

|  |
| --- |
| **Universidad Internacional de La Rioja (UNIR)**  **Maestría en Seguridad Informática** |
|  |
| Cripto análisis de archivos de texto plano encriptados por malware 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 junio de 2021

**Maestría en Seguridad Informática**

****

**Tabla de Contenido**

Introducción 1

Título del Trabajo 1

Línea de Trabajo 1

Tipo de Trabajo de Investigación 1

Antecedentes 1

Objetivo General 2

Objetivos específicos 2

Justificación 3

Plan de Trabajo 4

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.

# Resumen (PASADO, 250 palabras)

Es innegable

**Palabras clave:**

# Abstract

Es innegable

In this paper, we are going to verify the possibility to create a ransomware simulation that will use an arbitrary combination of known tactics and techniques to bypass an antimalware defense. To verify this hypothesis, we conducted an experiment in which an agent was trained with the help of reinforcement learning to run the ransomware simulator in a way that can bypass anti-ransomware solution and encrypt the target files. The novelty of the proposed method lies in applying reinforcement learning to anti-ransomware testing that may help to identify weaknesses in the anti-ransomware defense and fix them before a real attack happens

# Capítulo 1 – Introducción (PRESENTE, 10 – 14 pag)

## Línea de Trabajo

Seguridad en sistemas operativos

Criptografía y mecanismos de seguridad

## Tipo de Trabajo de Investigación

Piloto experimental

# 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 General

Analizar el comportamiento de ransomware haciendo uso de redes neuronales para la recuperación de activos de la información.

## Objetivos específicos

* 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 moderno.

# 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, sería posible tomar un enfoque preventivo o que se valga de otros mecanismos para la restauración de los archivos, como podría ser el estudio de memoria estática o dinámica.

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 a pesar de los inconvenientes mencionados, así como una propuesta preventiva en los casos de estudio desarrollados a continuación.

# Metodología (PASADO)

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, así como los casos de estudio en los que este mismo sea relevante.

**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. Las 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.

Agregar descripciones específicas??



Figura 1- Etapas del desarrollo del piloto experimental

* + Desarrollo del modelo a utilizar
  + Análisis del ransomware
  + Generación de las bases de datos para el cripto análisis
  + Entrenamiento del modelo
  + Análisis de resultados
  + Implementación en el ambiente afectado\*\*\*\*\*

**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.

# Capítulo 2 - Marco teórico (PRESENTE 12-20 pag)

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.

## 2.1 - 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](https://blogs.microsoft.com/on-the-issues/2020/09/29/microsoft-digital-defense-report-cyber-threats/)
3. <https://vxug.fakedoma.in/archive/other/VxHeavenPdfs/Comparative%20analysis%20of%20various%20ransomware%20virii.pdf>
4. [598.pdf (iacr.org)](https://eprint.iacr.org/2018/598.pdf)
5. <https://dl.acm.org/doi/pdf/10.1145/3052973.3053035>
6. <https://ieeexplore.ieee.org/document/9298128>

## 2.2 - 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>

## 2.4 - 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 otra 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](https://dictionary.cambridge.org/dictionary/english/encryption)
2. [What is Data Encryption? | Kaspersky](https://www.kaspersky.com/resource-center/definitions/encryption)
3. [AES IP for hybrid cryptosystem RSA-AES | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/document/7348109)

## 2.5 - Estado del arte

En el artículo de S. Zhu et al [4] se explica cómo con la mejora de la capacidad de cálculo de las computadoras modernas el uso del aprendizaje profundo (deep learning) en el criptoanálisis, especialmente en la extracción automática de características en muestras de texto cifrado a gran escala, se está volviendo extensivo, incluso como parte de los ataques criptográficos perfilados.

1. [Time Sequence based AES S-box Implementation Cryptoanalysis using Deep Learning Approaches\* | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/document/9188503)

INVESTIGATION OF POSSIBILITY OF RSA SYSTEM BREAKING Considerable drawback of known methods of RSA system cryptoanalisys is a necessity to perform last step calculations with large dimension matrices. For example, an almost square matrix of size with 5037191 columns appeared in factorization of RSA-I60 number. For processing of the matrix one used powerhl computers which are not always available. Other possible approach to RSA system cryptoanalisys is obtaining of Euler function value. It is not necessary in this approach to use high speed computers.

[Cryptoanalysis of RSA system of enciphering with public key | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/document/1365964)

In 2008, Bafghi et al. [2] used a neural network model to represent the diferential operation of block ciphers to fnd the diference features. Te diferential feature space of the block cipher can be represented by a multilevel weighted directed graph, so the problem of fnding the best diferential feature can be transformed into the problem of fnding the least weight multibranch path between two known nodes in the directed graph. In this paper, the cyclic neural network (RNN) is used to fnd the path by minimizing the network cost function in the diferential operation graph of the block cipher. In 2010, Alallayah et al. [3–5] performed neural network-based cryptanalysis on classical cryptography, sequence ciphers, and simplifed DES (SDES). Tey regard cryptanalysis as a black box problem, using neural networks as an ideal tool for identifying black boxes, combining system identifcation technology with adaptive system technology, and constructing a neuroidentifer that simulates the target cryptosystem. A neural model and the key can be determinedfrom a given pair of ciphertexts. In 2012, Alani et al. [6, 7] implemented a known plaintext attack based on neural networks for DES and 3DES. In the experiment, they trained a neural network to retrieve plaintext from ciphertext without having to retrieve the keys used in encryption. Tis kind of attack is successfully applied to DES and 3DES. For DES, an average of 211 pairs of ciphertext pairs are needed, and cryptanalysis can be completed in about 51 minutes to achieve a global deductive efect. For the 3DES cryptanalysis, only an average of 212 plaintext-ciphertext pairs are needed, and the analysis can be completed in about 72 minutes. Corresponding results signifcantly reduce the number of known plaintexts required and reduce the time required to perform a full attack compared to other attacks. In 2013, Bahubali Akiwate et al. [8] proposed a neural network-based cryptanalysis of the DES algorithm, aiming to analyze the nonlinear characteristics of DES through neural networks. Te text data is used as the plaintext in the analysis, and the ciphertext encrypted by DES is used as the input of the neural network and is trained together with the plaintext to obtain the corresponding output. Te neural output is compared with the plaintext, and performance error indicators are established for analyzing the data to improve efciency. In 2014, Danziger et al. [9] used the method in [5] to fnd out the mapping relationship between plaintext, ciphertext, and key in SDES. When experiments are performed using 102,400 sample data, the keys of the frst, second, and ffh bits of the 10-bit key can be obtained; when the number of samples is reduced to 2000, the keys of the frst and second bits can be obtained. Together with the diferential analysis, they improved the S-box of SDES, which reduced the correlation between adjacent keys in the key space, making it resistant to key restoration attacks in neural network cryptanalysis. Inspired by [6, 7], this paper uses the methods in [6, 7] to construct an analysis framework similar to them. According to the block length and data type of AES algorithm, the data processing process suitable for AES algorithm is set. Ten, according to the research object and the structural characteristics of neural network, the specifc experimental system structure is designed. Te AES algorithm in ECB and CBC mode is used to perform cryptanalysis using neural network. Without knowing the key, the plaintext is restored from the ciphertext, trying to achieve the efect of global deduction. Te results show that the plaintext restored by the AES-128 and AES-256 algorithms over the neural network is more than 63% higher than the plaintext compared with the original plaintext. Section 2 of this paper introduces the specifc process of cryptanalysis based on neural networks. Section 3 designs the steps of the overall experiment and selects the relevant parameters. Section 4 is the experimental results. Section 5 gives the conclusion. 2. Cryptanalysis Process Based on Neural Network Te cryptanalysis method based on neural network utilizes the learning ability of the neural network to train the neural network with the known Ming ciphertext. Afer the training is completed, the neural network can restore the plaintext from the ciphertext that does not belong to the training set. Te corresponding system structure is shown in Figure 1. Te system contains an information forward propagation process (ciphertext �→ neural network �→ plaintext), and an error backpropagation process (plaintext �→ error function �→ weight correction �→ neural network). Te ciphertexts are input in the neural network, and the output results are compared with the known plaintexts to obtain an error function. Te weight is continuously corrected according to the error until the neural network is successfully trained. And fnally the plaintext can be restored with a certain probability. Tis attack method is considered to be a global deductive attack, which is functionally equivalent to the original decryption algorithm without knowing the key. Tis analysis method is similar to the global approximation method for multilayer feedforward neural networks in [11]. In order to train a neural network with an acceptable error rate, it is necessary to expand the network size, so it is necessary to increase the time of each training cycle. Tere are many related parameters that need to be set in the neural network, such as the number of neurons, the number of hidden layers, the training function, etc. Tese parameters will be specifed in Section 3.

[6868506.pdf (hindawi.com)](https://downloads.hindawi.com/journals/scn/2018/6868506.pdf)

The obtained results during the experiments show that RL can help to discover an attack strategy that can overcome a behavior-based anti-ransomware protection. It is worth noting, that the experiment was conducted on the Ransomware Detector that only represents the limited number of detection methods imitating the behavior of a real EDR solution. However, the presented results look promising and the proposed RL approach for anti-ransomware testing can be further applied to the anti-malware products with antiransomware modules available on the market that mostly rely on behavior analysis. The future work can be also conducted on applying the RL approach to network penetration testing. So, the Agent can learn how to discover the optimal attack path that can be used by an ethical hacker in security testing of a service or product.

[Reinforcement Learning for Anti-Ransomware Testing | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/document/9225141)

Accuracy is the main concern in the field of detection. A strong detection system will detect ransomware in less time with high accuracy. We address the problem of ransomware population drift by proposing a novel adaptive preencryption crypto-ransomware early detection framework that will deal with the population drift concept and limitations of existing research work. It will be unique work in terms of its working mechanisms. It is an imperative solution to all the existing detection systems for cryptoransomware attacks that encrypt user files and making them inaccessible for legitimate users. The contribution of the proposed study is described below: 1. An enhanced pre-encryption boundary definition and features extraction scheme to weight the preencryption features more accurately. 2. An adaptive pre-encryption crypto-ransomware early detection model by training an online classifier with data and features extracted and selected from pre-encryption boundary definition to increase the detection accuracy of both known and zero-day attacks

Different studies were carried out relating to the preencryption and population drift concepts. Some of them are briefly explained below: A study given by [17] deals with the detection of malware using early data of execution but this study was having low accuracy and a high false alarm rate. Another approach was followed for ransomware detection by monitoring user files. User files were observed for malicious changes done to these but studies including this approach cannot distinguish between changes done by a benign program and malicious program [25]. So these studies also carried high false alarm generation [14, 16]. In [18] a machine learning-based ransomware classifying approach named EldeRan was proposed. It selects the features which help to find ransomware. This work also discussed the limitations of Dynamic analysis. An obfuscation detector named ANDRODET for crypto-ransomware was proposed in [28]. The system was developed for the android platform thus working in an incremental and online manner. Identifier renaming, control flow obfuscation, and string encryption obfuscation issues were addressed. A feature extraction scheme was proposed in [29]. presented scheme DPBD-FE presented the dynamic threshold for defining the boundary definition. The proposed work was able to accurately extract the related APIs for the early detection of crypto-ransomware. A feature selection approach was discussed in [30] which deals with feature redundancy and overfitting issues. An integration of redundancy gradual up weighting and mutual information was done to get the most relevant feature selection. In [24] an ensemble learning-based early detection model was proposed. They utilized the ibagging technique for early detection. This work was presented to overcome the shortcoming of limited data before the encryption process. An enhanced selection technique for the selection of most informative features was also proposedwhich helps to increase accuracy and low false alarms. These relevant features were used for the training of the detection model. A context-aware and adaptive malware detection study was presented in [25] for android systems. This study considered the concept of malware population drift. Graph kernel was used to find malicious programs. In [31] an IRP-API-based pre-encryption method correlation cannot be true all the time. Cryptography related API cannot correlate with the IRP of the interacting process at the same time. If multiple processes are loaded then an API can be from one process while an IRP can be from any other process. Secondly, an API can be generated at time t1 and IRP at time t2 so in that case correlation can be undescribed

[A proposed Adaptive Pre-Encryption Crypto-Ransomware Early Detection Model | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/document/9392548)

In this paper, we present an enhanced neural cell to incorporate attention in learning from ransomware sequences, known as ARI (Attended Recent Inputs). The ARI cell, while processing the input sequence, also learns from a recent history in the form of a subsequence. It learns attention weights corresponding to each recent input and uses their corresponding significance when processing the input. We present an implementation of the ARI cell with LSTM networks, called ARI-LSTM. We enhance the LSTM cell by incorporating ARI mechanism within the cell, and use the resulting neural network for sequence learning with ransomware. Through evaluation on a ransomware dataset for the Windows operating system environment, we show that ARI-LSTM improves the performance of an LSTM in detecting ransomware from emulation sequences. The paper first explains the significance of repeated local patterns in ransomware sequences and relates the use of attention mechanisms for such tasks. We then describe the ARI cell in detail with its LSTM adaptation. This is followed by the system description for using ARI-LSTM in ransomware detection. We then present results on a large dataset in Windows environment, concluding with a discussion of the described approach

[Attention in Recurrent Neural Networks for Ransomware Detection | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/document/8682899)

RANSOMWARE ANALYSIS The ransomware procedure takes diverse bearings relying upon the client activity and the way coming about because of the offenders after they get the payoff. The authors in [4], presented a diagram that portrays the means associated with the ransomware procedure. The ransomware process is explained in figure 1. The accompanying is the means recommended in ransomware process: 1. Infection contaminates the PC 2. Usefulness lost – clients read emancipate note 3. Client chooses to pay deliver (or not) 4. Due date broadened 5. Client chooses to pay subsequent to going of broadening due date 6. Usefulness either returned or lost for good depending if paid or not paid

A. Strategies for ransomware removal Ransomware dangers have distinctive procedures to assault casualties, which fluctuate from a simple level and stretches out to serious level. In this way, each sort of ransomware needs an uncommon way to deal with and is taken to be expelled from the framework. There are normal ways to deal with disposing of ransomware programs, while it's activated by numerous criteria, for example, the sort of working framework, and the machine demonstrate. Program assaults by ransomware are less demanding to evacuate contrasting with hard drive assaults particularly Master File Table(MFT) on the hard drive. The principal suggested activity amid ransomware assault the framework must be killed all together to be separated from programmers' server. This activity will keep the ransomware to go to other associated gadgets and systems. After the framework, has been killed, the machine ought to be booted up with the protected mode alternative. Protected mode permits just default projects of the working framework to be worked to settle basic issues in the framework. It's very prescribed not to erase the ransomware documents in the framework before it's perceived on the grounds that making this move by non-mastery individuals may make harm the framework records, and potential information misfortune due to interfering with the association with aggressors. Consequently, it ought to be taken circumspectly and painstakingly. In the first place, protected mode alternative may have distinctive key per the machine demonstrate. As a rule, the most PCs can be signed in into experimental mode by squeezing F8 key before the windows begin. Also, there are a few spots ought to be checked after the windows signed into an experimental mode for example, framework registry, run, assignment chief, and framework arrangements. Each place incorporates certain alternatives, which can be utilized to end ransomware from running in the framework. 1-System arrangement: this element contains numerous choices including startup programs while the working framework begins. Additionally, winding up the suspicious projects from running. This activity keeps ransomware from running in the following boot up. 2-Task supervisor: In this element, there is a tab called process demonstrates all the running programs: It's prescribed to stop suspicious and obscure projects, for example, ransomware dangers. 3-Looking for some specific documents in the registry framework records: This progression ought to be finished carefully altogether not to cause a genuine harm in the framework. Area requires being checked including: %localAppData%, %ProgramData%

[Strategies for Ransomware Removal and Prevention | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/document/8480941)

The main contributions of this work are the following: • We analyze the Avaddon business ecosystem and provide an step-by-step analysis of its technical capabilities, using advanced static and dynamic analysis. This analysis can be generalized to grasp an overview of how modern ransomware operates, since their modus operandi is similar. As a result of our study, we provide a set of indicators of compromise (IoCs) that may serve security analysts to develop further tools and countermeasures to detect Avaddon, such as signatures or heuristics. • We showcase a typographical error in the list of services that Avaddon checks to avoid re-infecting victims, which is also present in modern variants of another Ransomware family, i.e., MedusaLocker. Additionally, we highlight that some similarities on the code of both families hint that they are operated or developed by the same group. • We describe a method to recover the symmetric keys used for the encryption, thus allowing victims to recover the files from infected systems. Accordingly, we present and make publicly available a tool which can help victims to recover from these attacks in real time. While this tool was designed using the analysis of the first versions of Avaddon, we have confirmed that it still works with the most up-to-date versions of the ransomware, released in mid-January 2021.

<https://arxiv.org/pdf/2102.04796.pdf>

Ransomware is one of the most widespread and damaging threats that internet users face today [48] and is often classified by the type of encryption used [3]: • Symmetric Crypto-Ransomware (SCR) uses one key for both encryption and decryption allowing the attack to complete in a shorter time, reducing the chances of it being discovered. • Asymmetric Crypto-Ransomware (ACR) uses different keys for encryption and decryption. • Hybrid Key Crypto-Ransomware (HCR) firstly uses symmetric encryption to encrypt the user’s files as fast as possible. After which the symmetric key is encrypted using asymmetric encryption.

One important aspect of live forensics is the examination of the systems memory where the malicious code is running. This examination is normally performed off-line so that the contents of the memory are not affected by the examination or the memory capture tools used. To achieve this, the memory of the system to be examined needs to be captured and saved. According to Ruff[40] there are three main memory capturing techniques: 1. Software-based, which typically involves executing extraction programs. An issue with this approach being that the execution of software would impact the contents of the captured systems memory. 2. Hardware-based, which typically involves connecting devices, such as PCMCIA cards or USB sticks and are not always practical in live scenarios as physical access to the machine is required [31]. 3. Virtualization technology-based techniques. A detailed compilation of the techniques available is provided in [9] along with advantages and disadvantages of each approach. Using the virtualization approach, a snapshot of the analysed system’s volatile memory is extracted using tools provided by the virtualization software. This snapshot is then inspected by an analyst using a variety of specialised forensic tools[33]. Obviously to use this technique the system must be running in a virtualised environment. The advantages of this approach being that no trace of any extraction program exists in the captured memory and any running malicious programs are unaware that they are being analysed or that the memory dump was taken [12, 25, 30]. The challenge of memory acquisition in this context is to discover cryptographic artefacts, such as the encryption keys, in a manner that allows the target device to continue to operate normally, while the memory is being acquired [31]. 2.2. Cryptanalysis live forensics Some work has been previously performed into the possibility of using live forensic techniques to discover encryption keys that may be present in a computers memory. It was not possible to find any literature that focused specifically on ransomware in particular, however similar work on key determination in volatile memory has been performed for SSH tunnels, encrypted volumes, WinRAR, WinZip and Skype. [4, 26, 31]. Several research papers confirm the assumption that for a system to be able to encrypt/decrypt data, then the cryptographic algorithm needs to have access the encryption keys and these are normally held in volatile memory. Balogh [4] state that encryption in real-time is only performed in memory which means that the encryption keys must also be present there. So in the case of symmetric encryption it means that the keys also needed for decryption will also be recoverable from memory. With regards to key management, Maartmann-Moe [26] state that it is clear that cryptographic keys need to be present in memory during encryption when using standard computer hardware. In extensive tests conducted on 10 different cryptographic systems the researchers [26] were always able to retrieve all the cryptographic keys from memory for every application tested using their specifically developed tool called interrogate. While these researchers have not investigated ransomware specifically, their findings strongly suggest that it would be possible to extract ransomware cryptographic keys using similar techniques.

<https://arxiv.org/pdf/2012.08487.pdf>

The objective of this research work is to perform feature analysis in order to find the most discriminative features and sequences that can better classify the benign and attacked samples. For this purpose, RanSD (Ransomware Static and Dynamic Analysis) system is proposed, that follows the procedure as follow: 1. Firstly, generated static and dynamic datasets, by collecting different Ransomware samples from different sources. To generate dynamic data used Cuckoo Sandbox to analyze activities performed by any files in a controlled environment. Similarly, for the static dataset, PE File information was extracted from different executable files with the help of the built-in package of python library named as PE File. 2. By exploiting Wrapper based Mutual Information (MI), the most contributed behaviors in the detection of ransomware were reduced from 40,000 to 300. 3. Then, sequences followed by malicious and benign files were analyzed and some of them were extracted for further detection of maliciousness more efficiently. 4. Finally, passed the most contributing features to the machine learning based detection system. Transfer learning, using a pre-trained model on ImageNet of ReeNeT-18, has been applied along with SVM and Random Forest algorithm.

<https://arxiv.org/ftp/arxiv/papers/1910/1910.00286.pdf>

This paper investigates decrypting TLS communications of real-world malware. A framework uses a standard approach for decrypting TLS traffic to analyse and decrypt the secure communications. For malware, performance challenges can result from malware use of different cryptographic libraries. So, the framework is extended to accommodate the Windows cryptographic library. Experiments evaluate decrypting real bot and ransomware command and control communications using the extension. The contribution is a novel method to discovering cryptographic artefacts used by real malware. As these are discovered in single memory extracts and decrypted in less than a second, the communicated activities of unknown malware can be discovered

<https://arxiv.org/pdf/1907.11954.pdf>

contributions. To summarize, RWGuard makes the following contributions: 1. A decoy based ransomware detection technique that is able to identify ransomware processes in real-time. 2. A ransomware surveillance system that employs both process and file change monitoring (to detect ransomware encrypting files other than decoy). 3. A classification mechanism to distinguish benign file changes from ransomware encryption by hooking relevant CryptoAPI functions and learning the user’s file encryption behaviors. 4. An extensive evaluation of our ransomware detection system on 14 most prevalent ransomware families to date.

[(PDF) RWGuard: A Real-Time Detection System Against Cryptographic Ransomware: 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings (researchgate.net)](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)

Restoring the Encrypted Files By changing the name of the file vssadmin.exe the Ransomware could not fmd and delete the restore points created before infection, which made it possible to recover from one of these previously saved restoration points. After a full system restore the system was scanned using the antivirus software Avast, downloaded from avast.com, and the virus scan tool Malwarebytes, from malwarebytes.org. This search showed that the Trojan was not retained on the system after restoration of the shadow copy. The only artifacts that were recovered from the CTB locker and Teslacrypt infections were traces of the Crypto Ransomware in the fonn of images and text files with payment instructions. Locky removes itself completely from the system after infection. The infection and restoration procedure was repeated 10 times for each type of Crypto Ransomware family to get a more reliable result, and each time all files could be completely restored accordingly. C. Automation of Protection Procedure Since the result of the experiment shows that restoring files after amending vssadmin.exe was successful, a script was created to automate the solution in a simple manner. The script searches the system for the file vssadmin.exe in the system32 folder. If the file is available, the script asks the user for the ownership and required access, followed by a question if the user wishes to rename the file vssadmin.exe. After the name change is completed, the script creates a restore point. If the file vssadmin.exe is not to be found by the script it directly creates the recovery point. This makes it possible to use the same script both for enabling the protection, as well as creating subsequent recovery points.

[IEEE Xplore Full-Text PDF:](https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7924925)

Piloto experimental

* Implementación de redes neuronales para des encriptación
* Deconstrucción y análisis de ransomware a estudiar
* Mecanismos preventivos y correctivos contra ransomware
* Uso de aprendizaje automático e inteligencia artificial como mecanismos de defensa contra ransomware u otros tipos de malware afines
* Casos de estudio

# Capítulo 3 - Desarrollo del experimento (PASADO, 12-20 pag)

Durant

## Propuesta y descripción general del experimento

Piloto experimental

Text

Description automatically generated

## Configuración experimental

Configuración experimental

* Herramienta de virtualización
* Sistema operativo
* Versión o binarios de ransomware a usar
* Set de datos a usar en el experimento

## Recursos

Piloto experimental

* Herramientas, ambientes y versiones usadas.
* Código fuente
* Datos utilizados
* Métricas

## Parámetros de análisis

Piloto experimental

* Eficiencia del proceso de entrenamiento (uso computacional y tiempo de ejecución)
* Precisión (correctness) y completitud en la restauración de archivos
* Genericidad (genericity) de la solución

## Procedimiento

Piloto experimental

* Pasos ejecutados
* Variaciones
  + Cada tipo de ransomware
  + Re-ajustes de parámetros de entrenamiento de la red neuronal
  + Tipos de datos de entrenamiento
  + Ambientes de ejecución
* Exposición de los recursos en repositorios públicos para su reproducibilidad

## Reproducibilidad

Piloto experimental

## Resultados

Parámetros de análisis

# Capítulo 4 – Discusión (PASADO)

Durante

Análisis de resultados

Pruebas específicas y variaciones

Discusión

* Obstáculos
* Retos
* Ventajas y desventajas del método propuesto

# Capítulo 5 – Conclusiones (PASADO, 2-6 pag)

Durante

## Hallazgos

Piloto experimental

## Líneas de investigación futuras

Piloto experimental

# Referencias

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. A. 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

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

Conjunto de datos

Base de datos

Archivos de texto plano

AES

Bitcoin

script

A brief background on cryptography and cryptanalysis

Cryptography (from greek: hidden or secret writing) is the practice of techniques that prevent third parties or the public from reading private messages. It uses encryption to scramble plaintext (the unencrypted data) into ciphertext (the encrypted data) to obfuscate the content of the message to third parties looking in.

There are 4 main objectives to cryptography:

Confidentiality: only the intended recipient can understand the message,

Integrity: the message cannot be edited without the tampering being detected,

Non-repudiation: the sender cannot deny their intentions in creating the message, and

Authentication: the sender and recipient can identify that they are communicating with one another.

Modern cryptography exists at the intersection of the disciplines of mathematics, computer science, and electrical engineering, and has broad applications including military communications, electronic commerce, ATM cards, computer passwords. Basically anything where secure communication is essential.

Cryptanalysis, defined as the study of cryptographic systems with the goal to find weaknesses and retrieve the contents of encrypted messages, evolved in parallel with cryptography. Even though the word was only coined in the 1920s, the practice of breaking codes and ciphers dates back as far as the 9th century. Cryptanalysis played a decisive part during World War II to decipher German intelligence secured by Enigma machines, which used an unencrypted symmetric key for both the sender and receiver of the message.

Since then, the emergence and explosive growth of computing power allowed ever more complex encryption methods that are not only used to secure your email messages and web traffic but also, sadly, to hold your files for ransom by cyber criminals.

Encryption methods: Symmetric v asymmetric

Encryption methods have become incredibly complex compared to the earliest substitution ciphers used by Julius Caesar, where the plaintext letters are simply replaced by shifting it back or forth a fixed amount of positions along the alphabet.

Since the structure of the plaintext is the same as the encrypted message, it’s easy to spot patterns using frequency analysis to identify how often and in which sequence the characters appear.

Symmetric encryption: Fast but vulnerable

Yet the above cipher is a simple example of how symmetric encryption works: both the sender and the receiver shift the letter the same fixed number of positions. This number of positions is what is referred to as the “key”. It is therefore imperative that the key is kept secret, which is why symmetric encryption is often referred to as “secret key encryption”.

There are two major types of symmetric encryption: Block ciphers and stream ciphers.

As the name implies, block ciphers operate by encrypting in byte-sized blocks using the same key. The message itself is split across multiple blocks depending on the length, and predictable data is appended to extend it to a full block in a process known in cryptography as padding. The most common block ciphers you’ll come across are AES and Blowfish, the former of which is often used in ransomware encryption (more on that further down in the article).

Stream ciphers, on the other hand, encrypt each digit of the plaintext at a time (usually in the form of a bit), with the help of a pseudo-random key stream. This means a different key from the stream is used for each of the bits. A mathematical operator “exclusive or” (XOR, for short) then combines the the two to create the ciphertext. Common stream ciphers in use today include RC4 and Salsa20.

What characterizes each of these symmetric encryption methods is that they have low computational requirements to function, and use only one key for both encryption and decryption of the message.

The latter raises an important question however: how do you secretly communicate the key without someone listening in on the exchange?

Asymmetric encryption: Secure but slow

Asymmetric encryption, also known as public key encryption, has the answer: it uses a pair of keys, consisting of a public and a private one.

The public key is intended to be distributed widely, while the private key is only in the possession of the key pair owner. Messages encrypted with the public key can only be decrypted using the private key, while messages encrypted with the private key can only be decrypted using the public key.

As a result, asymmetric encryption algorithms not only allow you to encrypt and decrypt messages, but also allow for authentication of that message, as only the private key owner can create a message decryptable by the public key.

At the base of most asymmetric cryptography systems is usually an algorithm that requires computationally heavy operations. The most common in use today are Diffie-Hellman-Merkle, RSA and Elliptic Curve Cryptography. While explaining each of these would be beyond the scope of this article, there’s a fairly in-depth article on ArsTechnica if you’re brave enough to dive into the maths behind them.

The bottom line is that asymmetric encryption allows for relatively secure encryption without the need of a shared secret key, yet the complex computational nature makes it unfeasible for large sets of data.

Given that both encryption types have their unique advantages and disadvantages, most implementations (including ransomware authors) will use a combination of both: symmetric encryption with a randomly generated key, usually referred to as the session key, to encrypt the actual message or files, then an asymmetric algorithm to encrypt the session key used.

Common cryptology terms

Message: The data that you want to protect through encryption; this can be text as in an actual message or a file

Plaintext: The unencrypted message

Ciphertext: The encrypted message

Key: The component that is used by an encryption algorithm to turn a plain text into a ciphertext and vice versa.

Block cipher: symmetric encryption in blocks of bytes.

Stream cipher: use of a keystream to encrypt one bit at a time.

Symmetric (secret) key encryption: The same key is being used for encryption and decryption of a message.

Asymmetric (public) key encryption: Different keys are being used for encryption and decryption of a message.

How ransomware authors use encryption: CryptoLocker

To see how exactly malware authors use encryption for their nefarious purposes we are going to take a look at the one ransomware family that was so prolific and widespread that its name is, to this day, used synonymously for all ransomware: CryptoLocker.

CryptoLocker was certainly not the first file encrypting ransomware, but it definitely was the first major ransomware family that got widespread media attention. It uses 256 bit AES symmetric encryption for the actual file encryption, and asymmetric RSA encryption for communication and securing the symmetric session key.

Unsurprisingly, the way CryptoLocker operates has become a blueprint for many other ransomware families that followed, making it an ideal case study to show how it uses encryption to lock up your files.

Establishing secure communications

When it arrives on a system, CryptoLocker comes with nothing more than a RSA (=asymmetric) public key, used by the ransomware to establish a secure channel to its command and control server. The ransomware handles all communication between itself and the malware author’s server via this channel. Using public key encryption has two major advantages in this case:

Any third parties listening in on the network communication won’t be able to see the plaintext messages being exchanged between CryptoLocker and its server. All a malware analyst would see when trying to understand the protocol by sniffing the network traffic is a bunch of encrypted gibberish.

More important however is that malware authors not only hide their messages from prying eyes, but also ensure that the server the ransomware is talking to belongs to the malware authors.

Remember: Any public key only decrypts messages that were encrypted by their corresponding private key, which only the malware author will have access to.

As a result, encrypting the communication with RSA not only ensures its secrecy but also its authenticity. This way, a law enforcement agency seizing command and control domains can’t simply take over control of the malware by issuing its own commands.

Encrypting the files

During the communication, CryptoLocker will request a second RSA public key from its server that is unique to the victim. It then goes ahead and creates a 256 bit AES session key that it will use to encrypt the victim’s files. As mentioned before, asymmetric cryptography like RSA isn’t well suited to encrypt large amounts of data directly as it is relatively slow compared to its symmetric cousins. Using a symmetric algorithm like AES to encrypt the bulk of the user data is therefore much more efficient.

Securing the shared key

As a final step, CryptoLocker encrypts the 256 bit AES key using the victim-specific, asymmetric RSA public key and stores it together with the encrypted file data.

Once the encryption process finishes, the ransomware will erase the AES session key from its memory, making sure no trace is left anywhere. Only the owner of the victim’s private key, which was generated and is stored only on the malware author’s server, is able to decrypt the AES session key from within the encrypted files and decrypt the files again once the victims have paid the ransom.

Ransomware leverages the advantages of both asymmetric and symmetric encryption to lock up the victim’s files within a matter of seconds, rather than hours. Recovering them without paying the criminals is almost impossible.

<https://blog.emsisoft.com/en/27649/ransomware-encryption-methods/>

Combating ransomware is difficult for a number of reasons. First, this malware is easy to obtain or create [48] and elicits immediate returns, creating lucrative opportunities for attackers. Second, the operations performed by such malware are often difficult to distinguish from those of benign software. Finally, ransomware often intentionally targets unsophisticated users who are unlikely to follow best practices such as regular data backups. Accordingly, a solution to automatically protect such users even in the face of previously unknown samples is critical. In this paper, we make the following contributions:

• Develop an early-warning system for ransomware: CryptoDrop is fundamentally different from existing methods of detecting ransomware, which inspect programs and their activity for malicious characteristics. Our system is the first ransomware detection system that monitors user data for changes that may indicate transformation rather than attempting to identify ransomware by inspecting its execution (e.g., API call monitoring) or contents. This allows CryptoDrop to detect suspicious activity regardless of the delivery mechanism or previous benign activity. Our system does not attempt to prevent all files from loss and is not intended to replace a user’s normal anti-malware software; rather, CryptoDrop is designed to be effective even when the user’s anti-malware software has failed to block the malware. Our system is built on Windows, a platform frequently targeted for ransomware attacks, providing a realistic solution to “inthe-wild” threats. In doing so, we attack the core behavior of ransomware in a novel and practical manner that other anti-malware technologies fundamentally cannot.

• Identify three primary indicators suited to detect malicious file changes: These indicators each measure an aspect of a file’s transformation, and when all three have manifested, a ransomware file transformation has likely occurred. This union indication assists CryptoDrop in reliably detecting ransomware while incurring few false positives. These indicators have not been previously employed in a ransomware detection system, and our analysis of their effectiveness in isolation and unison provides insight into the ability to detect ransomware.

• Perform most extensive analysis of encrypting ransomware to date: Demonstrate a 100% true positive rate over 492 distinct ransomware samples across 14 families after as few as 0 and a median of 10 (0.2%) files lost from our test corpus. Finally, we discuss the observed behavior of our samples and discuss how CryptoDrop remains robust despite the significant behavioral differences between families. Through reduction of the number of files lost, we demonstrate that CryptoDrop reduces the need for the victim to pay the ransom, choking attackers’ revenue and rendering the malware ineffective

<https://regmedia.co.uk/2016/10/27/scaife-icdcs16.pdf>

The goal of this work is to understand whether ransomware can be identified, with a high degree of accuracy, using a limited number of characteristic features and before infecting victims. Furthermore, EldeRan provides an automatic way for creating signatures for new variants of ransomware families. EldeRan complements well AVs signature-based mechanisms, as it can be used to identify cases where AVs may have missed new, or unknown, ransomware families. The main contributions of this paper are: • We propose EldeRan, a framework to identify the most significant ransomware dynamic features, and use them to detect ransomware. Through the Mutual Information criterion, we have identified the most relevant dynamic features amongst a large set of considered ones. EldeRan exploits a relatively small set of features without reducing the performance of the machine learning classifier. By following this approach, EldeRan is also well suited to detect new ransomware families. • We have evaluated the accuracy of the Regularized Logistic Regression by comparing it against other machine learning classifiers, namely the Support Vector Machine (SVM) and Naive Bayes. We found that the Logistic Regression outperforms Naive Bayes and it is competitive with respect to the SVM. Moreover, the Logistic Regression is easy to train and adapt compared to the SVM. The regularization technique applied to the Logistic Regression helps the classifier to generalize better to unseen samples by preventing overfitting. • We compare the classification results with those of VirusTotal: EldeRan’s average error rate is 2.4% while that of VirusTotal is 5.6%, and EldeRan achieves a remarkable 96.3% detection rate. We also tested the ability of EldeRan to detect new families of ransomware, obtaining an average detection rate of 93.3%.

Recovering from Ransomware The recovery of a system compromised by locker-ransomware can be usually done, for example, by rebooting the system in safe-mode, and running an on-demand virus scanner, or through similar system-restore techniques. On the other hand, recovering the data from crypto-ransomware is usually a more challenging task. In fact, unless the user pays the ransom, and provided cyber-criminals are willing to give in exchange the key (which is usually the case, as they want to preserve their “reputation”), in general it is very hard to recover the encrypted data (i.e., computationally infeasible). Usually, the private key is either stored on the server, or it is stored encrypted on the user storage (sometimes on the header of the encrypted files themselves) with public key mechanism. Obliviously, in case the encryption mechanism is not implemented correctly, e.g. cyber-criminals design and develop their own encryption algorithm, or the key length is too short, or there are flaws in the protocol [34], it is possible to recover the data. However, note that usually this is not the case as ransomware, such as TeslaCrypt 2.0 exploits very sophisticated key generation mechanisms [33]. In some cases, the private keys is stored in clear (i.e., not encrypted with a public key) on the victim’s disk, and they can be recovered. For example, in some GPcoder variants, the keys are stored in a registry key, and, hence, it is possible to recover the key. In other cases, files can be recovered by simply rebooting the machine before the ransomware terminates, as sometimes files are zeroed only after completing the encryption. We have also to consider those cases where the encryption keys is (by mistake) deleted after use, and not stored in the attacker’s server [6]. In these case, the only way users can recover their files is if they restore them from a backup. Ransomware may also sometime delete shadows copies containing old copies of files [25]. Note that some forensics techniques might be able to recover some files (or retrieve the key) or, similarly, sometimes cyber-criminals might disclose the private keys, e.g. after regretting their actions [21]. Alternatively, the law enforcement might be able to access the database of encrypting keys or, in some other case, some AV vendors have made available an online repository of recovered decryption keys, e.g. for CoinVault and Bitcryptor2 , or an online repostory of several decrypters for free3 .

<https://arxiv.org/pdf/1609.03020.pdf>

Ransomware breaches the victim’s computer in the same way as the other computer virus such as Trojan horse, worms, and spyware. But they differ in the way they attack the victim’s system as Trojan horse gets into a system by hiding inside other soft wares, such as an email attachments and downloads. Worm on the other hand are most prevalent type virus, which spreads itself not only from file to file but also from computer to computer also via email and other internet traffic [6]. Worm finds the email address book of the victim computer and help themselves to addresses and using victim’s email addresses as return addresses it sends themselves to all the contacts. Whereas the spyware is a client-side software component that is mainly used for monitoring the client activity and then send all the data to a remote machine. These spyware mostly comes hidden in free downloadable software, it also uses the computer’s CPU and storage for some tasks about which victim knows nothing [6]. Information collection goes on even if the system is offline also and moreover spyware can stay on the victim’s computer long after they’ve uninstalled the original software. Different from the other malicious programs, the ransomware mainly seeks the system vulnerabilities potentially caused by its precedents. Ransomware is the second-generation of the malicious software which typical targets the system which are already victimized earlier by a worm or Trojan and then grabs a slew of files [8]. Ransomware mainly encrypts the files or locks the system and hence makes it inaccessible for the victim. Spreading of ransomware can be several ways including through spam or drive-by download that exploits the browser’s vulnerability.

[Comprehensive Survey on Petya Ransomware Attack | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/document/8520323)

Definition 1 A neural network is a pamllcl, distributed information processing structure consisting of processing elements (which can possess a local memo y and can carry out localized information processing operations) interconnected together with unidirectional signal channels called connections. Each processing element has a single output connection which branches ("fans out") into as many collateml connections as desired (each canying the same signal - the processing element output signal). The processing element oatput signal can be of any mathematical type desired. All of the processing that goes on Mthin each processing elemeni must be completely local: i.e., it must depend only upon the camnt values of the input signals am'uing at the processing element via impinging connections and upon values stored in the processing element 'a local memory. The importance of restating the neural network definition relates to the fact that (as pointed out by Carpenter and Grossberg [E]) traditional forms of the backpropagation architecture are, in fact, not neural networks. They violate the locality of processing restriction. The new backpropagation neural network architecture presented below eliminates this objection, while retaining the traditional mathematical form of the architecture. The backpropagation neural network architecture is a hierarchical design consisting of fully interconnected layers or rows of processing units (with each unit itself comprised of several individual processing elements, as will be explained below). Backpropagation belongs to the class of mapping neuml network architectures and therefore the information processing function that it carries out is the approximation of a bounded mapping or function f : A C R" --\* Rm, from a compact subset A of n-dimensional Euclidean space to a bounded subset f[A] of m-dimensional Euclidean space, by means of training on examples (XI, yl), (xz,y~), . . . , (xk, yk), . . . of the mapping, where yk = f(xh). It will always be assumed that such examples of a mapping f are generated by selecting xr vectors randomly from A in accordance with a fixed probability density function p(x). The operational use to which the network is to be put after training is also assumed to involve random selections of input vectors x in accordance with p(x). The backpropagation architecture described in this paper is the basic, classical version. Many variants of this basic form exist (see Section 5). The macro-scale detail of the backpropagation neural network architecture is shown in Figure 1. In general, the architecture consists of h’ rows of processing units, numbered from the bottom up beginning with 1. For simplicity, the terms mw and layer will be used interchangeably in this paper, even though each row will actually turn out to consist of two heterogeneous layers (where the term layer is used to denote a collection of processing elements having the same form of transfer function). The first layer consists of n fanout processing elements that simply accept the individual components I; of the input vector x and distribute them, without modification, to all of the units of the second row. Each unit on each row receives the output signal of each of the units of the row below. This continues through all of the rows of the network until the final row. The final (Iith) row of the network consists of m units and Droduces the network’s estimate y’ of the correct output vector y. For the purposes of this paper it will always be assumed that I( 2 3. Rows 2 thr; I< - 1 .. .\_ are called hrdden rows (because they are not directly connected to the outside world)

Besides the feedforward connections mentioned above, each unit of each hidden row receives an ‘error feedback’ connection from each of the unit.s above it. However, as will be seen below, these are not merely fanned out copies of a broadcast output (as the forward connections are), but are each separate connections, each carrying a different signal. The details of the individual “units” (shown as rectangles in Figure 1) are revealed in Figure 2 (which depicts two units on adjacent rows and shows all of their connections). Note that each unit is composed of a single sun processing element and several planet processing elements. Each planet produces an output signal that is distributed to both its sun and to the sun of the previous layer that supplied input to it. Each planet receives input from one of the suns of the previous layer as well as its own sun. As stated above, the hidden row suns receive input from one of the planets of each of the suns on the next higher row. The output row suns receive the ‘correct answer’ yi for their component of the output vector on each training trial. As discussed in detail below, the network functions in two stages: a forward pass and d backward pass. A scheduling processing element (not shown) sends signals to each of the processing elements of the network telling it when to apply its processing element transfer function and whether to apply the forward pass part of it or the backward pass part of it. After the transfer function is applied, the output signal is latched to the value determined during the update. This value is therefore constant until the next update. The exact equations of the processing elements of the network are given in Table 1

<http://www.andrew.cmu.edu/user/nwolfe/esr/pdf/backprop.pdf>

## 2.3 - Backpropagation

Piloto experimental