y exige un rescate para restaurar el acceso.

- **Scareware**: este tipo de ransomware muestra advertencias y alertas falsas para asustar a la víctima para que pague una multa o compre software innecesario.

- **Ransomware móvil:** este tipo de ransomware se dirige a dispositivos móviles, como teléfonos y tabletas en el cual se bloquea el acceso al terminal para solicitar un pago el cual nos va a permitir usarlo de nuevo, este tipo de Ransomware sigue el mismo esquema que el ransomware para computador el cual es un secuestro de datos.

- **Malware de cryptojacking:** este tipo de malware utiliza el dispositivo de la víctima para extraer criptomonedas sin su conocimiento, mayormente se desarrolla para computadoras, pero existen unos pocos que se dedican a extraer estas criptomonedas mediante dispositivos móviles .

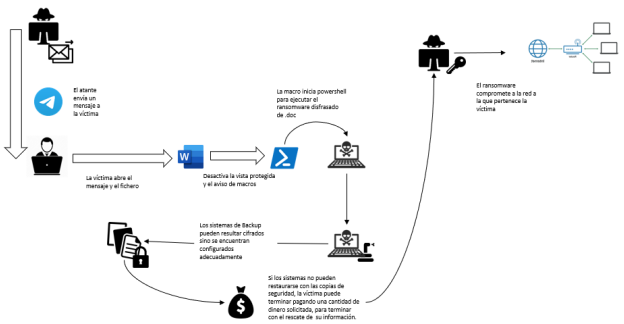
- **Ransomware-as-a-service (RaaS):** este es un modelo de negocio en el que el ransomware se ofrece como un servicio a otros ciberdelincuentes.

Funcionamiento

Un ransomware intentará tomar el control de nuestro equipo mediante distintas técnicas, una de estas es el phishing, la cual es conocida como la “pesca del distraído”, pues por lo general mediante vía email, un atacante fingirá ser alguien que no es, haciendo uso de un diseño de correo electrónico convincente aprovechando tu distracción, te dará alguna excusa para que descargues el archivo que lo adjuntan en el correo de esta manera para que lo ejecutes y es ahí donde comienza lo peor para la persona que sufre este tipo de ataque, pues la encriptación de archivos es algo que nuestro sistema operativo trae predeterminado.

El atacante utilizará esa herramienta para empezar a cifrar los archivos del equipo, y debido a esto, el sistema operativo generará una clave privada, ya que, esta clave es utilizada tanto para cifrar y descifrar los archivos, no obstante, ahí es donde se presenta el problema, debido a que dicha clave se guarda localmente, entra en juego el malware que se descargó, pues este liberará su clave pública (generada con anterioridad) para encriptar la clave privada, como siguiente paso el malware enviará una segunda clave pública al servidor del atacante, la cual se encargará de desencriptar la clave privada y es de esa manera, donde una vez que la persona afectada pague la cantidad de dinero solicita, se enviará la segunda clave alojada en el servidor (del atacante), para desencriptar la clave privada y permitir el descifrado de los archivos, dando por terminado el ataque criptográfico.

Arquitectura de un ransomware



Componentes

Los componentes de un ataque típico de ransomware suelen incluir:

- **Cifrado:** el malware cifra los archivos en la computadora infectada utilizando un algoritmo de cifrado fuerte.

- **Nota de rescate:** el malware deja una nota de rescate, que puede tener la forma de una ventana emergente o un archivo de texto, que explica la situación y exige el pago a cambio de la clave de descifrado.

- **Mecanismo de pago:** los atacantes suelen exigir el pago en forma de criptomoneda, como Bitcoin, para mantener el anonimato y dificultar que las autoridades rastreen la transacción.

- **Servidor de comando y control:** el ransomware se comunica con su servidor C&C para enviar los datos de la víctima, recibir la clave de cifrado o los comandos.

- **Ingeniería social:** muchos ataques de ransomware utilizan técnicas de ingeniería social, como correos electrónicos de phishing, para engañar a las víctimas para que abran un archivo adjunto o hagan clic en un enlace que envía el malware a su computadora.

Sobre Telegram.

Telegram es una aplicación de mensajería instantánea basada en la nube que se centra en la velocidad y la seguridad. Permite a los usuarios enviar mensajes, fotos y videos entre sí, así como crear grupos para comunicarse con hasta 200,000 miembros. También ofrece una función llamada "Chat secreto" que proporciona cifrado de extremo a extremo para mayor seguridad. Telegram está disponible en una amplia gama de plataformas, incluidas iOS, Android y escritorio.

Es posible que el malware se propague a través de Telegram, al igual que con cualquier otra plataforma de mensajería, los principales métodos de infección pueden ser:

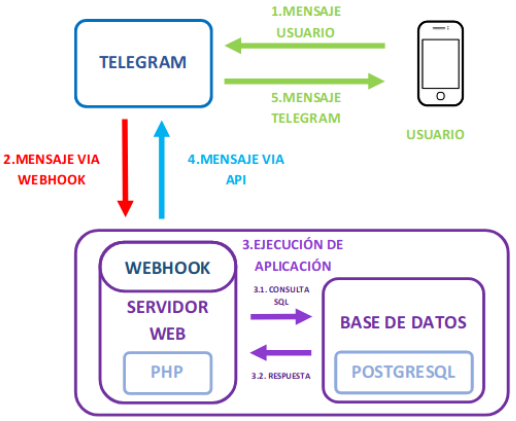
- Enlaces sospechosos los cuales son recibidos de fuentes desconocidas, el malware a menudo se puede propagar a través de enlaces que conducen a sitios web maliciosos.
- Archivos ejecutables o comprimidos de fuentes desconocidas son la principal causa de infección en la mayoría de computadores del mundo , ya que en el mayor de los casos estas pueden contener malware.

Telegram es conocido por su enfoque en la seguridad y la privacidad. Estas son algunas de las funciones de seguridad que ofrece Telegram:

- Cifrado servidor-cliente: los mensajes en Telegram se cifran en su camino desde el remitente hasta el servidor y desde el servidor hasta el destinatario.
- Chats secretos: Telegram ofrece una función llamada "Chat secreto" que proporciona cifrado de extremo a extremo para mayor seguridad. Esto significa que los mensajes se cifran en el dispositivo del remitente y solo se pueden descifrar en el dispositivo del destinatario, sin que nadie en el medio (incluido Telegram) pueda leer los mensajes.
- Mensajes autodestructivos: Telegram permite configurar un temporizador para que los mensajes se eliminen después de que haya pasado una cierta cantidad de tiempo. Esto puede ser útil para la información confidencial que no desea conservar para siempre.
- Almacenamiento en la nube: Telegram almacena todos los mensajes y medios en la nube, lo que significa que puede acceder a ellos desde cualquier dispositivo y no tiene que preocuparse por almacenar respaldos en un medio externo.

Es en este último punto en el cual nos vamos a enfocar con mayor énfasis, porque va a ser el medio por el cual vamos a esparcir el ransomware a las posibles víctimas.

Arquitectura de Telegram



Componentes

Telegram es una aplicación de mensajería que consta de varios componentes, que incluyen:

- **Aplicación del lado del cliente:** esta es la aplicación que los usuarios instalan en sus dispositivos (como teléfonos inteligentes o tabletas) para acceder a las funciones de Telegram.

- **Infraestructura del lado del servidor:** Telegram utiliza una red distribuida de servidores para almacenar y transmitir mensajes, gestionar la autenticación de usuarios y realizar otras tareas.

- **Protocolo MTProto:** Este es el protocolo que utiliza Telegram para cifrar y transmitir mensajes entre el cliente y el servidor.

- **Telegram Bot API**: Esta es la API que permite a los desarrolladores crear y administrar bots que pueden interactuar con los usuarios de Telegram.

- **Telegram Open Network (TON):** un sistema de pago basado en blockchain, que se planeó para ser utilizado para micropagos y compras dentro de la aplicación, pero el proyecto se suspendió.

- **Chats seguros y chats secretos:** Telegram ofrece cifrado de extremo a extremo para mensajería segura y privada, utilizando el protocolo MTProto y Signal Protocol.

III. METODOLOGÍA DE EXPERIMENTACIÓN

Durante el desarrollo del ransomware se comenzó probando cuál era su efectividad al momento de cifrar cierto tipo de archivos, por lo mismo se empezó a probar el cifrado en archivos de tipo texto cuya extensión sea ".txt", por seguridad de la máquina en la cual se trabaja este proyecto se implementó la prueba de esta primera ejecución en una máquina virtual la cual está configurada con el sistema operativo Windows 10 x64 como se puede observar en la Figura 3.1.

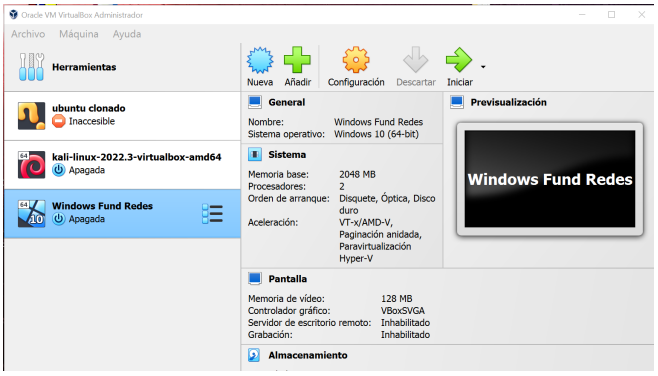


Figura 3.1. Configuración de máquina virtual en Oracle VM Box la cual ejecuta Windows 10.

Para observar de manera más sencilla el comportamiento que presentaba el ransomware al ser ejecutado, se acondicionó la ruta en la que iba a actuar al ser ejecutado como se observa en la Figura 3.2.

```
# Usamos sysroot para crear todos los path parea archivos, y para poder encriptar el
self.sysRoot = os.path.expanduser('~')
# Definimos la ruta en la que queremos que cifre los archivos
self.localRoot = r'C:\Users\Proy Redes\Downloads\Python-Ransomware-master\localRoot'
```

Figura 3.2. Modificación de ruta en la cual el ransomware va a cifrar los datos.

Para dicha primera prueba se creó una carpeta la cual contenía tres activos de texto los cuales deben ser cifrados si el ransomware estaba ejecutándose de manera adecuada, al ejecutarse los archivos si lograron ser cifrados con éxito como se observa en la Figura 3.3.

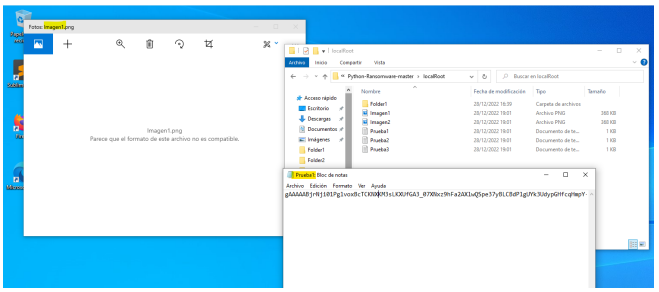


Figura 3.3. Archivo de texto cuyo contenido fue cifrado por el ransomware desarrollado en este proyecto.

Luego de realizado este procedimiento se crea va a crear un archivo de texto con las instrucciones para recuperar el acceso a los datos cifrados, el cual se va a localizar en el escritorio y se ejecutará automáticamente cuando el Ransomware finalice el proceso de secuestro de datos, la ruta exacta del escritorio se va a obtener mediante la función self.sysRoot que permite obtener el nombre de nuestro equipo como se observa en la Figura 3.4.

```
    f.write(enc_fernent_key)
#Escribe una key fernet en el escritorio con las indicaciones para recuperar el acceso
with open(f'{self.sysRoot}/Desktop/EMAIL_ME.txt', 'wb') as fa:
    fa.write(enc_fernent_key)
# asigna una contraseña de encriptacion
self.key = enc_fernent_key
#remueve la clave de encriptación
self.crypter = None
```

Figura 3.4. Creación del archivo que va a contener las indicaciones y la key para recuperar el acceso.

Cuando ya se envió este archivo con la clave de encriptación y pagando el respectivo rescate, se procedió a utilizar el método para desencriptar el archivo desde la terminal base. Se desarrolló un método en python el cual lee el archivo y lo identifica porque está está nombrado de una manera específica, ejecutamos dicho método y nos va a devolver un archivo que reenviamos a la víctima para que se pueda validar la clave y se desencripten sus archivos.

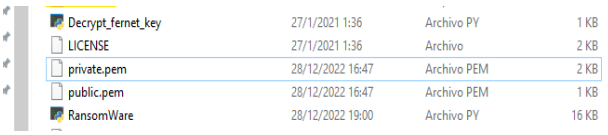


Figura 3.5. Carpeta que contiene el archivo .py que se va a encargar de desencriptar la clave de rescate.

Luego de reenviar la clave mencionada se coloco el archivo en el escritorio de nuestra máquina y el ransomware válido

si se trataba de la misma key que este género para cifrar los archivos, como se realizó la simulación del pago del rescate si se envio la clave de rescate y los datos volvieron a ser accesibles para nuestra máquina vulnerada.

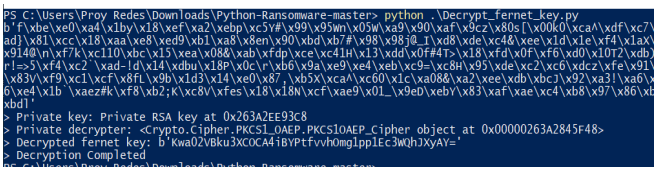


Figura 3.6. Proceso de desencripción exitoso luego de enviar el archivo de texto con la key de rescate de la Figura 3,5.

Finalmente pudimos observar que el esquema en el cual se planteó y desarrolló el ransomware era correcto porque se verificó el contenido de los archivos anteriormente encriptados y estos coincidieron en su totalidad con los datos recuperados luego de pagar por el rescate y colocar la key.

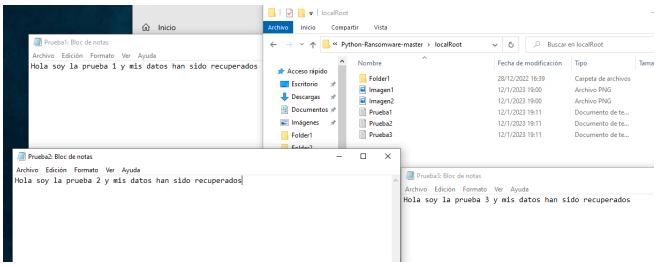


Figura 3.7. Proceso de verificación de contenido de los archivos de texto después de recuperar la información cifrada.

IV. RESULTADOS

Como notamos en el desarrollo de la práctica, tenemos entendido que el ransomware partirá desde su despliegue usando distintas estrategias que puedan comprometer a un sitio web, aprovechando que esto no es seguro, logrando vulnerar. Siendo de esta manera que los atacantes toman en control de dicha web la cual cumplió y aprobó todos los filtros establecidos por el navegador, donde crearán otro sitio web para que este suplante la identidad de este sitio. Es ahí donde comienza un ataque de estos, pues una vez que se propague información de dicho sitio para que usuarios que no prestan atención a los temas de seguridad, se vean afectados.

En este caso, el ransomware se propagó por Telegram, que quiere decir esto, bueno no hemos suplantado la identidad de Telegram, sino que hemos aprovechado la velocidad de envió y su seguridad, entonces ¿como hemos suplantado la identidad para poder pasar desapercibido el ransomware? Bueno esto es algo sencillo, pues como se mencionó con anterioridad, los atacantes usan distintas estrategias para comprometer sitios web, sin embargo, esto no solo va dirigido a los sitios web, pues el atacante puede suplantar la identidad de otra extensión de un archivo, por ejemplo en este caso no enviaremos el ejecutable tal y como se diseñó, sino que hallaremos la forma de cambiar si tipo de archivo, puede ser “.doc” o alguna otra extensión que pueda pasar desapercibida por el usuario que será la víctima.

Entonces, cómo se logró esto, se pudo llegar a la instalación y propagación del ransomware, pues una vez que la víctima caiga en la trampa, el ransomware se ejecutará en segundo plano, realizando los siguiente:

```python
# Genera una [SYMMETRIC KEY] en la maquina de la victima en la cual se va a encriptar los archivos
def generate_key(self):
    # Generatesuna url segura(basada en base64)
    self.key =  Fernet.generate_key()
    # Createa un objeto fernet de encripción y desencripcion
    self.crypter = Fernet(self.key)


#escribe la clave simetrica en un archivo de texto
def write_key(self):
    with open('fernet_key.txt', 'wb') as f:
        f.write(self.key)


# Encriptación simetrica que va a ser creada en la maquina de la victima

def encrypt_fernet_key(self):
    with open('fernet_key.txt', 'rb') as fk:
        fernet_key = fk.read()
    with open('fernet_key.txt', 'wb') as f:
        # key RSA publica
        self.public_key = RSA.import_key(open('public.pem').read())
        # creamos el objeto publico para encriptar
        public_crypter =  PKCS1_OAEP.new(self.public_key)
        # Encripta una clave fern
        enc_fernet_key = public_crypter.encrypt(fernet_key)
        # Escribe la clave encriptada en un archivo
        f.write(enc_fernet_key)
    #Escribe una key fernet en el escritorio con las indicaciones para recuperar el acceso
    with open(f'{self.sysRoot}/Desktop/EMAIL_ME.txt', 'wb') as fa:
        fa.write(enc_fernet_key)
    # asigna una contraseña de encriptacion
    self.key = enc_fernet_key
    #remueve la clave de encriptación
    self.crypter = None
```

Figura 4.1. Generación de la Key.

Como podemos observar, en este apartado nos centramos en generar la key que servirá para cifrar los archivos, a la vez que esta se guarda en un archivo para que después que se haga el rescate, esta key sea liberada y descifra los archivos que se vieron afectados por el ransomware, sin embargo, como sabe este malware a donde atacar, es decir, ¿como sabe que dirección seguir para dar con la información relevante o de suma importancia para el usuario?

Bueno, claramente no lo sabe, el atacante que desplegó este malware, no sabe qué tipo de información tiene la víctima, sin embargo, tiene una ventaja, pues el ransomware no se va a centrar a buscar esa información, pues como sabemos, este malware está programado por un humano y es ahí donde entra en juego, la dirección a la que queremos atacar y se hablará a continuación.

```python
'''
# Usamos sysroot para crear todos los path parea archivos, y para poder encriptar el sistema
self.sysRoot = os.path.expanduser('~')
# Definimos la ruta en la que queremos que cifre los archivos
self.localRoot = r'C:\Users\Proy Redes\Downloads\Python-Ransomware-master\localRoot' # Debugging/Testing
```

Figura 4.2. Selección de la ruta a cifrar.

Como lo mencionamos anteriormente, el ransomware no sabe a qué dirección atacar a menos que se la establezca y es aquí donde entra en juego lo programación que se llevó a cabo, ya que, comenzamos llamando a una función que se encargue de crear los paths necesarios para los archivos y poder encriptar el sistema.
Y a la vez que definimos la ruta a la cual el ransomware va a atacar o donde queremos que se cifren los archivos.

Una vez que se logró eso, que es lo que sigue, eso es sencillo, pues debemos hacerle saber a la víctima que su dispositivo se encuentra bajo ataque, y una excelente manera es hacerlo es desplegando de un archivo txt, para que la víctima lo note y lo tenga presente.

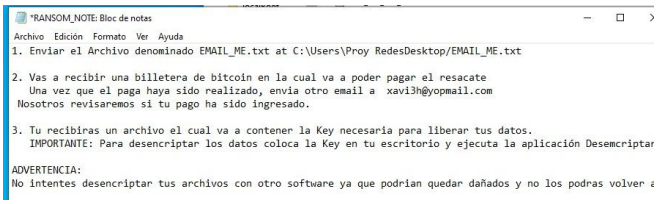


Figura 4.3. Despliegue del txt.

Como observamos, es aquí donde comienza la extorsión, pues una vez que el ransomware ha cifrado los archivos que se encuentran presente en la ruta que se había especificado, este solicitará el método de pago para que se pueda realizar el descifrado de los archivos, entonces ¿qué hace el txt desplegado? Bueno, no hace más que informar a la víctima que su dispositivo está bajo ataque y que debe enviar una cantidad de dinero para que el descifrado se lleve a cabo.

Entonces, una vez que se haya realizado el pago, el atacante solicitará que envíe un correo y mensaje para que de esa manera poder hacerle llegar la key y logre liberar los datos y dar por terminado el ataque.

Empaquetamiento

Ahora bien, anteriormente mencionamos que el método que usa el ransomware para poder infiltrarse en el ordenador de cualquier persona, es suplantando la identidad de un sitio web que cumplió con requisitos impuestos por el navegador o en el caso de ser compartido como archivo mediante Telegram, podemos realizarlo cambiando la extensión del archivo, no obstante, podemos saltarnos el paso de suplantar la identidad de un archivo, haciendo que la extensión ".exe" que se obtiene al convertir el código que elaboramos, no sea reconocida por lo motores de búsqueda de virus.

Para poder llevar a cabo esto, tenemos una manera de lograrlo y es adicionando la extensión de "archivos por lote", pues estos tipos de archivos contienen comandos que ordenan al sistema que realice actividades específicas, ya que, forman parte del sistema operativo. Es de dicha manera que haremos que el archivo ejecutable que se obtiene de convertir nuestro código a un ".exe" lleve adicionalmente la extensión ".bat" para lograr que el ransomware no sea detectado.

Entonces, ¿cómo convertimos nuestro código a un archivo ejecutable con .bat? para poder realizar esto, debemos identificar la ubicación en la que se encuentra este nuestro fichero de python ".py" para nuestro caso deberemos identificar la ubicación del archivo llamado "RansomWare.py"

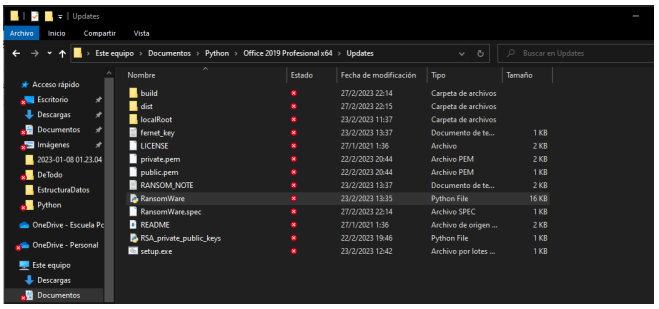


Figura 4.4. Ubicación del fichero .py.

Ahora, crearemos un nuevo archivo de tipo texto ".txt", en el cual vamos a escribir lo siguiente:

Figura 4.5. Creación del ejecutable .bat.

Como podemos apreciar en la figura 4.5. Haciendo uso de comillas doble llamamos al ejecutable de python, el cual se encuentra en dicha dirección. Seguido de esa dirección, vamos a colocar la ruta en la que se encuentra el fichero .py que vamos a convertir a ejecutable. Finalizamos guardando el archivo con la exención .bat y tendremos el siguiente archivo:

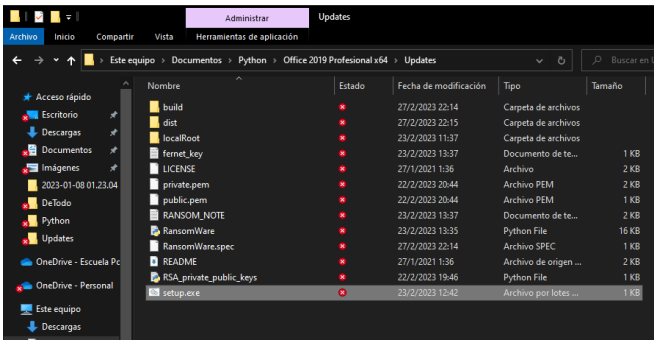


Figura 4.6. Ejecutable finalizado.

Una vez que tenemos nuestro ejecutable, vamos a ingresar a la página llamada “VirusTotal”, en la cual vamos a cargar el archivo ejecutable que se obtuvo anteriormente para poder saber si los motores de búsqueda de virus que se encuentran ahí reconocen a nuestro archivo como algún tipo de malware.

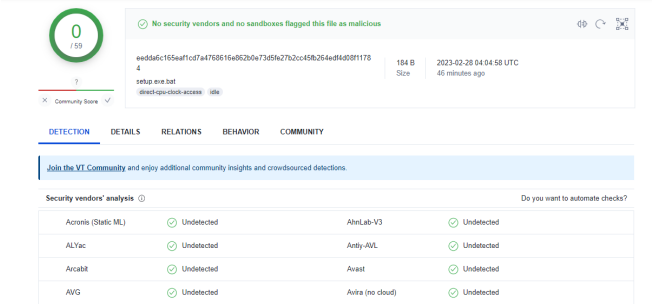


Figura 4.7. Uso de VirusTotal con “.bat”.

Como podemos notar en la figura 4.7. VirusTotal no identifica a nuestro ransomware como algún malware, por lo cual obtuvimos éxito en el empaquetamiento.

Si bien, logramos que los motores de búsqueda de virus presentes en VirusTotal, no identificaran al ransomware como algún tipo de malware, veamos qué hubiera pasado si en lugar de agregar la extensión “.bat”, solo convertimos nuestro código de python en un ejecutable “.exe”. Para ello, vamos a dirigirnos a donde se encuentra el fichero .py que vamos a pasar a ejecutable:

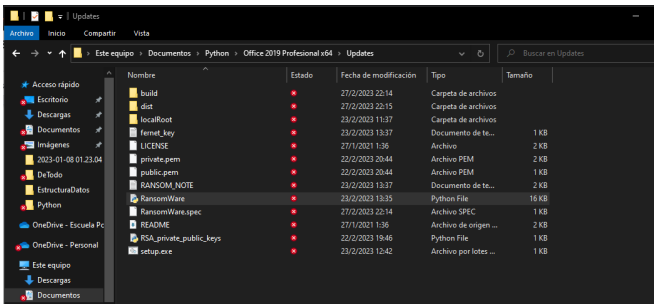


Figura 4.8. Ubicación del fichero .py.

Ahora, vamos a identificar la ruta en la que se encuentra el fichero:

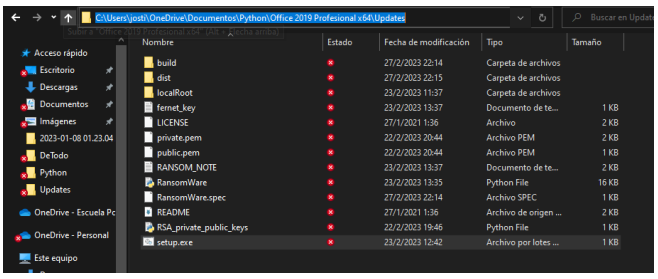


Figura 4.9. Ruta del fichero.

Seguido de esto, vamos a abrir el símbolo del sistema o cmd, para poder empezar con la instalación de un paquete que necesitaremos para poder convertir en un ejecutable a nuestro fichero .py.

Figura 4.10. Abriendo el cmd.

Una vez que se abre el cmd, vamos a escribir el siguiente comando “pip install pyinstaller”

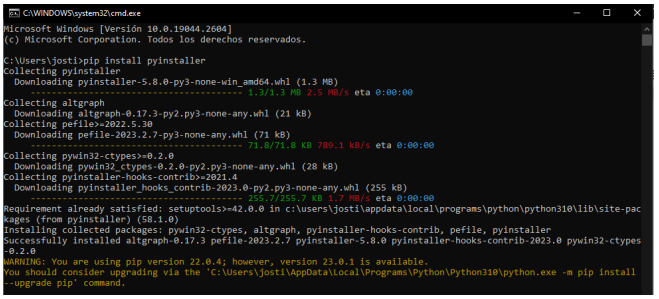


Figura 4.11. Instalación del paquete a utilizar

Dado que se terminó con la instalación de dicho paquete, vamos a escribir el siguiente comando: “pyinstaller --onefile -w RansomWare.py”, donde estamos salicitando que convierta al fichero “RansomWare.py” en un ejecutable “.exe”.

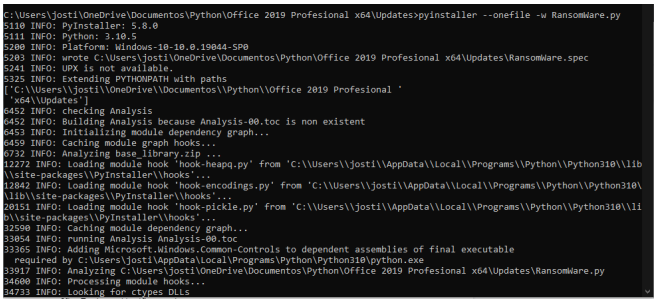


Figura 4.12. Convirtiendo el fichero .py al ejecutable.

Finalmente, se crearán dos carpetas, donde la que nos interesa se llama “dist”, pues es ahí donde se guardó el ejecutable “.exe”.

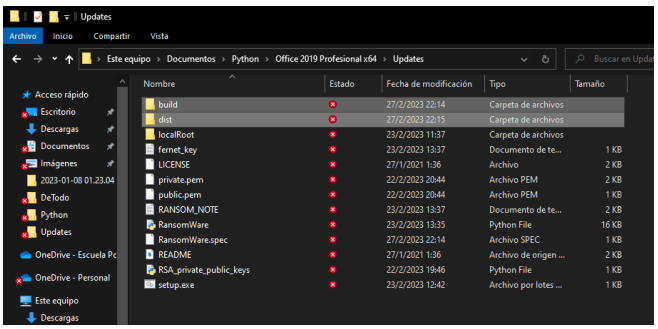


Figura 4.13. Carpetas creadas durante la creación del ejecutable.

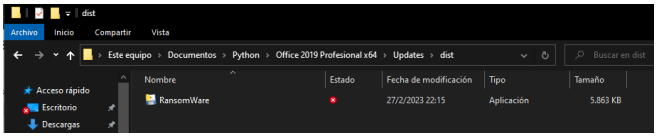


Figura 4.14. Ejecutable ubicado dentro de la carpeta dist.

Ahora, si analizamos este ejecutable ".exe" en VirusTotal, obtenemos lo siguiente:

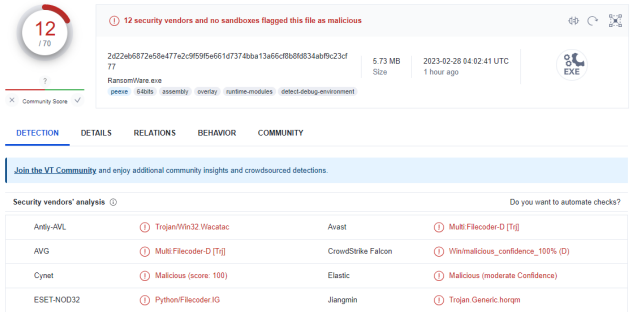


Figura 4.15. Uso de VirusTotal con el ".exe".

Como logramos apreciar, los motores de búsqueda de virus de esta página si logran detectar al ejecutable como un malware, razón por la cual, para poder realizar el empaquetado y este logre que el ransomware llegue a ejecutarse en la máquina de la víctima, se optó por el primer método, haciendo que se empaqueta el ejecutable con la extensión ".bat". En consecuencia, logra que el ransomware se ejecute dentro del ordenador de la víctima.

V. CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- Mediante el desarrollo del presente proyecto se puede concluir que es muy sencillo ocultar un archivo malicioso, en este caso un ransomware, en un aparente programa de uso cotidiano,de tal manera que el usuario piense que está instalando un programa de licencia de manera gratuita, pero en realidad va a ejecutar el ransomware.
- Con los resultados obtenidos realizando una simulación de víctima que descarga el programa gratuito de Telegram, se pone a discusión el peligro al que estamos expuestos al descargar cualquier tipo de software de sitios web que no sean oficiales o no cuenten con algún grupo calificado para realizar software libre de malware.
- A raíz de este proyecto podemos concluir que el objetivo principal, el cual era crear un software malicioso de tipo ransomware y que este sea capaz de propagarse por medio de la aplicación de mensajería Telegram se ha logrado, ya que gracias a la buena elección del un lenguaje de programación con gran facilidad de sintaxis y amplitud de librerías la encriptación y descifrado de información se logró de manera menos compleja y exitosa.
- El proceso de encriptación y desencriptación se basa en una estructura muy compleja, pero gracias a las librerías implementadas para Python 3 sobre estos procesos, el desarrollo del código se vuelve sumamente sencillo comparado con realizar el mismo proyecto en otros lenguajes de programación que no tienen la opción de importar estas librerías.

Recomendaciones

- Tener en cuenta que estos tipos de software no son para nada éticos, por lo cual solo debería realizarse para propósitos didácticos.
- Realizar las primeras pruebas de encriptación en una máquina virtual para evitar daños colaterales en el equipo, también se recomienda empezar cifrando archivos de texto para confirmar que funciona de manera correcta.
- Utilizar un lenguaje de programación que permite importar librerías que faciliten el proceso de encriptación y desencriptación con la finalidad de hacer el código más corto, sencillo de entender y más óptimo en términos de coste computacional.
- Ejecutar las primeras pruebas del ransomware en una ubicación controlada para ver el comportamiento del mismo y evitar problemas si es que el proceso de descripción no es totalmente funcional.

VI. ANEXOS

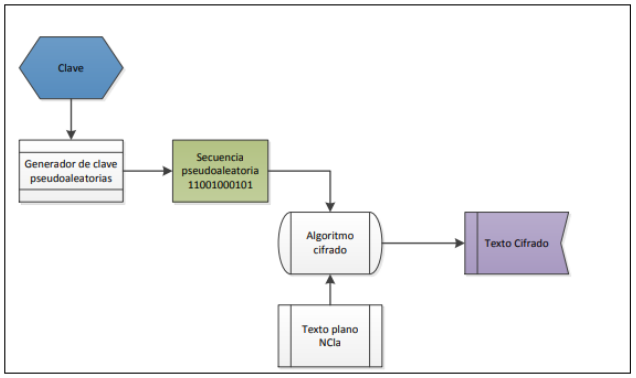


Figura 4.2. Cifrado de datos de flujo para texto plano (Moya, ECB Cifrado, 2015).

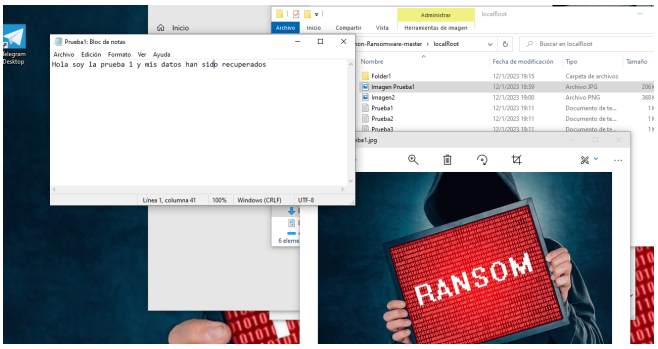


Figura 4.2. Datos de tipo texto e imagen con sus contenidos antes de realizar el ataque ransomware.

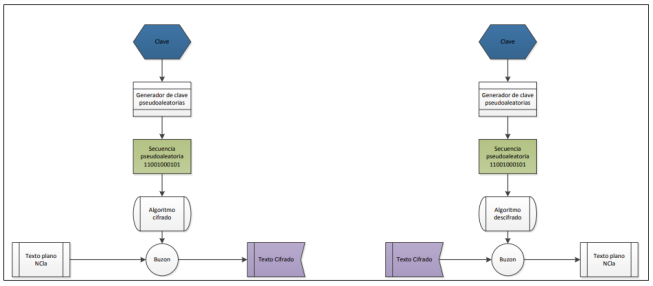


Figura 4.3. Cifrado Síncrono utilizado en el proyecto (Moya, ECB Cifrado, 2015)

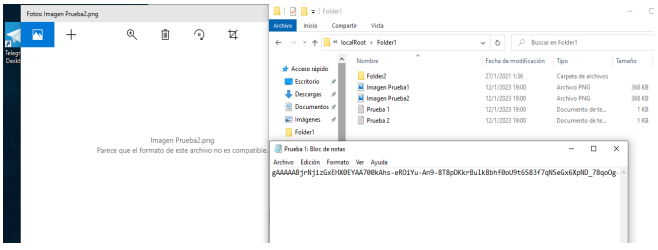


Figura 4.4. Datos de tipo texto e imagen con sus contenidos después de realizar el ataque ransomware.

VII. REFERENCIAS

[1] Moya, J & Escobar, F. (2015). Desarrollo de una aplicación para encriptar información en la transmisión de datos en un aplicativo de mensajería web. First Edition; Repositorio Pontificia Universidad Católica del Ecuador . [Online]. Disponible en: http://repositorio.puce.edu.ec/bitstream/handle/22000/8335/Disertacion_MoyaCazaJohannaBeatriz_EscobarErazoFranklinAndres.pdf

[2] Jimenez, J (2022). Qué tipos de ransomware existen y en qué me afecta. [Online]. Disponible en: https://www.redeszone.net/tutoriales/seguridad/tipos-ransomware/

[3] López, M. (2021). 9 formas en que los atacantes utilizan los sitios web comprometidos. [Online]. Disponible en: https://www.welivesecurity.com/la-es/2021/02/23/formas-atacantes-utilizan-sitios-web-comprometidos/

[4] IBM. ¿Qué es el Ransomware? [Online]. Disponible en: https://www.ibm.com/cl-es/topics/ransomware

[5] López, P. (2020). ¿Qué es Telegram y para qué sirve? [Online]. Disponible en: https://www.geeknetic.es/Telegram/que-es-y-para-que-sirve