

|  |
| --- |
| **Universidad Internacional de La Rioja (UNIR)**  **Maestría en Seguridad Informática** |
|  |
| Análisis criptográfico de archivos de texto plano encriptados por ransomware haciendo uso de redes neuronales |

**Trabajo de Investigación**

**Presentado por:** Mariscal Muñoz, Cristina

**Director:** Ibarra Reyes, Jaime

Ciudad: Guadalajara, Jalisco, México.

Fecha: 30 de julio de 2021

**Resumen**

**Palabras Clave**:

Queda pendiente redactar el resumen y el abstract

**Maestría en Seguridad Informática**

****

**Tabla de Contenido**

1 Introducción 1

1.1 Antecedentes 1

1.2 Objetivo 2

1.2.1 Objetivo General 2

1.2.2 Objetivo Específico 2

1.3 Justificación 2

1.4 Metodología 4

1.4.1 Revisión de bibliografía y fuentes de información 4

1.4.2 Diseño experimental 5

1.4.3 Configuración experimental 6

1.4.4 Desarrollo del experimento 7

1.4.5 Evaluación de la propuesta y sus los resultados 7

1.4.6 Desarrollo de conclusiones 8

1.5 Organización del Documento 8

2 Marco Teórico 9

2.1 Ransomware 9

2.1.1 Ransomware de cifrado simétrico 10

2.1.2 Ransomware de cifrado asimétrico del cliente 11

2.1.3 Ransomware de cifrado asimétrico del servidor 11

2.1.4 Ransomware de cifrado asimétrico del servidor y cliente + cifrado simétrico 12

2.1.5 Análisis de cripto ransomware 12

2.2 Redes neuronales 13

2.2.1 Backpropagation 15

2.2.2 Herramientas para aprendizaje profundo 15

2.3 Encriptación y cifrado 15

2.3.1 Análisis criptográfico 15

2.4 Estado del arte 16

3 Desarrollo del experimento 17

3.1 Propuesta 17

3.2 Configuración experimental 17

3.2.1 Conjunto de datos para la experimentación 17

3.2.2 Configuración de sistemas virtualizados 18

3.2.3 Binarios de ransomware 18

3.2.4 Herramientas de análisis de datos 19

3.3 Análisis de malware 19

3.3.1 Infección de ransomware 19

3.3.2 Análisis dinámico 19

3.3.3 Recolección de archivos 19

3.4 Análisis criptográfico 19

3.4.1 Evaluación manual 19

3.4.2 Limpieza y preprocesamiento de datos 20

3.4.3 Programación y entrenamiento del modelo de análisis (red neuronal) 20

4 Discusión 24

4.1 Análisis de resultados 24

4.2 Reproducibilidad 24

4.3 Retos y obstáculos 24

5 Conclusiones 25

5.1 Hallazgos 25

5.2 Líneas de investigación futuras 25

Referencias 26

Apéndice A - Glosario de Términos 28

**Tabla de Figuras**

[Figura 1 - Etapas del desarrollo del piloto experimental 7](#_Toc78392184)

Infinitos agradecimientos:

A mi familia, a quienes que se los debo todo.

A José Pizaña, el mejor gerente y mentor que me apoyó en el curso del postgrado.

A mi marido, por ser la persona que siempre ha creído en mí.

En honor a todos los seres queridos que no pudimos abrazar.

# Introducción

## Antecedentes

Es innegable la relevancia que hoy en día tienen los activos de información para una empresa. Según lo define P. Johnson [1], estos pueden ser definidos como aquellos datos o cualquier otro conocimiento que tenga valor para una organización. Estos activos suelen verse comprometidos ante amenazas de seguridad como lo son los ataques humanos, riesgos ambientales, o incluso software malicioso diseñado con fines específicos. Un ejemplo de este último es el ransomware, el cual es definido por R. Joven [2] como un tipo de malware que por medio de encriptación secuestra los activos del sistema vulnerado con la finalidad de pedir un rescate (del inglés “ransom”) para recuperar el acceso o restauración a estos mismos.

Actualmente los métodos de recuperación de activos informáticos ante una infección de ransomware son escasos, y la mayoría están enfocados en estrategias preventivas [3], lo cual resulta inconveniente para organizaciones y sistemas que no implementen las medidas requeridas para la defensa ante este tipo de ataques. Igualmente, en su publicación de NIST (por sus siglas en inglés National Institute of Standards and Technology) 1800-11 [4] describe el gran reto que representa el proceso de desencriptación de los archivos afectados con los mecanismos actuales, no siendo efectivos para la recuperación de los activos de forma íntegra en la mayoría de los casos.

Según NIST [3] los métodos modernos para la prevención de infecciones por ransomware están enfocados en la instalación de antivirus (aún siendo bien sabido que ningúno de ellos es 100% viable ante estos ataques), mantener actualizados los parches de seguridad de los sistemas en todas sus capas y la implementación de políticas de seguridad para los usuarios de los sistemas. Sin embargo, existe investigación como la propuesta por Xinyi Hu y Yaqun Zhao [5] sobre la restauración de texto sin formato de AES haciendo uso de redes neuronales, lo cual podría representar una buena base para el proceso de descripción y recuperación de archivos vulnerados por ransomware, el cual es el objetivo del presente piloto experimental.

Por otro lado, trabajos como el expuesto por M. Wecksten et al [6] sobre un método para la recuperación de una infección de ransomware por encriptación, o incluso la publicación de S. Mehnaz [7] sobre un sistema de detección de ransomware en tiempo real, permiten contextualizar este marco teórico a una aplicación práctica para la experimentación en el tratamiento de los sistemas infectados.

## Objetivo

### Objetivo General

Analizar criptográficamente archivos de texto plano afectados por ransomware haciendo uso de redes neuronales para la recuperación de los datos cifrados.

### Objetivo Específico

* Identificar los trabajos existentes sobre el uso de redes neuronales en la desencriptación de archivos cifrados por algoritmos simétricos.
* Determinar la viabilidad del uso de redes neuronales para la desencriptación y recuperación de archivos afectados por ransomware.

## Justificación

Durante la última década [8] se ha observado una mayor incidencia y desarrollo en lo referente a infección de sistemas corporativos por medio de ransomware. Durante el mes de mayo del 2021 se registraron diversos ataques de alto impacto a industrias como la de distribución de carnes en Estados Unidos de América, o incluso la red de tuberías de combustibles en USA, cobrando rescates de hasta $4.4 millones de dólares americanos [9].

Igualmente, según el artículo de The Washington Post [10], este tipo de ataques no solo tienen impacto en empresas de gran escala, actualmente ha afectado a personas regulares al causar, por ejemplo, el cierre de escuelas o el retraso de quimioterapias.

Dada esta situación, desde hace al menos 6 años [2] los ciberdelincuentes han adoptado un modelo de “negocio” que es nombrado como Ransomware as a Service, promoviendo este tipo de malware como un servicio y volviendo a esta modalidad de ciber extorción sumamente lucrativa. Es por ello que la investigación técnica sobre mecanismos de control tanto preventivos como correctivos cobra importancia tanto en un contexto técnico, como económico, reputacional, operativo, social y político.

La presente investigación pretende ser una referencia para el desarrollo de soluciones correctivas en sistemas infectados con ransomware que ejecuten el cifrado de los documentos por medio de algoritmos de llave de encriptación simétrica. Las fuentes de estudio base para este experimento es expuesto por la investigación de Xinyi Hu y Yaqun Zhao [5] sobre la restauración de texto sin formato de AES basada en una red neuronal, en conjunto con la investigación de Arslan Ashraf et al [11] para el análisis de ransomware mediante ingeniería de funciones y redes neuronales profundas.

De lograr encontrar un modelo o un proceso efectivo para la recuperación o restauración de archivos en un sistema infectado sin tener que realizar el pago supuesto por la extorción, no solo representaría una solución técnica para este tipo de ataques por malware, sino también una gran contramedida para la oposición y desmantelación de este modelo de negocio que los ciberdelincuentes han explotado.

Sin embargo, uno de los principales retos que enfrenta esta investigación es la delimitación del tipo o versiones de ransomware que se va a analizar, ya que el modelo propuesto haciendo uso de redes neuronales y backpropagation podría no ser efectivo [12] para los ataques que hagan uso de algoritmos de encriptación asimétricos o bien con múltiples llaves de salteo. Dadas estas limitaciones, al hacer un análisis del malware más reciente, se realizará un análisis dinámico de la ejecución del ransomware en el ambiente de pruebas de seguridad.

De esta forma, aún si fuera rechazada la hipótesis inicial sobre el enfoque correctivo para la restauración de los archivos encriptados, se espera llegar a conclusiones relevantes respectivas al trabajo existente relacionado con el criptoanálisis en este tipo de escenarios, aún a pesar de los inconvenientes mencionados.

## Metodología

Para el desarrollo del presente piloto experimental se aplicó una metodología de investigación cuantitativa, generando datos los datos de análisis por medio de la experimentación [1] y con un enfoque de desarrollo en cascada e iterativo [2]. Esto con el fin de desarrollar el experimento incrementalmente pero de forma consistente con los requerimientos y expectativas definidas inicialmente. La metodología que se siguió cuenta con las etapas explicadas a continuación.

### Revisión de bibliografía y fuentes de información

Para desarrollar el presente piloto experimental, así como su marco teórico, se requirió recopilar la información más reciente sobre aplicaciones e implementaciones de estrategias de aprendizaje automático, como lo son las redes neuronales y “backpropagation” en ciberseguridad, más específicamente en el criptoanálisis y para el diseño de contramedidas conta ransomware.

Las principales referencias y fuentes utilizadas para realizar esta investigación se obtuvieron del portal IEEE Xplore, ediciones Elsevier a través del portal Science Direct, arxive.org y Google Scholar, con el fin de identificar las investigaciones existentes relacionadas con el tema de este artículo.

Los términos de búsqueda seleccionados para este análisis literario se enlistan a continuación de forma textual en el idioma inglés para incremental la cantidad de resultados relevantes: “ransomware”, “malware analysis”, “cryptanalysis”, “AES”, “encryption algorithms”, “deep learning”, “neural networks” y “backpropagation”. Estos se combinaron con el operador "y", para obtener resultados más precisos de acuerdo con el tema principal de la investigación. Al examinar los recursos recolectados, los principales criterios de análisis y evaluación para seleccionar fuentes fueron:

* La validez de la fuente de información
* Las metodologías utilizadas para las investigaciones y experimentos.
* La reproducibilidad de los resultados del experimento.
* El nivel de impacto tanto del artículo como de la revista
* Evaluar si el tema estaba en consonancia con el propósito del presente experimento.
* Se prefirió utilizar la fuente primaria de investigación. Por lo tanto, el uso de la investigación de fuentes secundarias se limitó a experimentos o implementaciones concretas.
* Las credenciales (impacto, otros artículos, credibilidad) de los autores fuente.

Una vez recabada la bibliografía requerida, esta fue analizada detalladamente para la definición y redacción del marco teórico en conjunto con los conceptos clave respectivos al área de estudio la presente investigación mencionadas con anterioridad. Igualmente se identificaron los trabajos existentes relacionados y que sirvieran de soporte para el experimento.

### Diseño experimental

Para el desarrollo del diseño experimental fue requerido acotar sintetizar el problema con el fin de definir más precisamente la forma en el que este mismo sería desarrollado. La siguiente pregunta de investigación e hipótesis fueron planteadas:

*¿Es posible restaurar (desencriptar) los activos de un sistema afectado por ransomware haciendo uso de redes neuronales?*

*Usando como datos de entrenamiento archivos de contenido conocido que serán posteriormente encriptados por diversos ransomwares, se busca entrenar una red neuronal para la restauración de los activos existentes de un sistema.*

Dada la existencia de la gran cantidad de tipos de ransomware, se determinó un único binario para la ejecución del experimento, ya que el realizar este estudio haciendo uso de múltiples ejemplos de este malware aumentaría la complejidad en el proceso de análisis criptográfico y de entrenamiento del modelo de aprendizaje.

Igualmente, como parte del diseño experimental, fue necesario identificar los recursos necesarios para su desarrollo correcto. Esta evaluación abarcó desde las necesidades de cómputo (hardware en conjunto con sus requerimientos mínimos), los recursos digitales (binarios, ejecutables, software de virtualización y para el diseño del modelo de aprendizaje automático entre otros), hasta los recursos necesarios de forma más indirecta para el análisis y presentación de los resultados y conclusiones (herramientas, software, datos adicionales, entre otros).

Como parte primordial del diseño experimental, se definieron las etapas del experimento (expuestas en la figura 1) que sería necesario ejecutar para el desarrollo exitoso del proyecto. Se propuso la ejecución iterativa del experimento con el fin de realizar correcciones en los pasos ejecutados, así como para reevaluar los parámetros experimentales con el fin de obtener mejores resultados. Del mismo modo se identificaron los criterios de aceptación que nos ayudaron a comprender los resultados y reconocer si se confirmó o no la hipótesis de la propuesta inicial.

### Configuración experimental

En esta etapa se realizó una investigación sobre las especificaciones de los recursos necesarios para la ejecución del experimento, como son las herramientas de análisis de datos, el sistema de virtualización y las imágenes de los sistemas operativos sobre los que se realizó el análisis, la adquisición de los archivos binarios del ransomware, se generaron los datos de prueba y entrenamiento, entre otros. Una vez definidos dichos detalles, se realizó la recolección, preparación / configuración y, de ser el caso, instalación necesarios previos a la puesta en marcha del experimento.

### Desarrollo del experimento

En la siguiente figura 1 se muestran las principales fases para el desarrollo del experimento, incluyendo el apartado anterior referente a la configuración experimental, mencionando de forma breve algunas de las actividades más destacables en cada sección. Se muestra por igual el que, de ser necesario, se reiteraron sobre las fases de análisis de malware, análisis criptográfico y evaluación de resultados incorporando los reajustes de acuerdo a los resultados obtenidos en una iteración previa. Para la presente investigación, se realizaron dos iteraciones y se presentan únicamente los resultados óptimos de entre las mismas.

Diagram, text

Description automatically generated

Figura 1 - Etapas del desarrollo del piloto experimental

### Evaluación de la propuesta y sus los resultados

Una vez desarrollado el experimento fue necesario llevar a cabo la evaluación de los resultados obtenidos con respecto a la propuesta o hipótesis inicial. Para ello se realizó un análisis de la precisión del modelo entrenado con respecto al conjunto de datos encriptados generado y extraído de los ambientes de pruebas con el fin de identificar si estos mismos fueron óptimos en la restauración o desencriptación de los archivos originales. En esta etapa se evaluó si la hipótesis inicial fue rechazada y hasta qué punto la pregunta de investigación pudo ser respondida.

### Desarrollo de conclusiones

En esta sección se realizó una síntesis del trabajo de investigación en retrospectiva, incorporando los detalles conclusivos durante su diseño, ejecución y evaluación de resultados. Igualmente, se identificaron las líneas de investigación futuras que pueden ser desarrolladas a partir de los puntos expuestos en presente piloto experimental.

1- https://libguides.macalester.edu/c.php?g=527786/&p=3608643

2- https://prepinsta.com/software-engineering/iterative-waterfall-model/

## Organización del Documento

El presente documento está organizado en 5 capítulos, los cuales desglosan desde la introducción (capítulo 1) los objetivos y la justificación del diseño experimental desarrollado, seguido por el marco teórico (capítulo 2) para la contextualización de la investigación con respecto al conocimiento existente, y el desarrollo del experimento (capítulo 3) en donde se explican los pasos realizados para la obtención de resultados así como el análisis de los mismos y del proceso ejecutado (capítulo 4), hasta las conclusiones (capítulo 5) donde se exponen los hallazgos y el trabajo a futuro en esta línea de investigación.

# Marco Teórico

En este capítulo se exponen los resultados de la investigación realizada relativa a las áreas de estudio más relevantes para el presente piloto experimental, con el fin que estos sirvan como las bases para la comprensión apropiada y la ejecución de este mismo. La investigación está enfocada en los conceptos clave referentes al tipo de malware ransomware, orientado a sus variantes de encriptación y, por ende, se desarrollan temas de criptoanálisis y algunos algoritmos tanto simétricos como asimétricos.

Igualmente, dado que el experimento está orientado a la restauración de los archivos encriptados por medio de un modelo de aprendizaje automático, se detallan los fundamentos conceptuales necesarios en los tópicos de aprendizaje profundo, redes neuronales y backpropagation.

## Ransomware

Para llegar a una definición del término ransomware, se han tomado diversas fuentes de referencia. La universidad de Berkeley [1] menciona que “es un tipo de software malicioso que infecta una computadora y restringe el acceso de los usuarios hasta que se paga un rescate para desbloquearla”. Este mismo va a restringir, ya sea un servicio de forma en que no sea accesible o usable, o los archivos del usuario en sí mismo al encriptarlos volviéndolos ilegibles. Los rescates que suelen exigirse por medio de divisas virtuales como lo es el Bitcoin.

Otras referencias, como el reporte emitido por Microsoft [2], refieren que este tipo de ataques se han vuelto más sofisticados ya que, en su caso, durante el proceso de infección, el ransomware puede intentar extenderse por su red a computadoras conectadas, servidores, y cualquier otro nodo de red accesibles. Otro comportamiento observado es que este malware puede haber infectado un sistema sin presentar los cambios típicos (de denegación de accesos a archivos o servicios), existiendo en la computadora en un estado de inactividad o gestación donde los cibercriminales extraen datos o activos de información valiosos para la posterior extorsión.

Abonando a lo explicado previamente, A. Gazet [3] expone que en la mayoría de los casos es posible distinguir tres fases de alto nivel que seguirá el ransomware, las cuales pueden ser secuenciales o estar agrupadas para su ejecución conjunta:

* Buscar objetivo: lo cual debe entenderse como los archivos que van a ser afectados, esto dado que el cifrar el disco completo requiere de mucho tiempo y poder de cómputo, y puede resultar una labor improductiva con respecto al objetivo final de este tipo de ataques. Es por ello que dentro de su programación los ransomwares suelen definir una lista de formatos de archivo específicos que serán los objetivos principales, por ejemplo, todos aquellos con extensiones como: rtf, doc, odt, zip, etc.
* Extorsión: la cual puede basarse en el restringir el acceso a la información de la víctima, o incluso hacer uso de los datos valiosos que fueron extraídos del sistema afectado a modo de chantaje.
* Solicitud de rescate: ya que el objetivo final es ganar dinero, de acuerdo con el modelo de ransomware, se presentará la notificación con las instrucciones tanto para la realización de pago, como para el rescate y restauración de los archivos posterior al mismo.

Por su parte, adentrándonos en los detalles del funcionamiento del ransomware, los artículos de V. Cracium et al [4] y E. Kolodenker et al [5] expone las generalidades técnicas de este tipo de malware bajo una perspectiva de criptografía, exaltando los problemas que cada uno de los siguientes enfoques podría llegar a tener y cómo esto ha causado que se implementen técnicas mixtas o más complejas en el proceso de cifrado de los archivos objetivo:

### Ransomware de cifrado simétrico

Ya que este tipo de algoritmos, como es el caso de la familia AES, tiene un tiempo de ejecución baja es utilizado para la encriptación física de todos los archivos de usuario en disco, almacenando de forma local las llaves usadas en este proceso. Esto con la finalidad de que al momento en que la víctima realiza pago del rescate, otro módulo interno del ransomware abrirá este archivo con las claves y comenzará a descifrar los archivos.

El principal problema de este método es que el archivo que almacena las llaves de cifrado sigue siendo accesible, permitiendo a un especialista hacer uso de estas para generar un script o una herramienta similar al módulo de desencriptación que permita restaurar los archivos sin necesidad de pagar por el rescate de los archivos.

### Ransomware de cifrado asimétrico del cliente

Por otra parte, nos encontramos con este enfoque que genera un par de llaves de cifrado asimétrico (usadas por algoritmos como RSA), de forma que todos los archivos serán cifrados con la llave pública, y se almacenará la llave privada al servidor. Una desventaja de este método es el hecho de que este tipo de algoritmos de cifrado son bastante lentos, consumiendo mucho tiempo en su ejecución y, de esta forma, aumentando la posibilidad de ser detectado o detenido de forma oportuna durante su operación.

Otra gran desventaja de este método es que el ransomware necesita enviar la llave privada a un servidor, volviendo un requerimiento el que la computadora infectada debe esté conectada a internet y el servidor deberá estar conectado. Esta no es una buena opción ya que, de fallar una de estas dos precondiciones, la llave privada deberá ya sea ser eliminada o, al igual que en el caso anterior, almacenada temporalmente en el disco.

### Ransomware de cifrado asimétrico del servidor

Similar a lo mencionado previamente, en este esquema el servidor generará un par de llaves. Sin embargo, en este caso, la llave pública que se utilizará para encriptar los archivos estará embebida dentro del código fuente del ransomware. El problema con este método es que, una vez se pague el rescate, un especialista puede ser capaz de extraer la llave privada al momento de que esta sea enviada desde el servidor a la computadora cliente infectada, de forma que esta llave pueda ser usada para restaurar el resto de los sistemas afectados dentro de la red sin tener que pagar rescates adicionales.

### Ransomware de cifrado asimétrico del servidor y cliente + cifrado simétrico

Este es el esquema más usado por la mayoría de ransomware modernos, el cual es un modelo híbrido, ya que usa tanto el cifrado simétrico como el asimétrico, y no necesita conexión a Internet en el proceso de encriptación, solo en el de desencriptación. En este método se generan dos pares de llaves, uno de ellos desde el servidor, y otro desde el cliente.

El proceso de infección consistirá en los siguientes pasos: para el cifrado de los archivos de la víctima se hace uso de un algoritmo simétrico, se guardan las claves simétricas en un archivo el cual posteriormente será encriptado por medio de las llaves generadas dinámicamente por el ransomware en la computadora cliente. De igual forma, este mismo contendrá la llave pública del servidor dentro de su código fuente que será utilizada para encriptar la llave privada del cliente sin necesidad de contar con una conexión a internet ni de transmitir información inicialmente con el servidor.

### Análisis de cripto ransomware

Existen numerosas investigaciones con respecto a la detección y análisis del ransomware, un ejemplo destacable es el marco de trabajo propuesto por I. Kara et al [6], donde se propone el siguiente método para llevar a cabo un análisis técnico y, de ser necesario, forense de este tipo de software malicioso:

1. Se realiza una imagen del equipo atacado por el cripto ransomware, o se configura un ambiente de experimentación: esto con la finalidad de que todas las pruebas y procesos de análisis sean realizados en un entorno seguro previniendo posibles daños al sistema en vivo u otros sistemas de producción.
2. Investigaciones de la imagen: para ello, se identificarán los archivos binarios del ransomware en ejecución, recopilando información como los metadatos, sus dependencias, si este ha realizado cambios en registros del sistema o inyectado código en otras aplicaciones, etc. Si el ransomware se encuentra en ejecución es necesario realiza un análisis de comportamiento, como son las modificaciones de la matriz de archivos, arquitectura de código, operaciones en memoria o en ejecución, entre otros.
3. Después de completar el análisis, es necesario informar los procedimientos y toda la información recabada por medio de un reporte técnico forense, o una memoria o artículo de investigación para su uso contra posibles ataques similares.

1- https://security.berkeley.edu/faq/ransomware/

2- Microsoft report shows increasing sophistication of cyber threats - Microsoft On the Issues

3- https://vxug.fakedoma.in/archive/other/VxHeavenPdfs/Comparative%20analysis%20of%20various%20ransomware%20virii.pdf

4- 598.pdf (iacr.org)

5- https://dl.acm.org/doi/pdf/10.1145/3052973.3053035

6- https://ieeexplore.ieee.org/document/9298128

## Redes neuronales

Para estudiar de forma comprehensiva los conceptos clave en el área de redes neuronales, es necesario llegar a una definición concreta del concepto. Por un lado, S. Kukreja1 et al [1] la definen de manera simple como un intento de imitación del cerebro humano para llegar a la capacidad de aprender nuevas cosas, adaptarse a los cambios y al ambiente.

Estas redes neuronales son conformadas por unidades de procesamiento llamadas neuronales, en las cuales se tratará de replicar la estructura y comportamiento de una neurona natural, componiéndose así por dendritas (entradas), y axones (salidas por medio de sinapsis). La arquitectura de una red neuronal es conformada por los siguientes componentes:

* Capa de entrada: la cual recibe los valores.
* Capa o capas ocultas: cada una de las cuales pueden estar compuestas por una o varias neuronas
* Capa de salida: la cual tiene una sola neurona y será la encargada de integrar los valores de la última capa oculta en los valores de salida finales.

Un concepto importante en el estudio de las redes neuronales es el de función de activación, la cual es el modelo matemático utilizado para procesar los datos de entrada y obtener los de salida. Algunas de las funciones de activación las más populares o usadas son: de paso, sigmoide, lineal, hiperbólicas tangenciales, de rampa, entre otras.

Los pesos son las unidades que permiten realizar un control de señales y pueden entenderse como la habilidad de procesamiento y la fortaleza de la conexión entre dos neuronas. Dicho en otras palabras, los pesos van a determinar qué tanta influencia tendrán ciertas entradas en las salidas.

Por otro lado, el sesgo de una red neuronal son valores constantes que serán enviados como entradas adicionales en la siguiente capa de procesamiento. A diferencia de los pesos, estos no están influenciados por las capas previas.

Ahora, en términos prácticos, una red neuronal estará basada en modelos de aprendizaje supervisado y no supervisado. Esto permite resolver problemas tanto de clasificación y agrupación, como de predicción de acuerdo al tipo de conjunto de datos usados en el entrenamiento, así como el modelado y configuración de la red neuronal.

A una red neuronal se le da primero un conjunto de datos de entrada conocidos y se le pide que obtenga una salida conocida. A esto se le llama entrenar la red. La red pasa por muchas épocas de este tipo hasta que el error (diferencia entre la salida real y la salida deseada) está dentro de una cierta tolerancia). Ahora se dice que la red está entrenada. Este proceso de entrenamiento establece los pesos entre todas las neuronas en todas las capas. Los pesos así obtenidos de una red entrenada se utilizan para calcular la respuesta de la red a un dato desconocido.

Igualmente, es importante definir en primera instancia el concepto de aprendizaje profundo (por su traducción del inglés “Deep learning”). Según el artículo de IBM [1], el aprendizaje profundo es una subsección del área de estudio del aprendizaje automático (o aprendizaje de máquina, según su traducción del inglés “machine learning”), esto se puede entender esencialmente como una red neuronal con al menos tres capas adicionales.

1- https://www.researchgate.net/profile/Kuldeep-Shiruru/publication/319903816\_AN\_INTRODUCTION\_TO\_ARTIFICIAL\_NEURAL\_NETWORK/links/59c0fe55458515af305c471a/AN-INTRODUCTION-TO-ARTIFICIAL-NEURAL-NETWORK.pdf

2- https://www.ibm.com/cloud/learn/deep-learning

### Backpropagation

Queda pendiente completar este tema del marco teórico

### Herramientas para aprendizaje profundo

Queda pendiente completar este tema del marco teórico

## Encriptación y cifrado

La universidad de Cambridge define la encriptación como “el proceso de cambiar información o señales electrónicas en un código secreto (sistema de letras, números o símbolos) que las personas no pueden entender o usar sin un equipo especial” [1]. Mientras que Karpesky [2] provee una definición más a fin a la seguridad informática donde menciona que el cifrado es un control de seguridad básico que ayuda a garantizar que los datos no puedan ser robados ni leídos por usuarios no autorizados o malintencionados.

Generalmente los métodos de cifrado se clasifican en dos tipos: algoritmos simétricos y asimétricos, donde el primer hace uso de la misma llave tanto para encriptar como para desencriptar los datos cifrados, haciéndolo muy rápido en sus tiempos de ejecución, sin embargo, tiene siempre el inconveniente relativo a la forma en que la clave es intercambiada [3].

Por otro lado, el cifrado asimétrico es más seguro ya que utiliza dos llaves, una para encriptar y otra para desencriptar, sin embargo esto implica peores tiempos y rendimiento en la ejecución del algoritmo debido a los cálculos complejos de este proceso. Existen igualmente, y como se ha mencionado previamente en el documento, implementaciones híbridas donde se utiliza un algoritmo simétrico para cifrar la información y uno asimétrico para el intercambio de llaves de cifrado [3].

1- ENCRYPTION | meaning in the Cambridge English Dictionary

2- What is Data Encryption? | Kaspersky

3- AES IP for hybrid cryptosystem RSA-AES | IEEE Conference Publication | IEEE Xplore

### Análisis criptográfico

Queda pendiente completar este tema del marco teórico

## Estado del arte

Queda pendiente completar este tema del marco teórico

# Desarrollo del experimento

## Propuesta

La propuesta para presente experimento consistió en el análisis del ransomware académico de código abierto [1], el cual hace uso del algoritmo simétrico AES-255-CRT para encriptar los archivos del sistema, y el asimétrico RSA-4096 para asegurar el intercambio con el servidor. La funcionalidad de esta versión de malware está basada en el ransomware Cryptolocker.

El desarrollo del modelo de aprendizaje automático con redes neuronales fue programado haciendo uso de tecnologías diferentes: con la librería Tensorflow para el lenguaje de programación Python, y con Matlab. Esto con la finalidad de realizar una verificación doble del modelo diseñado, descartando así el error respectivo a la tecnología utilizada.

Para su ejecución es necesario realizar la instalación del servidor en una máquina virtual el cual almacena la clave de identificación de la víctima junto con la clave de cifrado utilizada por el malware. Posteriormente se realizó la instalación del malware en 2 sistemas operativos diferentes para su análisis y la generación del conjunto de datos de entrenamiento. Finalmente se llevó a cabo el criptoanálisis por medio del modelo de redes neuronales diseñado para la desencriptación de texto plano.

1. https://github.com/abhir98/ransomware

## Configuración experimental

### Conjunto de datos para la experimentación

Fue necesario generar un acumulado de aproximadamente 750 000 archivos de texto plano que contienen entre 64 y 2048 caracteres aleatorios soportados por el formato de codificación de caracteres UTF-8. Estos archivos fueron generados por medio del siguiente script.

Adjuntar script

### Configuración de sistemas virtualizados

Para la configuración del laboratorio de pruebas de malware se siguieron los pasos propuestos por la universidad de Hawaii [1], haciendo uso de la herramienta de virtualización VirtualBox, y las imágenes de los sistemas operativos Windows 10 y Debian 10. Las máquinas virtuales fueron configuradas con los siguientes recursos:

* Servidor Debian 10: CPU . RAM . HDD .
* Cliente Debian 10: CPU . RAM . HDD .
* Cliente Windows 10: CPU . RAM . HDD .

Igualmente, se instalaron las siguientes herramientas de ante mano en las máquinas virtuales cliente previo a la infección de las mismas con el ransomware. Para el monitoreo del sistema de archivos y almacenamiento: Process Monitor y ProcDOT. Para el monitoreo de procesos: Process Explorer y Process Hacker.

Añadir screenshot?

1. https://westoahu.hawaii.edu/cyber/vulnerability-research/vulnerability-tutorials/building-a-vulnerability-malware-test-lab/

### Binarios de ransomware

Los binarios y código fuente del ransomware fueron adquiridos del repositorio de código abierto: <https://github.com/abhir98/ransomware>. Se siguieron los pasos documentados dentro del repositorio para compilar tanto el binario del malware por si mismo, así como la imagen contenerizada (en Docker) para la ejecución del servidor.

Añadir screenshot / comandos ejecutados

[Get Docker | Docker Documentation](https://docs.docker.com/get-docker/)

### Herramientas de análisis de datos

Se realizó la instalación de las siguientes librerías y software de análisis de datos para la ejecución del experimento, siguiendo la documentación oficial de cada herramienta.

Añadir screenshots / comandos

<https://la.mathworks.com/products/deep-learning.html>

<https://www.tensorflow.org/api_docs/python/tf?hl=es-419>

## Análisis de malware

### Infección de ransomware

Pendiente

### Análisis dinámico

Pendiente

### Recolección de archivos

Pendiente

## Análisis criptográfico

### Evaluación manual

Agregar tests y scripts de encripción para el cripto análisis manual de AES

### Limpieza y preprocesamiento de datos

Procedimiento en el que los datos de los archivos de texto plano fueron extraídos y procesados en el archivo csv (tstnew1\_output.csv) separado por comas en conjunto con el texto no encriptado, de forma que la primera columna es el texto original (“TEXT”), y la segunda es el texto cifrado (“ENCRYPTED”).

### Programación y entrenamiento del modelo de análisis (red neuronal)

Agregar el código fuente de Matlab

Agregar capturas de pantalla de las métricas y resultados de ambos entrenamientos (con tensorflow y con Matlab)

Agregar descripción de por qué se utlizaron los siguientes métodos:

* Tokenizer
* Sequence
* Embedding
* LSTM
* SparseCategoricalCrossentropy
* sparse\_categorical\_accuracy

import pandas as pd

import numpy as np

from tensorflow.keras.preprocessing.text import Tokenizer

import tensorflow as tf

import ast

import os

import json

import matplotlib.pyplot as plt

from nltk import tokenize

import seaborn as sns

import binascii

dataset\_pd = pd.read\_csv("C:\\Users\\crist\\Documents\\AES tests\\tstnew1\_output.csv")

dataset\_pd["ENCRYPTED"] = dataset\_pd["ENCRYPTED"].apply(lambda t: [i for i in binascii.unhexlify(t)])

def create\_character\_tokenizer(list\_of\_strings):

tokenizer = Tokenizer(filters=None,

char\_level=True,

split=None,

lower=False)

tokenizer.fit\_on\_texts(list\_of\_strings)

return tokenizer

tokenizer = create\_character\_tokenizer(dataset\_pd["TEXT"])

tokenizer\_config = tokenizer.get\_config()

word\_counts = json.loads(tokenizer\_config['word\_counts'])

index\_word = json.loads(tokenizer\_config['index\_word'])

word\_index = json.loads(tokenizer\_config['word\_index'])

def strings\_to\_sequences(tokenizer, list\_of\_strings):

sentence\_seq = tokenizer.texts\_to\_sequences(list\_of\_strings)

return sentence\_seq

seq\_texts = strings\_to\_sequences(tokenizer, dataset\_pd["TEXT"])

dataset\_pd["TEXT"] = seq\_texts

x = dataset\_pd["TEXT"]

x = [np.asarray(i) for i in x]

y = dataset\_pd["ENCRYPTED"]

y = [np.asarray(i) for i in y]

test\_pct = .2

batch\_size = 64

buffer\_size = 10000

embedding\_dim = 256

epochs = 50

seq\_length = 200

rnn\_units = 1024

x\_test = x[:int(len(x)\*test\_pct)]

y\_test = y[:int(len(y)\*test\_pct)]

x\_train = x[int(len(x)\*test\_pct):]

y\_train = y[int(len(y)\*test\_pct):]

tst\_full = tf.data.Dataset.from\_tensor\_slices((x, y))

dataset\_train = tf.data.Dataset.from\_tensor\_slices((x\_train,y\_train))

train\_data = dataset\_train.batch(batch\_size, drop\_remainder=True)

train\_data = tf.data.Dataset.from\_tensor\_slices((x\_train,y\_train))

valid\_data = tf.data.Dataset.from\_tensor\_slices((x\_test,y\_test))

model = tf.keras.Sequential()

model.add(tf.keras.layers.Embedding(input\_dim=vocab\_size, output\_dim = 256, mask\_zero=True, batch\_input\_shape=(batch\_size, None)))

model.add(tf.keras.layers.LSTM(units=512, return\_sequences=True,stateful=True))

model.add(tf.keras.layers.Dropout(0.2))

model.add(tf.keras.layers.LSTM(units=512, return\_sequences=True,stateful=True))

model.add(tf.keras.layers.Dense(256))

model.add(tf.keras.layers.Dense(256))

model.add(tf.keras.layers.Dense(256))

model.add(tf.keras.layers.Dense(256))

checkpoint\_callback=tf.keras.callbacks.ModelCheckpoint(filepath='.\\models\\ckpt', save\_weights\_only=True, save\_best\_only=True)

model.compile(optimizer='adam', loss=tf.keras.losses.SparseCategoricalCrossentropy(from\_logits=True), metrics=['sparse\_categorical\_accuracy'])

history = model.fit(tst\_full, epochs=30, callbacks=[checkpoint\_callback, tf.keras.callbacks.EarlyStopping(patience=2)])

model.compile(optimizer='sgd', loss='mse')

history = model.fit(x, y, batch\_size=32, epochs=1)

<https://www.kaggle.com/c/ciphertext-challenge-ii/data>

https://towardsdatascience.com/generating-text-with-recurrent-neural-networks-based-on-the-work-of-f-pessoa-1e804d88692d

https://www.tensorflow.org/text/tutorials/text\_generation

https://towardsdatascience.com/generating-text-with-tensorflow-2-0-6a65c7bdc568

https://towardsdatascience.com/sequence-to-sequence-models-from-rnn-to-transformers-e24097069639

# Discusión

## Análisis de resultados

Pendiente

Evaluación de parámetros experimentales

## Reproducibilidad

Pendiente

## Retos y obstáculos

Pendiente

# Conclusiones

Pendiente

## Hallazgos

Pendiente

## Líneas de investigación futuras

Pendiente

# Referencias

PENDIENTE INCORPORAR LAS REFERENCIAS EN VERDE DENTRO DEL PRESENTE DOCUMENTO Y ADAPTAR LAS SIGUIENTES EN FORMATO IEEE

1. P. Johnson. “ISO 27001 Key terms”. PJR. Recuperado el 06/06/2021 de: https://www.pjr.com/standards/iso-27001/iso-27001-key-terms
2. R. Joven. “Ransomware-as-a-Service: Rampant in the Underground Black Market”. 2017. Fortinet. Recuperado el 06/06/2021 de: https://www.fortinet.com/blog/threat-research/ransomware-as-a-service-rampant-in-the-underground-black-market?utm\_source=blog&utm\_campaign=ransomware-as-a-service-rampant-in-the-underground-black-market
3. NIST. (2021). “PREPARING YOUR ORGANIZATION FOR RANSOMWARE ATTACKS”. Recuperado el 06/06/2021 de: https://csrc.nist.gov/CSRC/media/Projects/ransomware-protection-and-response/documents/NIST\_Tips\_for\_Preparing\_for\_Ransomware\_Attacks.pdf
4. McBride T., Ekstrom M., Lusty L., Sexton J., Townsend A. (2017). “1800-11: Data Integrity Recovering from Ransomware and Other Destructive Events”. NIST. Recuperado el 06/06/2021 de: https://www.nccoe.nist.gov/sites/default/files/library/sp1800/di-nist-sp1800-11-draft.pdf
5. X. Hu, Y. Zhao. “Research on Plaintext Restoration of AES Based on Neural Network”. 2018. Hindawi. Recuperado el 06/06/2021 de: https://downloads.hindawi.com/journals/scn/2018/6868506.pdf
6. M. Weckstén, J. Frick, A. Sjöström and E. Järpe, "A novel method for recovery from Crypto Ransomware infections," 2016 2nd IEEE International Conference on Computer and Communications (ICCC), 2016, pp. 1354-1358, doi: 10.1109/CompComm.2016.7924925. Recuperado el 06/06/2021 de: https://ieeexplore.ieee.org/document/7924925
7. Mehnaz S., Mudgerikar A., Bertino E. "RWGuard: A Real-Time Detection System Against Cryptographic Ransomware". Purdue University. Recuperado el 06/06/2021 de: https://www.researchgate.net/publication/327469473\_RWGuard\_A\_Real-Time\_Detection\_System\_Against\_Cryptographic\_Ransomware\_21st\_International\_Symposium\_RAID\_2018\_Heraklion\_Crete\_Greece\_September\_10-12\_2018\_Proceedings
8. Lakhani. “Analyzing the History of Ransomware Across Industries”. 2021. Fortinet. Recuperado el 06/06/2021 de: https://www.fortinet.com/blog/industry-trends/analyzing-the-history-of-ransomware-across-industries
9. R. Iyengar, C. Duffy. “Hackers have a devastating new target”. 2021. CNN Bussines. Recuperado el 06/06/2021 de: https://edition.cnn.com/2021/06/03/tech/ransomware-cyberattack-jbs-colonial-pipeline/index.html
10. H. Kelly. “Ransomware attacks are closing schools, delaying chemotherapy and derailing everyday life”. 2021. The Washington Post. Recuperado el 06/06/2021 de: https://www.washingtonpost.com/technology/2021/07/08/ransomware-human-impact/
11. Arslan Ashraf et all. “Ransomware Analysis using Feature Engineering and Deep Neural Networks”. 2019 Recuperado el 06/06/2021 de: https://arxiv.org/ftp/arxiv/papers/1910/1910.00286.pdf
12. K. Shihab. “A Backpropagation Neural Network for Computer Network Security”. 2006. Journal of Computer Science. Recuperado el 06/06/2021 de: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.134.4053&rep=rep1&type=pdf

# Apéndice A - Glosario de Términos

Redes Neuronales

Backpropagation

Ransomware

Malware

Deep learning

Aprendizaje de máquina / machine learning

Modelado de datos

Criptografía

Criptoanálisis / análisis criptográfico

Virtualización

Encriptación

Desencriptación

Llaves de cifrado

Algoritmos simétricos

Algoritmos asimétricos

Análisis estático de malware

Modelado de datos

Entrenamiento de modelo de aprendizaje

Archivos de texto plano

AES

RSA

Bitcoin

Script

Máquina virtual

UTF-8

Queda pendiente agregar las definiciones a este glosario