UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

DISEÑO E IMPLEMENTACIÓN DE SISTEMA AUTOMATIZADO DE RECOLECCIÓN Y ANÁLISIS DE MENSAJES DE TEXTO FRAUDULENTOS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

GERARDO MATÍAS TRINCADO MUÑOZ

PROFESOR GUÍA:
EDUARDO RIVEROS ROCA

MIEMBROS DE LA COMISIÓN:
BENJAMÍN BUSTOS CÁRDENAS
MAURICIO CERDA VILLABLANCA

SANTIAGO DE CHILE
2024

Resumen

Los fraudes mediante telefonía han ido en aumento junto con la expansión de la tecnología. Esta memoria aborda el problema de los fraudes realizados mediante mensajes de texto, desarrollando un sistema automatizado de recolección y análisis de mensajes de texto fraudulentos, con el propósito de realizar reportes a las autoridades a tiempo y prevenir este tipo de fraudes. El sistema, de código abierto, da el primer paso para futuros estudios sobre el comportamiento de las estafas y posibles nuevas implementaciones para automatizar la detección de actividades sospechosas.

Para esta memoria, se utilizaron dispositivos como una Raspberry Pi 5 junto con una serie de módems USB conectados a ella.

Para la implementación del sistema, fue necesario obtener datos de mensajes tanto fraudulentos como legítimos para entrenar un modelo de clasificación de datos. El modelo se entrenó utilizando el método *Random Forest*, que posteriormente serviría para clasificar los mensajes antes de su almacenamiento.

Se implementó una base de datos local dentro del sistema, que permite el acceso y almacenamiento de los datos recogidos por los dispositivos conectados. Se desarrolló un *backend* que gestiona la lógica necesaria para el procesamiento de los datos y la generación de reportes automáticos. El *frontend* del sistema ofrece una visualización de las métricas de los datos obtenidos, implementando gráficos con la información procesada por el *backend*.

El proyecto demuestra la viabilidad de la automatización para la prevención de fraudes, con la posibilidad de extenderse y adaptarse a futuros avances. El enfoque principal es fortalecer la privacidad y seguridad de los usuarios de teléfonos móviles en Chile.

A mi madre y familia, por su amor y apoyo incondicional.

A mis amigos, por su constante ánimo y compañía.

A mis mentores, por su inspiración y guía.

Agradecimientos

Quisiera expresar mi más sincero agradecimiento a todas las personas y entidades que han contribuido al desarrollo y finalización de este trabajo de título.

En primer lugar, agradezco a mi profesor guía, Eduardo Riveros, por su valiosa orientación, paciencia y apoyo continuo durante todo el proceso de desarrollo e investigación. Su apoyo ha sido parte fundamental para llevar a buen término este trabajo. En conjunto, doy gracias al CSIRT de gobierno por brindarme herramientas necesarias para este proyecto.

Agradezco a la Universidad de Chile y al Departamento de Ciencias de la Computación por proporcionarme los recursos y el entorno necesarios para llevar a cabo esta investigación. Un agradecimiento especial a mis compañeros del Laboratorio de Criptografía Aplicada y Ciberseguridad, quienes siempre estuvieron dispuestos a ayudar y colaborar. Quisiera hacer mención en especial a los profesores Alejandro Hevia y Camilo Gómez, por sus comentarios, retroalimentación y enseñanzas durante el transcurso del proyecto.

Deseo expresar mi gratitud a mi familia por su apoyo incondicional y comprensión durante estos años de estudio. A mi madre, Soledad, por su amor y cuidados en mi trayectoria, a mis hermanos Francisco y Fabian, por su confianza y motivación. Presento una gratitud especial a Edulia, mi abuela, quien junto a mi madre, ha sido la persona que se ha encargado de mi crianza y me ha ayudado a llegar hasta donde estoy. Doy gracias a mis amigos, por estar siempre ahí para ofrecerme ánimo y momentos de distracción necesarios.

Agradezco a Catalina, por su entendimiento, a Luke, por su buena acogida y disposición, a Marcel, por su guía y agradezco además a Ana y mis primos, a Sara y familia, a mis tíos Luis y Herminda, asímismo a Mario, Cecilia y familia, y a Erik por inspirarme a investigar dentro de la ciberseguridad.

Finalmente, agradezco a todas las personas que, de alguna manera, contribuyeron a la realización de este trabajo de título. Sin su apoyo, este trabajo no habría sido posible.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Introducción

El crecimiento tecnológico se extiende día a día, aumentando también las maneras de cometer crímenes en una era que cada vez se vuelve más digital, lo cual también trae consigo un aumento en las precauciones necesarias al circular por la red. Los usuarios de elementos tecnológicos con acceso a internet se ven expuestos a distintos tipos de estafas, las cuales pueden llevar a robo de datos bancarios o filtraciones de datos personales. Para prevenir las vulnerabilidades a que pueden sufrir los usuarios de la red, es necesario implementar distintas medidas de seguridad para proteger la privacidad e integridad de los usuarios, pues *Internet World Stats* [20] indica que la penetración en el internet afectaba al 67.9 % hasta el año 2023, lo cual deja en evidencia que una mayoría de la población se encuentra expuesta a los riesgos del internet. En las estadísticas presentadas, se tiene que América Latina presenta un 80.5 % de penetración en el internet, lo cual vuelve a este continente uno de los más propensos a caer en el cibercrimen.

En el contexto internacional del cibercrimen, se tiene que las estafas informáticas surgen con un alto grado de importancia debido al aumento en la cantidad de cibercrimen durante los últimos años. El FBI anunció que el *phishing*, nombre dado a un tipo de ciber estafas, era uno de los ciberdelitos más prevalentes, estableciendo en el reporte anual del *Internet Crime Complaint Center* [13] que aproximadamente el 22 % de todas las filtraciones de datos son producidas por *phishing*, además, se establece que el 83 % de todas las empresas han experimentado ataques de *phishing*. Para el informe anual del 2022 [14], el *phishing* es el delito más popular en el internet, teniendo en total 300, 497 víctimas en Estados Unidos reportadas por la entidad durante ese año.

Los casos de delitos informáticos en Chile han aumentado durante el transcurso de los años. Sofía Álvarez indica en su nota [31] que los delitos informáticos aumentaron un 61 % en 2021 y 2022 según cifras publicadas por la Policía de Investigaciones. La brigada de cibercrimen de la PDI indicó en un artículo de prensa [25] que, tras la pandemia, los delitos informáticos fueron en aumento, teniendo un crecimiento de un 30 % en las estafas informáticas contra particulares durante este periodo que provocó un alza en el uso de medios digitales.

El *phishing* se encuentra entre los ciberdelitos más prevalentes, este se refiere a la práctica de obtención de datos personales mediante estafas, usualmente realizadas utilizando páginas web fraudulentas. Las vías más frecuentes para las estafas mencionadas son el correo electrónico y los servicios de mensajería, en particular los mensajes de texto (SMS). Las credenciales robadas son utilizadas para diversos fines, variando desde datos filtrados o ven-

didos, y en algunos casos, es robada información bancaria, siendo el caso más común de esto el robo de información de tarjetas de crédito.

En la actualidad, existen campañas de *phishing* realizadas de manera exclusiva utilizando servicios de mensajería móvil, lo cual da fruto al llamado "*smishing*", correspondiente a los fraudes ejecutados aprovechando el uso de la mensajería móvil.

Un ejemplo de una campaña de *smishing* resulta ser el caso de una alerta levantada por el CSIRT sobre una página que suplanta a Correos de Chile [8], dedicada a enviar repetidos mensajes a los usuarios de telefonía. El sitio web utilizado para estafar busca obtener datos personales seguidos de datos bancarios, indicando a las víctimas que deben pagar por un paquete retenido en aduanas. Además, la página fraudulenta resulta sólo estar disponible en su versión móvil, imposibilitando la interacción de usuarios que intenten acceder en dispositivos de escritorio. Es posible denotar las herramientas utilizadas para la campaña, pues la página utilizada contaba con certificados TLS gratuitos, lo cual genera una sensación de falsa seguridad al visitar el sitio.

Dado este panorama, se vuelve crucial desarrollar estrategias que permitan detectar, entender y prevenir campañas de *smishing*. Una de estas estrategias es el uso de un sistema de muestreo, diseñado registrar mensajes de texto e identificar intentos de ataques, permitiendo así un análisis más profundo de las tácticas y técnicas utilizadas por los cibercriminales. Actualmente, existen sistemas de *honeypots*, es decir, anzuelos preparados y enfocados en atraer ataques más convencionales y no en formas específicas de *phishing* como el *smishing*.

Este informe trata sobre la implementación de un sistema capaz de capturar mensajes de texto e identificar intentos de *smishing* utilizando módems USB para conectarse a la red celular y obtener mensajes de texto.

Capítulo 1

Objetivos

1.1. Objetivo General

El objetivo general de este proyecto es diseñar e implementar un sistema de captura de mensajes de texto capaz de identificar intentos de *smishing*, creando una plataforma automatizada y de código abierto, utilizando una *Raspberry Pi* y módems USB para recibir mensajes, con el fin de detectar y registrar campañas de *smishing*.

1.2. Objetivos Específicos

1. Investigar sobre acciones y conductas que atraen ataques de *smishing*.

2. Documentar cómo operar con módems USB de telefonía celular para enviar y recibir SMS

3. Conectar módems USB con un servidor, al cuál se deben enviar los datos de manera automática.

4. Investigar y analizar el comportamiento de los mensajes de *smishing*, identificando patrones y tácticas comunes en ataques de *smishing*.

5. Implementar métodos de recolección de datos sobre los intentos de *smishing*.

6. Realización de pruebas del sistema utilizando el reenvío campañas de *smishing* previamente capturadas por el CSIRT.

1.3. Solución Propuesta

Para abordar el problema del *smishing*, se propone una solución consistente en dos partes, siendo estas la investigación y la implementación respectivamente. La investigación proporcionará conocimientos fundamentales sobre el *smishing*, tales como su uso, fines, prevención e identificación. La implementación, en cambio, garantizará la recopilación activa de datos relevantes para el estudio de los ataques.

El CSIRT de gobierno [9] colaborará con indicaciones para que el sistema resulte práctico y de utilidad en el contexto de detección de campañas de *smishing*, sin embargo, el trabajo debe ser extrapolable, es decir, cualquier organización debe ser apta para aprovechar las capacidades el dispositivo, dado el valor e impacto que este conlleva.

1.3.1. Investigación

Se debe realizar una investigación sobre *phishing* y *smishing*, con énfasis a este segundo punto. La investigación debe ser mediante la experimentación en distintos sitios utilizando métodos como la divulgación inadvertida de información personal o la interacción con enlaces sospechosos. Se testificarán las acciones tomadas para ser objetivo de fraudes, siguiendo además protocolos de seguridad para garantizar la objetividad de la información y no filtrar información confidencial.

Se efectuará la documentación sobre los indicadores comunes presentes en los sitios y mensajes que resulten maliciosos que incitan al usuario a compartir datos personales o números de teléfono, es decir, sitios que inciten a compartir información que permita recibir campañas de *phishing*. La información recopilada servirá como un recurso para la identificación y el análisis de sitios web fraudulentos durante futuras iteraciones de investigación.

El lector debe ser capaz de comprender como identificar páginas que filtren información y conlleven a la publicación de números telefónicos. Se tiene como fin el identificar patrones y comportamientos comunes en los ataques de *smishing*. La investigación por sí sola aporta un gran valor, pues sirve como un puente para futuras implementaciones de prevención de estafas por mensajes de texto.

El aporte de la investigación para esta memoria es la contextualización sobre el modo de operación de las campañas de *smishing*, dando a conocer medidas de identificación y prevención de este tipo de estafas. La investigación ayudará a la formulación de nuevos mensajes que sirvan para la realización de pruebas del sistema, con el objetivo de comprobar la correcta diferenciación entre estos mensajes y estafas reales.

1.3.2. Implementación

La implementación del sistema *honeypot* consistirá en el diseño y desarrollo de software para una *Raspberry Pi* conectada a distintos módems USB, utilizados para recibir mensajes fraudulentos. El dispositivo funcionará de manera autónoma y debe poder permanecer activo un tiempo definido por el usuario, durante este tiempo, debe estar recibiendo mensajes de texto y reportandolos automáticamente al organismo ejecutor.

Los mensajes de texto recibidos deben ser procesados, identificando el origen del mensaje, su destino, servidor utilizado para hospedar el sitio web, DNS asociados, servicio suplantado y la dirección web fraudulenta en caso de existir. La información anterior debe ser mostrada en una interfaz con formato de *dashboard* en la Raspberry Pi y serán enviados a un servicio externo, el cual obtendrá los datos utilizando *webhooks*.

La implementación será realizada utilizando los siguientes *framework*:

- **React:** Se utilizará React para la configuración de la interfaz.
- **FastAPI:** Se utilizará Python para la interacción con los módems, junto a FastAPI para la comunicación a través de una API REST.

Para garantizar la factibilidad técnica, el proceso de prueba del sistema se hará desarrollando mensajes y sitios similares a campañas de *smishing* anteriores, los cuales serán enviados a uno de los números del *honeypot* tomando las medidas de seguridad necesarias para no afectar a otros dispositivos. Los mensajes capturados por medio de la metodología anterior serán procesados, visitando los enlaces y obteniendo pruebas visuales, posteriormente serán reportados al igual que aquellos obtenidos en un uso real. Las páginas serán creadas en base a lo aprendido en la investigación.

Para eliminar el riesgo de que un usuario de internet pudiese visitar alguno de los sitios de *smishing* creados para probar la funcionalidad de el *Honeypot*, estos jamás se expondrán a la Internet ni se liberará su código, siendo visitables solamente desde el dispositivo Honeypot.

La implementación descrita será "*Open Source*", lo que implica transparencia a la hora de examinar y utilizar el dispositivo, ya que otras personas podrán desplegarlo y sugerir extensiones, además, esto proporciona mayor confianza en la seguridad de la implementación, ya que resulta posible revisar el código por los usuarios. El Centro de Respuesta a Incidentes de Seguridad Informática (CSIRT) de gobierno actuará como usuario de esta primera implementación, permitiendo recibir recomendaciones sobre diseño y ejecución de manera activa.

Visualización gráfica de la implementación

En la figura 1.1, se muestra un diagrama correspondiente al dispositivo a implementar. Se muestra el flujo preliminar que tendrá el dispositivo, además de su interacción con el

remitente de las campañas de *smishing* y el Centro de Respuesta a Incidentes de Seguridad Informática (CSIRT) de gobierno.

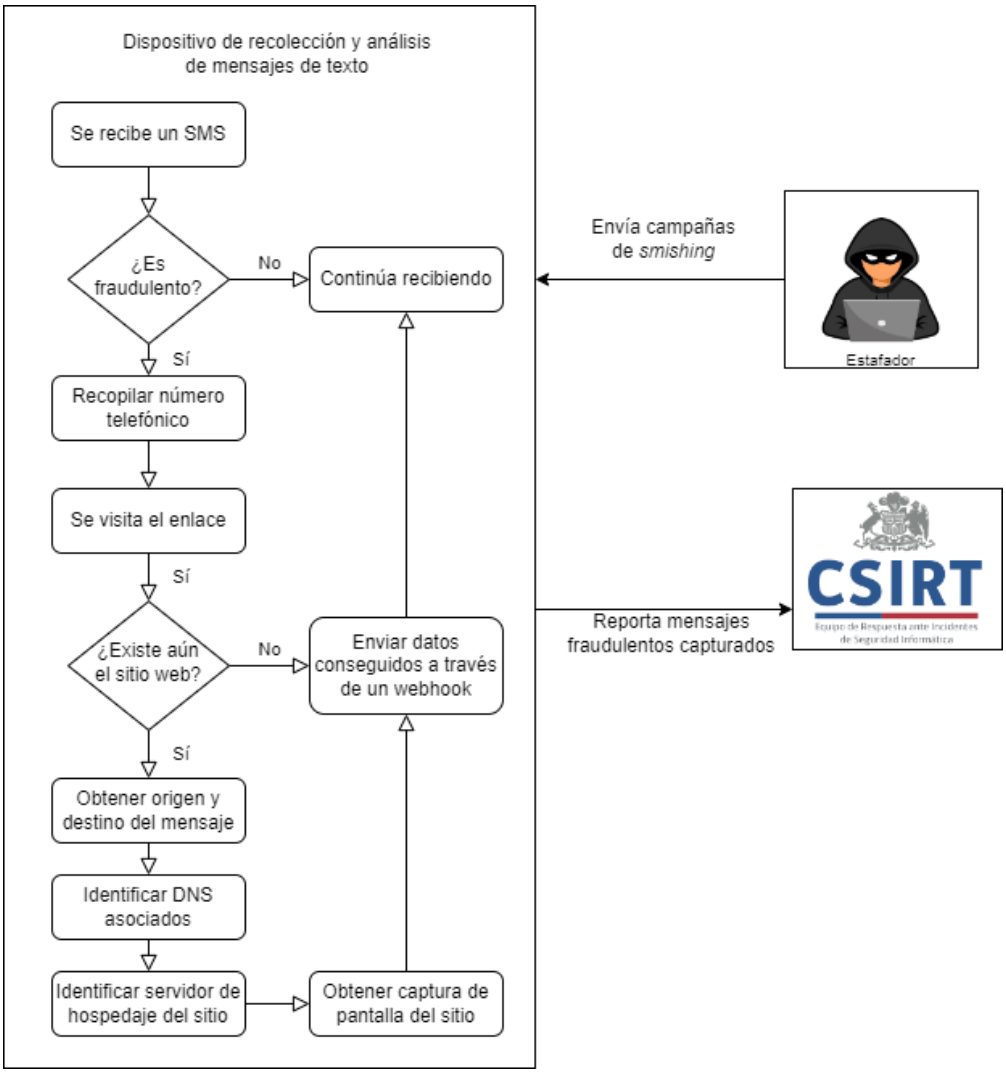


Figura 1.1: Diagrama de flujo del dispositivo y sus interacciones

Capítulo 2

Estado del Arte

2.1. Situación Actual

El panorama actual de la ciberseguridad, según el informe de costos de brechas de seguridad publicado por IBM en 2023 [17], revela que el phishing y el robo de credenciales representan el 16 % y el 15 %, respectivamente, de las filtraciones de datos, consolidándose como los dos vectores de ataque más prevalentes.

En un contexto donde la dependencia de la tecnología móvil es notable, con un total de 7.33 billones de usuarios de teléfonos móviles registrados hasta la fecha [3], resulta alarmante que solo el 35 % de la población esté familiarizada con los ataques de *smishing* [4]. Este fenómeno adquiere mayor relevancia al considerar que los ataques de *smishing* experimentaron un aumento excepcional del 328 % tan solo en el año 2020 [4].

La brecha entre la rapidez con la que evolucionan las tácticas de los ciberdelincuentes y la conciencia del público respecto a estas amenazas destaca la necesidad apremiante de fortalecer las medidas de seguridad y la educación en ciberseguridad para proteger a los usuarios y las organizaciones en un entorno digital cada vez más peligroso.

2.2. Protección de Datos en Chile

En el contexto chileno, la salvaguarda de datos respaldada por la Ley sobre Protección de la Vida Privada (Ley N° 19.628) [22] constituye un pilar fundamental para la privacidad de los ciudadanos. Sin embargo, la situación actual revela la presencia de desafíos, siendo el *smishing* uno de los más prominentes.

A pesar de las regulaciones establecidas por la Ley N° 19.628, el *smishing* representa una amenaza persistente en el entorno digital chileno al afectar directamente la confidencialidad de los datos de los chilenos obtenidos por medios fraudulentos. La legislación, al imponer

restricciones y responsabilidades a las entidades que manejan datos, proporciona un marco legal sólido. No obstante, la realidad actual sugiere que la concienciación pública sobre el *smishing* y otras formas de ataques similares aún es limitada.

2.3. Funcionamiento de los Mensajes de Texto

2.3.1. SMPP

Los mensajes de texto (SMS) utilizan el protocolo de Mensajería Corta de Presentación (SMPP), el cual es una norma crucial en la industria de las telecomunicaciones, diseñada para facilitar la transmisión eficiente de mensajes de texto entre aplicaciones y centros de servicios de mensajes cortos (SMSC). Este informe presenta un resumen simplificado de las principales características de SMPP [30]. El protocolo SMPP resulta ser utilizado por entidades que requieren enviar SMS de manera masiva, no así por el usuario común de telefonía.

El protocolo SMPP implica la interacción entre dos entidades principales:

- Entidad de Envío y Recepción de Mensajes (ESME)
- Centro de Servicios de Mensajes Cortos (SMSC)

La comunicación entre ESME y SMSC se realiza a través de SMPP. La conexión entre la entidad de envío y el centro de servicios se establece mediante TCP/IP, seguida de una sesión SMPP que incluye procesos de autenticación para garantizar la seguridad de la comunicación.

Las operaciones fundamentales de SMPP son:

- Envío de Mensajes
- Recepción de Mensajes

Tras el envío de un mensaje, el SMSC responde con un mensaje de confirmación, proporcionando un identificador único para rastrear el estado del mensaje. Asimismo, se incorporan códigos de error para gestionar problemas potenciales en la entrega de mensajes.

La sesión SMPP concluye con un cierre de conexión. Para salvaguardar la integridad de la comunicación, el protocolo SMPP implementa medidas de seguridad como autenticación mediante nombre de usuario y contraseña, junto con la posibilidad de encriptar la conexión mediante TLS/SSL.

2.3.2. Signalling System No. 7

El Sistema de Señalización No. 7 (SS7) es un protocolo desarrollado para la implementación de comunicaciones entre entidades existentes dentro de una red con el propósito de proporcionar instrucciones. Es una forma estándar y universalmente aceptada de transferir información entre oficinas de conmutación compatibles siguiendo convenciones de intercambio de datos en paquetes. La red SS7 es una red autosanante que maximiza su eficiencia y efectividad mediante el uso de nodos y enlaces redundantes [1]. SS7 es utilizado de manera general dentro de la telefonía móvil. La información transferida a través de la red SS7 puede clasificarse en una de las siguientes categorías:

- Información relacionada con el establecimiento y la liberación de llamadas (procesamiento de llamadas para celular o línea fija).
- Información relacionada con consultas a bases de datos (consultas de bases de datos como validaciones de tarjetas de crédito, traducciones de números 800, etc.).
- Información utilizada para mantener la integridad de la red SS7 (gestión y mantenimiento de la red).

Los mensajes de SS7 pueden ser transmitidos directamente a través de redes IP, o el equivalente funcional de un mensaje de control SS7 puede ser enviado como mensajes de control, por ejemplo, mensajes basados en texto (SMS), directamente entre elementos conectados a una red de datos [1].

En resumen, los mensajes SS7 pueden adaptarse y ser transportados tanto directamente sobre redes IP como a través de formatos equivalentes funcionalmente, como mensajes de texto, ofreciendo opciones flexibles para la comunicación de señalización entre elementos conectados a redes de datos.

2.4. Detección de *Smishing*

La preferencia de los atacantes por el uso de mensajes de texto (SMS) como medio de comunicación es debido a su mayor tasa de respuesta en comparación con los correos electrónicos [5]. Esta elección se traduce en una opción más rentable para los adversarios, quienes pueden enviar volúmenes significativos de mensajes de texto para interactuar con sus víctimas a un costo relativamente bajo [11] en comparación con su contraparte en forma de correo electrónico. Frente a esta realidad, se han implementado enfoques de detección, como el uso de listas negras (*blacklist*), que comparan enlaces presentes en los mensajes con aquellos previamente identificados y catalogados como *phishing*, marcando como *spam* los enlaces detectados en base a iteraciones y reportes previos.

En el ámbito de la detección y clasificación de mensajes, Roy et al. proponen un detector de *smishing* que utiliza técnicas de *Deep Learning*, incluyendo algoritmos como "Naive Bayes" y "Random Forest" [26]. Aunque su implementación se centra en la diferenciación entre mensajes de *spam* y mensajes legítimos, el modelo "Multilayer Perceptron" propuesto por Nandita et al. logra una exactitud notable del 98.8 % [27]. La combinación de estos enfoques contribuye a la formulación del detector DSmishSMS-A [23]. La efectividad de DSmishSMS-A se presenta en una comparación con otros detectores, como se detalla en el trabajo de Sandhya Mishra y Devpriya Soni [23].

Techniques and details	Feature based	Rule based	SmiDCA	Smishing Detector	S-detector	DSmish-A
Search engine domain matching	NO	NO	NO	NO	NO	YES
Source code domain matching	NO	NO	NO	YES	NO	YES
Existence of URL	YES	YES	YES	YES	YES	YES
Existence of phone number and email id in the message	YES	YES	YES	YES	NO	YES
Smishing keywords	YES	YES	YES	YES	YES	YES
Misspelled words	NO	NO	YES	NO	NO	YES
Leet words	NO	NO	NO	NO	NO	YES
Symbols	YES	YES	NO	NO	NO	YES
Special characters	NO	NO	YES	NO	NO	YES

Tabla 2.1: Comparación de técnicas de detección de *Smishing* [23]

El sistema de DSmish-A, presenta la ventaja por sobre otros detectores al poder detectar palabras mal escritas, procesar palabras con símbolos e identificar carácteres especiales. Lo realmente útil para el presente trabajo, corresponde al análisis de las URL mediante la exploración del sitio y sus componentes, como el código fuente y sus firmas. El flujo de la aplicación DSmish es representado en la Figura 2.1.

2.5. Reportes de fraudes

En la actualidad, existen plataformas para informar sobre fraudes que operan tanto a nivel nacional como de manera independiente. Un ejemplo de esta última categoría es la plataforma de informes de Google [16], que se presenta como un formulario sencillo donde solo se requiere la URL de la página de *phishing*, la verificación de la identidad como humano a través de Captcha y, opcionalmente, la entrega de detalles adicionales. Resulta notable ver que, estas herramientas resultan utilizadas de manera general para URL fraudulentas, sin contemplar otra información de identificación como números telefónicos o correos electrónicos.

Otros países también emplean páginas para informar sobre fraudes vinculadas a sus respectivos gobiernos. Estados Unidos mantiene activa la prevención de fraudes, pues su sitio de gobierno USA Gov[35] permite ubicar donde realizar reportes dependiendo del tipo

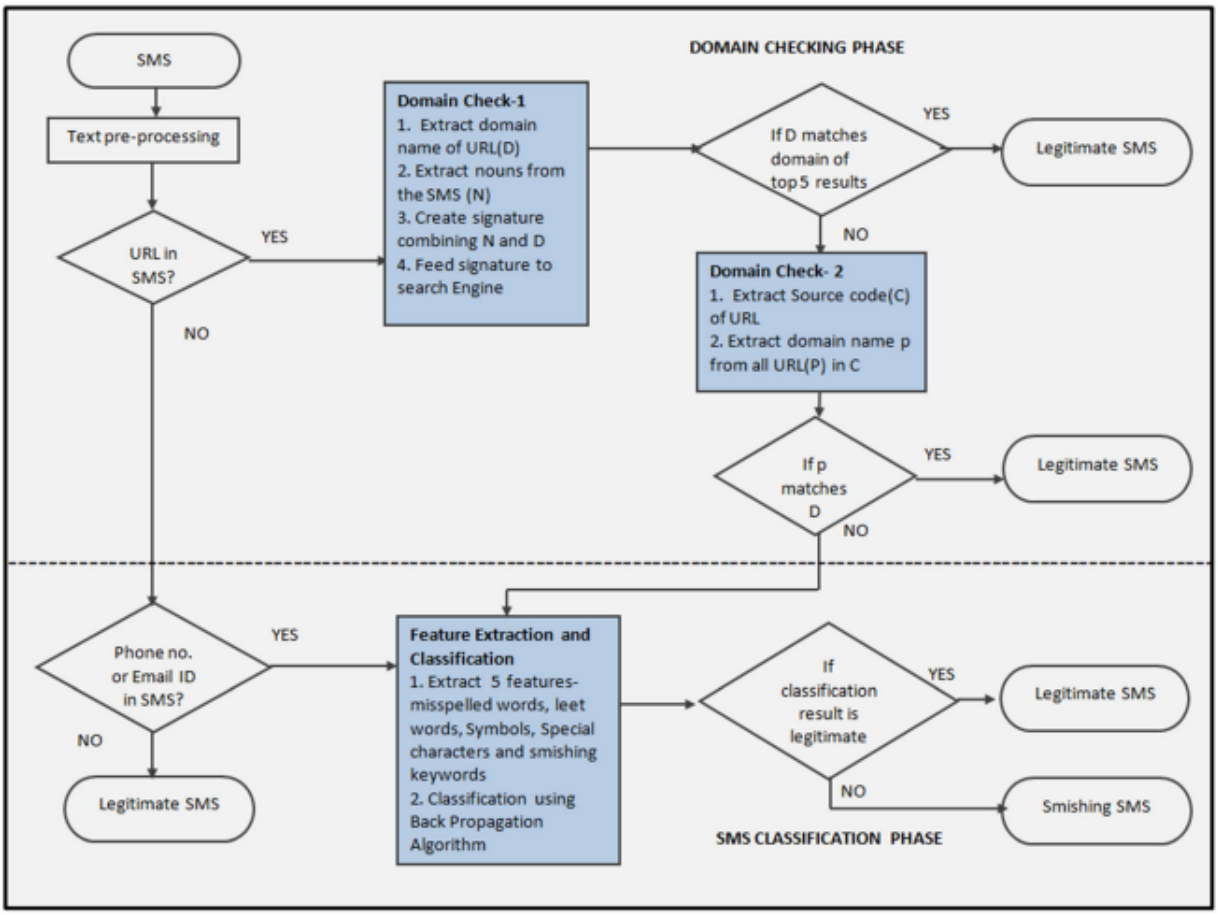


Figura 2.1: Arquitectura de DSmish-A

de fraude, a una entidad específica. Existen además entidades que publican casos de phishing correspondientes a la suplantación de un servicio ofrecido, lo cual permite mantener seguridad por medio de la difusión.

En Chile, es posible ver que bancos tales como BancoEstado, Banco de Chile y BCI presentan campañas públicas de prevención de *phishing*, informando a los usuarios sobre maneras para identificar y mantener su seguridad, sin embargo, no mantienen sitios de reportes para que los usuarios puedan reportar una estafa. Resulta ser crucial para este tipo de entidades velar por la seguridad de sus usuarios.

Para realizar reportes de fraudes en Chile, esto debe realizarse directamente con el CSIRT de Gobierno [9], que utiliza un formulario detallado centrado en los incidentes experimentados. Este formulario incluye información sobre el informante, la entidad afectada y los detalles específicos del incidente. Es relevante destacar que el CSIRT del Gobierno cuenta con una guía de notificación de incidentes que describe la identificación de niveles de peligrosidad e impacto, que varían desde “sin impacto” hasta “impacto crítico” [7].

2.6. Lectura de Mensajes utilizando Modems USB

Para el acceso a la información de los módems desde un sistema Linux como el utilizado por una Raspberry Pi, resulta ser eficiente el uso de Modem Manager.

Modem Manager es una interfaz gráfica basada en GTK diseñada para administrar módems de banda ancha compatibles con diversas tecnologías de comunicación. La herramienta facilita el control del módem a través de canales como USB, RS232 y Bluetooth, así como mediante la gestión de protocolos como AT, QCDM, QMI y MBIM. Aunque no proporciona funciones de acceso telefónico para la conexión a Internet, Modem Manager ofrece una interfaz sencilla para realizar operaciones útiles, como el envío de SMS, la visualización de información del dispositivo y estadísticas de tráfico móvil [10].

A pesar de la existencia de la interfaz gráfica proporcionada por ModemManager, resulta necesario el acceso a los mensajes por medio de CLI (Interfaz de línea de comandos). ModemManager cuenta con paquetes para el sistema operativo Linux, los cuales facilitan acceso a la información de módems.

2.7. Perpetradores del *Smishing* y sus Motivos

Choi, K. y Kim, M. han realizado un estudio sobre el perfil de los estafadores culpables de las estafas de *smishing*, donde se destaca la influencia extranjera y el carecimiento *de culpa de los atacantes* [6].A partir de la extrapolación del estudio titulado *"A study on the modus operandi of smishing crime for public safety"* realizado por Choi y Kim, resulta posible perfilar a los posibles atacantes, resultando en una variedad de actores malintencionados.

2.7.1. Ciberdelincuentes Comunes

Los ciberdelincuentes individuales son los actores más frecuentes en las campañas de *smishing*. Sus principales motivaciones pueden incluir:

- **Obtención de Información Personal:** Robar información personal y financiera, como números de tarjetas de crédito, credenciales de cuentas bancarias y números de seguridad social, que pueden ser utilizados para cometer fraudes.
- **Ganancias Económicas:** Vender la información robada en mercados negros o utilizarla directamente para realizar transacciones fraudulentas. En el propio estudio realizado, se menciona que el precio por dato es de aproximadamente 0.001 dólares, por lo cual se infiere que estos son vendidos en grandes cantidades.
- **Acceso No Autorizado:** Comprometer cuentas personales o empresariales para realizar actividades ilícitas.

2.7.2. Grupos Organizados de Cibercrimen

Existen grupos organizados que realizan *smishing* de manera sistemática y a gran escala. Estos grupos pueden estar motivados por:

- **Fraude a Gran Escala:** Ejecutar esquemas de fraude masivo que pueden afectar a miles de personas y generar grandes sumas de dinero.
- **Espionaje y Sabotaje:** Obtener información sensible de empresas o gobiernos para propósitos de espionaje o sabotaje.
- **Financiamiento de Actividades Ilícitas:** Utilizar los fondos obtenidos a través del *smishing* para financiar otras actividades ilegales, como el tráfico de drogas o el terrorismo.

2.7.3. Motivos Comunes Detrás del *Smishing*

Independientemente del tipo de perpetrador, existen varios motivos comunes que impulsan las campañas de *smishing*:

- **Facilidad y Eficacia:** La simplicidad de enviar mensajes de texto y la alta tasa de respuesta de las víctimas hacen del smishing una táctica efectiva [18].
- **Bajo Costo:** Realizar campañas de *smishing* requiere una inversión mínima en comparación con otros métodos de ciberataque [36].
- **Anonimato:** Los perpetradores pueden ocultar su identidad y ubicación fácilmente, lo que dificulta su rastreo y enjuiciamiento [19].
- **Escalabilidad:** Las campañas de *smishing* pueden ser escaladas rápidamente para dirigirse a un gran número de víctimas, de la misma forma que pueden escalarse campañas de *marketing* [28, 29].

Entender quiénes son los perpetradores del *smishing* y sus motivaciones es crucial para desarrollar estrategias efectivas de prevención y mitigación, protegiendo así a los usuarios y sus datos de posibles ataques.

Capítulo 3

Investigación

La investigación presentada a continuación se basa en los mensajes obtenidos tanto por el CLCERT (Laboratorio de Criptografía Aplicada y Ciberseguridad de la Universidad de Chile) como por el CSIRT de gobierno. Esta investigación puede extenderse mediante el uso continuo del dispositivo y la observación de la captura de mensajes a largo plazo.

El propósito principal de la investigación es entender el funcionamiento de los mensajes de *smishing* y comprender la mentalidad de los atacantes al realizar estafas.

3.1. Análisis de Mensajes de *Smishing*

Entre los mensajes de *smishing* obtenidos gracias a la colaboración con las entidades antes mencionadas, se han identificado patrones comunes.

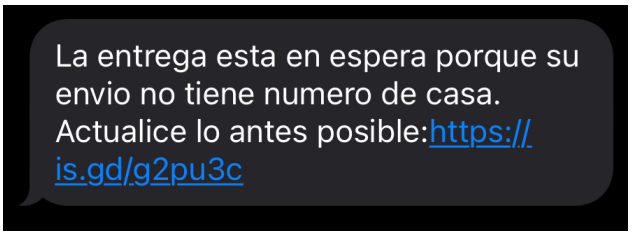


Figura 3.1: Mensaje Fraudulento

En los mensajes de las figuras 3.1 y 3.2, se puede observar que ambos presentan una URL acortada que redirige al sitio objetivo de la estafa. Cabe destacar que algunos mensajes fueron detectados y reportados por servicios de protección contra *spam* en teléfonos móviles.

Estos mensajes de *smishing* imitan servicios de mensajería y entrega de paquetes, presentando urgencia con frases como “Actualice lo antes posible” y “confirme sus datos o su artículo será devuelto”. Esto sugiere el uso de técnicas de ingeniería social para manipu-

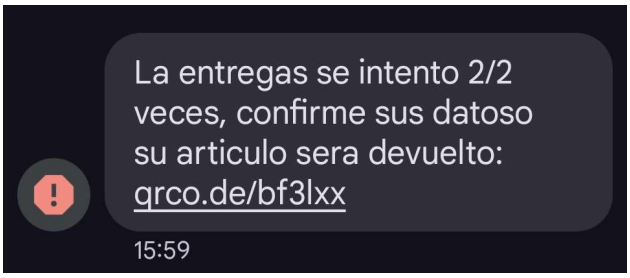


Figura 3.2: Mensaje Fraudulento con Urgencia

lar a los receptores, incitándolos a tomar decisiones apresuradas e ingresar su información personal.

Aunque muchos mensajes incluyen una URL, a menudo acortada, hay casos en los que intentan estafar sin incluir una URL. Estos mensajes, al igual que aquellos con URL, presentan un sentido de urgencia para incitar al destinatario a actuar rápidamente.

Figura 3.3: Mensaje Fraudulento sin URL

Otro patrón común en los mensajes de *smishing* son las faltas de ortografía. Como se observa en las figuras 3.2 y 3.3, estos errores gramaticales pueden ser intencionales para detectar a las personas más vulnerables a las estafas [33].

En la figura 3.3 se puede identificar un intento de suplantación de identidad a la organización "Correos de Chile", lo cual es común en otros mensajes estudiados.

Los mensajes en las figuras 3.4 y 3.5 intentan suplantar a Banco Ripley y BancoEstado, respectivamente. Ambos presentan URLs que simulan ser de los sitios oficiales de estos servicios, pero utilizan subdominios no comunes y dominios de nivel superior como `.info` en lugar de `.cl`, que es el utilizado por las páginas legítimas de Banco Ripley y BancoEstado.

Al verificar estos sitios en la herramienta `crt.sh`, se constató que no están registrados entre los certificados de las páginas legítimas `www.bancoripley.cl` y `www.bancoestado.cl`.

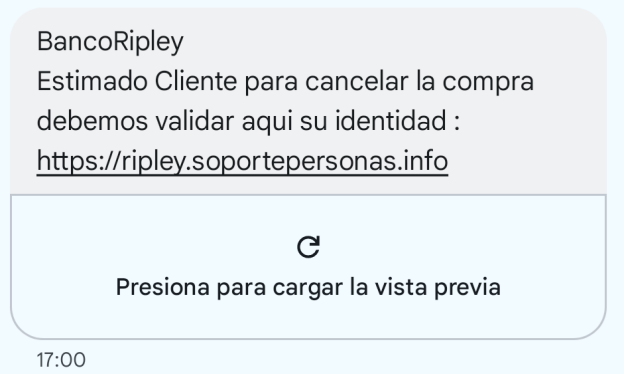


Figura 3.4: Mensaje Fraudulento suplantando Banco Ripley

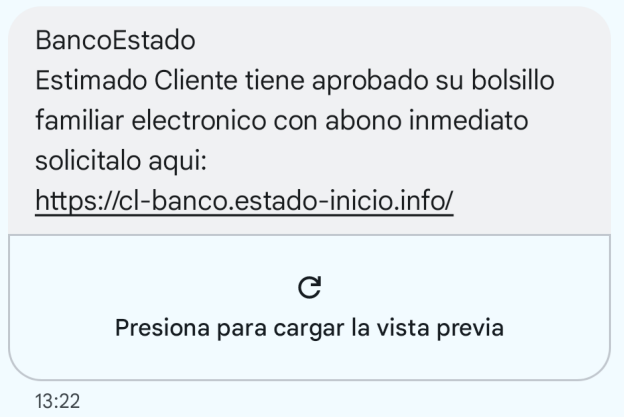


Figura 3.5: Mensaje Fraudulento suplantando BancoEstado

Al verificar estos sitios en la herramienta `crt.sh`, se constató que no están registrados entre los certificados de las páginas legítimas `www.bancoripley.cl` y `www.bancoestado.cl`. *crt.sh* es una herramienta que permite buscar y visualizar los certificados SSL/TLS emitidos para un dominio específico. Los certificados son emitidos por Autoridades Certificadoras (CA) y son esenciales para asegurar que un sitio web es auténtico y confiable. Si un sitio no aparece en los resultados de *crt.sh*, es una señal de alerta, ya que sugiere que el sitio no tiene un certificado válido emitido para el dominio en cuestión, lo que podría indicar que se trata de un sitio fraudulento o suplantado. En este caso, la ausencia de registros en *crt.sh* para los sitios en cuestión sugiere que no son legítimos, reforzando la hipótesis de que podrían estar involucrados en actividades fraudulentas.

3.1.1. Mensajes Capturados con Números no Públicos

Durante el desarrollo de esta investigación, los números utilizados no fueron publicados, sin embargo, se logró obtener mensajes de texto de otros números. La mayoría de los mensajes provienen del proveedor, pero también hay mensajes de números no identificados, que suelen corresponder a números utilizados por la empresa del servicio. Estos números, presu-

miblemente, son dados de baja y reciclados para fines de marketing por parte de los servicios móviles correspondientes.

Además de estos mensajes, se capturaron otros como el siguiente, aun sin exponer los números públicamente:

```
JOSE: Recuerda que tienes un compromiso de pago en Tarjeta
   FASHIONS PARK por el monto de $101.612. Paga en tiendas o
   aqui http://fshp.cl/iILhI
```

Este mensaje incluye una URL acortada y utiliza el protocolo HTTP, advirtiendo al usuario sobre la falta de seguridad (HTTPS). La URL proporcionada no está registrada entre los dominios legítimos de *Fashions Park* según `crt.sh` y al acceder a ella se encontró un error 503, indicando que el servicio no estaba disponible.

Este hallazgo sugiere que incluso sin publicar el número telefónico, es posible recibir intentos de *smishing*, lo que demuestra la vulnerabilidad a fraudes sin haber expuesto datos personales.

3.1.2. Sitios Web

Los sitios web actúan de manera diferente cuando se trata de usuarios de teléfonos móviles, presentando diversos factores que facilitan los ataques de *smishing*. Uno de estos factores es la visibilidad de los certificados de seguridad. En una interfaz móvil, los usuarios tienen una menor visibilidad de los certificados de conexión cuando se trata de HTTP en lugar de HTTPS. Esto permite a los atacantes realizar fraudes con mayor facilidad utilizando páginas fraudulentas con conexiones no seguras.

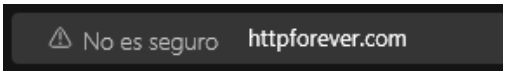


Figura 3.6: URL en navegador de escritorio

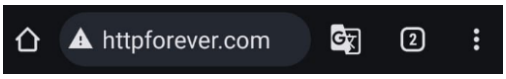


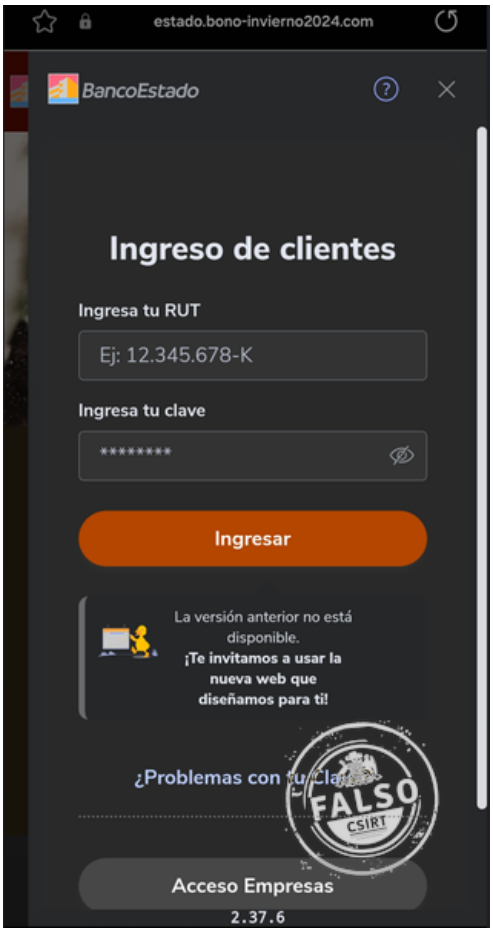
Figura 3.7: URL en navegador de teléfono móvil

Como se puede observar en las figuras 3.6 y 3.7, la conexión se muestra de manera diferente: en la primera figura, que corresponde a un navegador de escritorio, se indica claramente que la conexión no es segura, mientras que en la segunda figura, correspondiente a un navegador móvil, sólo se muestra un símbolo pequeño que indica el tipo de conexión.

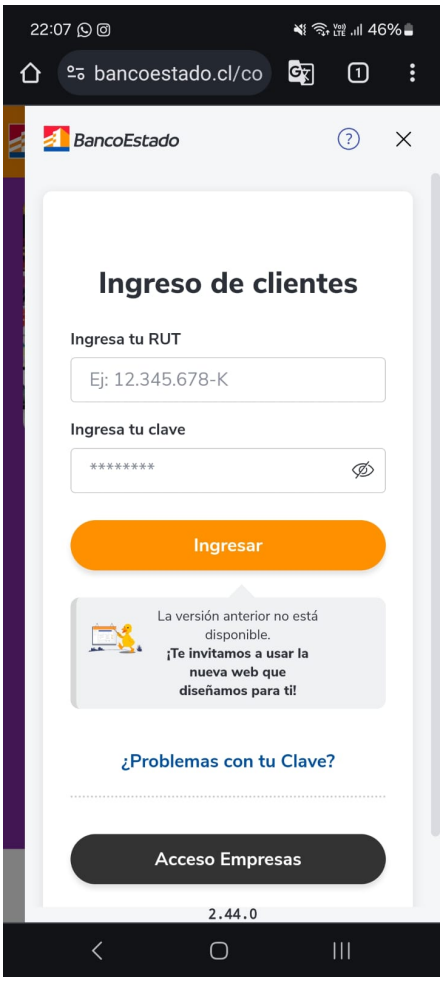
Los sitios de *smishing* también presentan diferentes resultados dependiendo de si se visualizan en un teléfono o en un computador. Se ha encontrado que los atacantes utilizan

recurrentemente sitios web que sólo son visibles desde el celular, redirigiendo a sitios legítimos cuando se accede desde un computador. Además, en los dispositivos móviles, el tamaño de la pantalla limita la visibilidad completa de la URL, lo que proporciona una ventaja a los perpetradores al utilizar URLs fraudulentas menos visibles.

Los sitios web fraudulentos intentan parecerse a los sitios legítimos para aumentar la persuasión y parecer auténticos a las víctimas de este tipo de estafa. Es posible observar esta similitud en las figuras 3.8a y 3.8b. A menudo, estos sitios fraudulentos incluyen hipervínculos a páginas legítimas en sus componentes, lo que puede confundir a los usuarios y provocar una falsa sensación de legitimidad.



(a) Sitio Fraudulento de BancoEstado reportado por CSIRT



(b) Sitio Legítimo de BancoEstado

Figura 3.8: Comparación entre un sitio fraudulento y un sitio legítimo de BancoEstado

3.2. Utilización de Números para Estudio de Estafas

La publicación y uso de números de manera controlada es una herramienta útil para identificar a los responsables de la distribución de datos mediante venta, filtraciones u otros medios que comprometan la confidencialidad de los números telefónicos.

3.2.1. Objetivos Futuros

Basado en lo anterior, los objetivos de distribuir los números en investigaciones futuras incluyen:

- **Segregación según servicio afectado por *smishing*:** Identificar qué servicios están más propensos a sufrir ataques de *smishing*.
- **Identificación de páginas utilizadas para recopilación de datos:** Publicar números en foros públicos para identificar cuáles son utilizados por estafadores para recopilar información.
- **Entender metodologías para recolección de datos:** Comprender cómo los estafadores recolectan datos, identificando si son extraídos mediante explotación de vulnerabilidades, sacados de fuentes públicas o recolectados por *bots*.

3.2.2. Metodologías para filtración

Las metodologías propuestas a continuación sirven como una potencial base para investigaciones futuras, permitiendo perfilar a los atacantes que realicen ataques relacionados con ingeniería social.

Creación de Cuentas

Para investigar la posibilidad de que ciertos sitios web estén filtrando datos, se pueden crear cuentas con números conectados al sistema. Si se reciben mensajes de *smishing*, se podría confirmar la exposición del número. Sin embargo, es necesario recibir mensajes de manera consistente para asegurar que el número ha sido filtrado.

Publicación de Números en Páginas Públicas

Se pueden publicar múltiples números en foros y medios de distribución de información como Github. Si varios números reciben estafas, se puede confirmar que provienen de la misma fuente.

Perfiles en Páginas Web

Para detectar recolecciones de datos por *bots* o procesos automatizados, se pueden crear perfiles con los números del sistema. Esto ayudaría a identificar sistemas automatizados de recolección de datos en redes sociales a partir de la información de contacto.

3.2.3. Factibilidad de la Investigación en el Tiempo

El tiempo requerido para llevar a cabo esta investigación es un factor impredecible, ya que la recolección de mensajes varía en función de cómo y dónde se utilicen los números como señuelos. Esto implica que el estudio se debe realizar a largo plazo.

Para acelerar la investigación mediante la experimentación y atraer mensajes de *smishing*, es necesario contar con extensores USB para el dispositivo utilizado, así como adquirir múltiples módems y tarjetas SIM. Aumentar la capacidad de recepción de mensajes incrementará el flujo de datos, lo que permitirá recibir más mensajes de *smishing*. Esto dependerá de la cantidad de módems y números conectados al dispositivo y de la eficacia en la filtración de números.

A partir del estudio, se pueden obtener resultados estadísticos como:

- Porcentaje de servicios afectados por *smishing* en un periodo de tiempo.
- Fechas con mayor cantidad de estafas realizadas.
- Tasa de crecimiento de delitos informáticos mediante SMS.
- Métodos más comunes para la filtración de datos.

En conclusión, los resultados serán factibles en periodos largos, estimados en al menos un año a partir de la filtración inicial de los números telefónicos, utilizando las metodologías mencionadas en la sección anterior.

Capítulo 4

Implementación

A continuación, se detalla el trabajo realizado, exponiendo las configuraciones en la Raspberry Pi y la estructura del proyecto.

4.1. Componentes Utilizados

En el desarrollo se emplearon los siguientes componentes:

- Raspberry Pi Zero 2 W.
- 4 módems USB Huawei.
- Chips Prepago: Entel, WOM, Claro, Movistar.
- Adaptador MicroUSB a USB 2.0.
- Almacenamiento de 16GB MicroSD.
- Fuente de alimentación con conector MicroUSB.

4.2. Instalación de Sistema Operativo

Para la instalación del sistema operativo, se utilizó una tarjeta de 16GB MicroSD en la que se creó una imagen de disco con Raspberry Pi OS Lite (32-bit), sin interfaz gráfica.

Se habilitó la conexión SSH con acceso por contraseña y conexión a la red local. Para el desarrollo del proyecto, se utilizará este acceso remoto a través de Visual Studio Code.

4.3. Instalación de Paquetes

Para el desarrollo del proyecto, fueron instalados los paquetes mínimos necesarios para el correcto desempeño del sistema. Los paquetes instalados son:

1. **ModemManager:** Permitirá el acceso a los módems y exploración de sus funcionalidades.

2. **Git:** Permitirá el versionamiento del sistema.

3. **pip:** Utilizado como gestor de paquetes para el backend.

4. **Node.js:** Utilizado para desarrollo de frontend.

5. **npm:** Utilizado como gestor de paquetes para el frontend.

4.4. Inicialización de Proyecto

En la fase de inicialización del proyecto que integra React.js y FastAPI. Se instalaron las herramientas esenciales, como FastAPI, uvicorn para el backend, y React con react-scripts para el frontend. Se configuraron adecuadamente los entornos de desarrollo para garantizar una ejecución eficiente, y se crearon los archivos principales tanto para el backend como para el frontend.

4.5. Acceso a Modems

Para acceder a los modems, se empleó la herramienta ModemManager. Se utilizó para identificar modems conectados y acceder a su información. Entre la información destacada, se puede monitorear el estado del sistema y obtener detalles para la identificación del dispositivo.

Con el exitoso uso de ModemManager, se confirma que se puede obtener el acceso a datos cruciales para el estudio, como el número de teléfono, texto recibido, SMSC y hora de los mensajes.

4.6. Planificación de Estructura de Solución

4.6.1. Diagrama de Estructura de Solución

En el contexto de nuestro sistema, la Raspberry Pi actúa como el nodo central para la recepción de mensajes. Se configura la Raspberry Pi para interceptar y recibir mensajes provenientes de módems USB conectados.

Una vez que los mensajes son recibidos por la Raspberry Pi, se inicia el procesamiento de datos. Este proceso implica en primer lugar, la información de los módems capturadores, la extracción del mensaje y su procesamiento, obteniendo información clave de los mensajes, como números de teléfono, contenido del mensaje, y otros metadatos relevantes. A continuación, se inicia la verificación de la URL contenida en el mensaje en caso de existir, para luego clasificar el mensaje como legítimo o fraudulento. La información extraída se organiza y formatea adecuadamente para su posterior envío y presentación.

La entidad que usará el sistema, rol que protagonizará el CSIRT de Gobierno durante el desarollo del proyecto, se encargará de supervisar y gestionar la información recopilada, recibe los datos procesados en formato JSON. Esta estructura de datos facilita la transmisión eficiente de la información, garantizando una integridad y coherencia de los datos.

Se implementa una interfaz de usuario utilizando React para que los usuarios puedan acceder y visualizar la información de manera intuitiva. Los datos procesados y almacenados en la Raspberry Pi están disponibles para los usuarios a través de esta interfaz, sirviendo para el monitoreo y la gestión de la información.

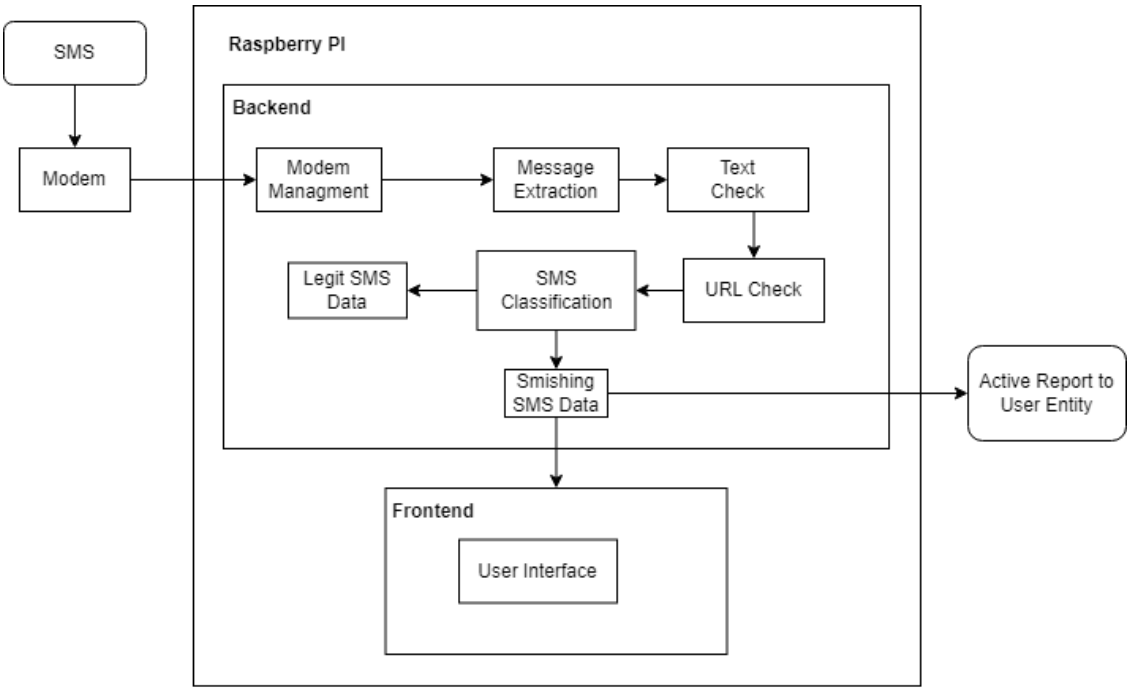


Figura 4.1: Estructura de Solución

4.6.2. Formato JSON

El JSON resultante debe encapsular la información crítica derivada del procesamiento de mensajes SMS en el entorno de detección de smishing. Se incluyen los detalles específicos del mensaje SMS de cada distinto `message_id`, identificador definido como la fecha de captura concateda con el *hash* del texto del mensaje, los detalles incluídos son el número de teléfono, el texto del mensaje, el tipo PDU, la marca temporal, la URL, el título de la página web, la captura de pantalla codificada en base64, y la información de la página con sus redirecciones. En `page_info_with_redirects`, se proporciona información adicional sobre la página, donde cara URL incluye los servidores DNS asociados y sus certificados, ofreciendo así una visión completa de la amenaza detectada.

A continuación, se presenta un ejemplo de archivo JSON.

```
{
    "message_id": {
        "number": "+569XXXXXXXX",
        "text": "Texto del mensaje",
        "pdu_type": "pdu type (ej: deliver)",
        "timestamp": "YYYY-MM-DDTHH:MM:SSZ",
        "url": "https://example.com",
        "title": "Titulo de la pagina web",
        "screenshot": "screenshot_base_64",
```

```
        "urls": ["url_1", "url_2", "url_n"],
        "ip": "192.168.1.1",
        "certificate": "BASE64 certificate"
    },
    ...
}
```

4.7. Limitaciones de Equipamiento

Inicialmente, para el desarrollo del proyecto, se empleó una Raspberry Pi del modelo *Raspberry Pi Zero 2 W*, cuyas especificaciones son las siguientes:

- CPU Arm Cortex-A53 de cuatro núcleos y 64 bits a 1 GHz
- SDRAM de 512 MB
- LAN inalámbrica de 2,4 GHz 802.11 b/g/n
- Alimentación a través de micro USB

Con módems de las siguientes características, como se detalla en las especificaciones de Huawei [21]:

- Compatible con HSPA+/HSDPA/HSUPA/UMTS en bandas duales
- Capacidad GSM/GPRS/EDGE (850/900/1800/1900MHz)

Según las especificaciones técnicas y recomendaciones, cada módem USB, considerando el estándar HSPA+, requiere aproximadamente 0.5A de corriente [24, 12]. La fuente de alimentación utilizada proporciona al equipo una corriente de 2A.

Se ha observado un problema de rendimiento en la Raspberry al utilizar Visual Studio Code para acceder a ella de manera remota. Los procesos en ejecución consumen la memoria de acceso aleatorio del dispositivo, y además, se ha detectado que la memoria swap se encuentra completamente utilizada.

Cuando se conectan 2 módems, el dispositivo experimenta errores significativos, afectando su funcionalidad y estabilidad. Esta limitación impide el uso efectivo de múltiples módems para ampliar la capacidad de conexión del dispositivo, resultando en fallos en la disponibilidad, ya que se provoca una detención total del dispositivo.

Debido a estas limitaciones, se decide cambiar a un nuevo dispositivo.

El nuevo dispositivo a utilizar para el desarrollo del proyecto corresponde a una *Raspberry Pi 5* con 16 GB de memoria RAM, lo cual ayuda a prevenir el uso desmedido de memoria swap. Al utilizar este dispositivo, es posible efectuar procesamientos de mayor rendimiento sin afectar la disponibilidad, pues cuenta con un procesador más potente, además, *Raspberry Pi 5* permite una mayor conectividad al contar con 4 puertos USB sin necesitar extensores.

Alternativamente, es posible desarrollar la presente memoria con un dispositivo Linux de características similares o mejores, en cuyo caso facilitaría la alimentación de una mayor cantidad de módems USB.

4.8. Obtención de Datos

4.8.1. Mensajes de Texto Fraudulentos

Los datos utilizados para el posterior entrenamiento del modelo evaluativo fueron provistos por el CSIRT de gobierno y el CLCERT (Laboratorio de Criptografía Aplicada de la Universidad de Chile). El CSIRT, al momento de realizar la memoria, contaba solamente con informes generados de manera manual en formato **PDF** a partir de los reportes realizados en su página web, los cuales contenían una imagen del mensaje fraudulento. Los datos provistos por el CLCERT estaban en formato de imágenes de mensajes.

Fue necesario procesar los reportes entregados por el CSIRT para extraer datos de entrenamiento.

Filtro de Reportes

Debido a que los reportes estaban mezclados entre *phishing* y *smishing*, fue necesario distinguir entre los 909 reportes entregados por el CSIRT.

Se realizó un filtro a partir de la lectura automatizada de los informes, identificando palabras clave para determinar el método de estafa. Se encontró que cada reporte que correspondía al método de estafa mediante correos electrónicos contenía la palabra "correo" dentro del texto, la cual fue utilizada para realizar el filtro.

Obtención de Imágenes de Mensajes de Texto

Luego de identificar los reportes correspondientes a la estafa del tipo *smishing*, fue necesario descargar la imagen con el mensaje de texto del PDF entregado. El reporte, sin embargo, presentaba más imágenes además de la captura de pantalla con el mensaje. Se observó que la captura de pantalla correspondía a la imagen cuyo almacenamiento requería una mayor

cantidad de bits, con lo cual fue posible identificar la imagen correspondiente al mensaje. Para esto, se utilizó la librería `fitz`.

Obtención del Texto del Mensaje

Al estar los mensajes en formato de imágenes, fue necesario extraer el texto. Esto fue posible mediante el procesamiento con la herramienta `textract`. Esta herramienta permitía obtener la totalidad del texto, por lo cual fue necesario realizar una limpieza de estos datos, eliminando horas del mensaje y números de teléfono en caso de aparecer. Este proceso se utilizó tanto para las imágenes extraídas de los informes como para las imágenes provistas por el CLCERT.

4.8.2. Mensajes de Texto Legítimos

Los mensajes legítimos fueron extraídos de un teléfono móvil de uso cotidiano.

Debido a la facilidad de uso de mensajes de texto entregada por Google, fue posible acceder a los mensajes mediante la conexión a través del navegador y vinculación con la cuenta utilizada en el dispositivo. Con lo anterior, fue posible extraer los mensajes directamente como texto plano, realizando limpieza según hora y número de mensaje.

4.8.3. Datos obtenidos para Entrenamiento de Modelo de Clasificación

Se obtuvieron 79 mensajes fraudulentos a partir de los reportes entregados por el CSIRT de gobierno. Esta cifra pudo ser corroborada posteriormente por la entidad a cargo. Además, se obtuvieron 29 mensajes a partir de las imágenes con mensajes fraudulentos entregadas por el CLCERT, totalizando 108 mensajes fraudulentos. Por otro lado, se extrajeron 170 mensajes legítimos a partir de un teléfono móvil.

4.9. Base de Datos

Al utilizar una Raspberry Pi como dispositivo para realizar el procesamiento de los datos obtenidos, se decide utilizar una base de datos local. Una vez enviados los datos con SMS capturados e identificados como *smishing*, el cliente, receptor de los datos, es el encargado de almacenar los reportes realizados.

El almacenamiento local es realizado utilizando SQLite, una biblioteca de software que implementa un motor de base de datos SQL, autónomo, sin servidor y sin configuración.

SQLite es adecuado para este propósito debido a su simplicidad y eficiencia en dispositivos con recursos limitados como la Raspberry Pi.

Para este proyecto, se utilizará solo una tabla que guardará la totalidad de los mensajes captados por los módems que mantengan conexión con el dispositivo. Los campos de la tabla incluyen:

- `id`: Llave primaria de tipo string. Identificador único generado por la concatenación de la fecha de recepción del mensaje y un hash del texto en él. Esto asegura que cada mensaje tenga un identificador único y facilita su búsqueda y gestión.
- `type`: Campo del tipo string que indica el tipo del mensaje, el cual puede ser "smish" en caso de ser fraudulento o "legit" en caso de ser legítimo.
- `path`: Campo del tipo string que indica la dirección en el módem correspondiente para el cual se mantiene almacenado este mensaje. Este campo ayuda a localizar físicamente dónde está almacenado cada mensaje dentro del hardware.
- `number`: Campo del tipo string que contiene el número telefónico remitente del mensaje. Este campo es esencial para identificar la fuente del mensaje y potencialmente obtener su origen.
- `pdu_type`: Campo del tipo string que indica el tipo de PDU (Protocol Data Unit) del mensaje.
- `state`: Campo del tipo string, corresponde al estado actual del mensaje dentro del módem. Este campo puede tener valores como received", read", "deleted", etc., y es útil para gestionar el flujo de los mensajes.
- `storage`: Campo del tipo string, indica cómo se está almacenando el mensaje dentro del módem. Por ejemplo, puede indicar si el mensaje está almacenado en la memoria interna del módem o en una tarjeta SIM.
- `smsc`: Campo del tipo string que contiene la dirección del Centro de Servicios de Mensajes Cortos (SMS Center) que ha gestionado el envío del mensaje. Este campo puede ser útil para fines de análisis y trazabilidad.
- `timestamp`: Campo del tipo Date, indica la fecha y hora de recepción del mensaje. Este campo es importante para el acceso e identificación según orden cronológico.

4.10. Backend

El **Backend** es el encargado del procesamiento de los datos capturados en la base de datos y consiste en múltiples endpoints utilizados para la comunicación con el lado del cliente. Su objetivo principal es ser el centro de operaciones para la obtención de información en los módems USB.

4.10.1. Utilidades

Parser

Debido al formato de los mensajes, es necesario transformarlos en un objeto iterable para poder acceder a su información. El **parser** se encarga de dar formato a los mensajes recibidos en forma de diccionario, extrayendo los campos necesarios para clasificar el mensaje y almacenarlo en la base de datos. Para lograr esto, se utilizan expresiones regulares que permiten identificar los campos a partir del resultado de la ejecución de ModemManager.

Obtención de Mensajes

Se utiliza el módulo ModemManager (mmcli) para manejar y acceder a la información de los módems. La utilidad de obtención de mensajes ofrece operaciones para la interacción con módems, tales como:

- **Información del módem:** Obtiene toda la información relacionada con la identificación del dispositivo USB conectado, permitiendo su correcta gestión y monitoreo.
- **Listado de mensajes:** Obtiene una lista con las direcciones de almacenamiento de los mensajes dentro de los módems, facilitando el acceso a cada mensaje almacenado.
- **Información de mensaje:** Obtiene la información de un mensaje específico a partir de un identificador (ID), permitiendo su análisis y clasificación detallada.

Normalización de Objetos DateTime

Los mensajes pueden tener diferentes zonas horarias. Para mantener un orden cronológico efectivo, es necesario normalizar los objetos de tiempo según la zona horaria del receptor del mensaje. Se establece UTC como la zona horaria predeterminada para el procesamiento posterior, asegurando coherencia en el manejo de tiempos.

Obtención de Información de Página

Esta utilidad se utiliza para obtener información de una página web, configurada para funcionar como un teléfono móvil, específicamente el modelo Galaxy S5. Se utilizan librerías como *socket* y *playwright* con Chromium. La información proporcionada por este módulo incluye:

- **Captura de pantalla:** Captura de pantalla del sitio final, después de redirecciones, para documentar visualmente el contenido de la página.

- **URLs visitadas:** Lista de todas las URLs visitadas mediante redirecciones, para rastrear el flujo de navegación.
- **DNS:** Obtiene la información del host de la URL especificada, permitiendo verificar la autenticidad y origen del sitio.
- **Certificados:** Obtiene el certificado de la página visitada, asegurando la legitimidad de la conexión.

Interacción con Base de Datos

Se utilizan funciones para la interacción con la base de datos a través de SQLAlchemy. Las operaciones realizadas incluyen:

- **Obtención de todos los datos:** Recupera todos los datos de la base de datos, permitiendo un acceso completo a la información almacenada.
- **Filtrado de datos según valor:** Recupera todos los datos con un valor específico en un campo determinado. Principalmente utilizado para filtrar por tipo, como obtener solo los mensajes de tipo "smish", facilitando la segmentación de datos.
- **Filtrado de datos según fecha:** Recupera todos los datos a partir de una fecha indicada en formato UTC, permitiendo el análisis temporal de la información.

4.11. Endpoints

Actualización de información (*/update*)

Endpoint encargado de la actualización de información en los módems. Al ejecutar este endpoint, se utiliza el clasificador para obtener el tipo de mensaje (*smish* o *legit*), luego se actualiza la información de los módems dentro de la base de datos con nuevos mensajes capturados. Esto asegura que la base de datos esté siempre actualizada con la última información recibida.

Dashboard (*/dashboard*)

Endpoint encargado de actualizar la información para métricas, guardando esto en formato *JSON* para facilitar la comunicación con el lado del cliente. La información guardada corresponde a:

- `smishing_count`: Corresponde a la cantidad total de *smishings* recibidos, proporcionando un indicador del volumen de amenazas detectadas.

- `legitimate_count`: Corresponde a la cantidad total de mensajes legítimos recibidos, ofreciendo una visión del tráfico normal.
- `connected_modems`: Corresponde al número de módems conectados, permitiendo monitorear la infraestructura en uso.
- `messages_per_phone_number`: Corresponde a la cantidad de mensajes recibidos por cada número de teléfono. Sólo guarda aquellos con mayor cantidad de números que la media, destacando los números más activos.
- `smishings_per_phone_number`: Corresponde a la cantidad de *smishings* recibidos por cada número de teléfono. Sólo guarda aquellos con mayor cantidad de números que la media, identificando las principales fuentes de amenazas.
- `total_messages`: Corresponde a la cantidad total de mensajes recibidos, dando una idea del volumen de mensajes manejados.
- `daily_messages`: Corresponde a la cantidad de mensajes recibidos en la última semana, permitiendo un seguimiento a corto plazo.
- `monthly_messages`: Corresponde a la cantidad de mensajes recibidos en los últimos 12 meses, proporcionando una visión a mediano plazo.
- `yearly_messages`: Corresponde a la cantidad de mensajes recibidos en los últimos 5 años, ofreciendo un análisis a largo plazo.
- `daily_smishings`: Corresponde a la cantidad de *smishings* recibidos en la última semana, ayudando a identificar tendencias recientes de amenazas.
- `monthly_smishings`: Corresponde a la cantidad de *smishings* recibidos en los últimos 12 meses, permitiendo el análisis de amenazas a mediano plazo.
- `yearly_smishings`: Corresponde a la cantidad de *smishings* recibidos en los últimos 5 años, ofreciendo un contexto histórico de las amenazas.

Envío de Información a Usuario (*/send_message*)

Este endpoint es el encargado de enviar la información al usuario receptor en formato JSON. Al ejecutarlo, se procesan los mensajes obtenidos durante una cantidad de tiempo en segundos especificada como parámetro, y se obtiene su información utilizando las utilidades de obtención de información de página. El *JSON* enviado contiene el siguiente formato:

- `number`: El número de teléfono desde el cual se envió el mensaje, identificando la fuente del mensaje.
- `text`: El contenido textual del mensaje recibido, proporcionando el mensaje completo.
- `pdu_type`: El tipo de PDU (Protocolo de Datos de Usuario) del mensaje, que indica el formato en que se ha codificado el mensaje, esencial para su correcta interpretación.

- `timestamp`: La marca temporal del mensaje, indicando la fecha y hora exactas en que se recibió, en el formato AAAA-MM-DD HH:MM en zona UTC, asegurando la trazabilidad temporal.
- `url`: La URL extraída del mensaje, si es que el mensaje contiene un enlace, permitiendo la evaluación de posibles amenazas web.
- `title`: El título de la página web correspondiente a la URL extraída del mensaje, proporcionando contexto adicional sobre el contenido web.
- `screenshot`: Una captura de pantalla de la página web correspondiente a la URL extraída del mensaje, mostrando cómo se ve la página después de posibles redirecciones, útil para análisis visual.
- `page_info_with_redirects`: Información detallada de la página web, incluyendo todas las redirecciones y el DNS del host de la URL especificada, ofreciendo un análisis del sitio web a partir de las redirecciones.

4.12. Frontend

El Frontend de este proyecto se desarrolla empleando `React.js`. La información necesaria para la visualización se adquiere a través del endpoint `/dashboard`.

La interfaz de usuario permite la visualización de métricas mediante una variedad de gráficos, incluyendo gráficos de barras, líneas y circulares. La interacción entre la aplicación y el cliente receptor de la información, se realiza de manera automática, sin necesidad de interacción. Además, se presenta el JSON de forma general para una comprensión más completa en caso de requerirlo.

La información se actualiza de manera continua, refrescándose cada minuto para asegurar que la información mostrada es reciente.

El procesamiento de información se realiza en el *backend*, es decir, el `json` mostrado en el portal del usuario sólo muestra la información utilizada para graficar.

4.13. Desarrollo de un Modelo para la Detección de Mensajes Fraudulentos

Se opta por entrenar el modelo utilizando el método *Random Forest*, ya que es adecuado para la clasificación de mensajes debido a su robustez y capacidad de manejo de datos complejos , esto en comparación con métodos como *Decision Tree*[15]. Dada la naturaleza variada y a menudo ruidosa de los mensajes, la capacidad de *Random Forest* para mane-

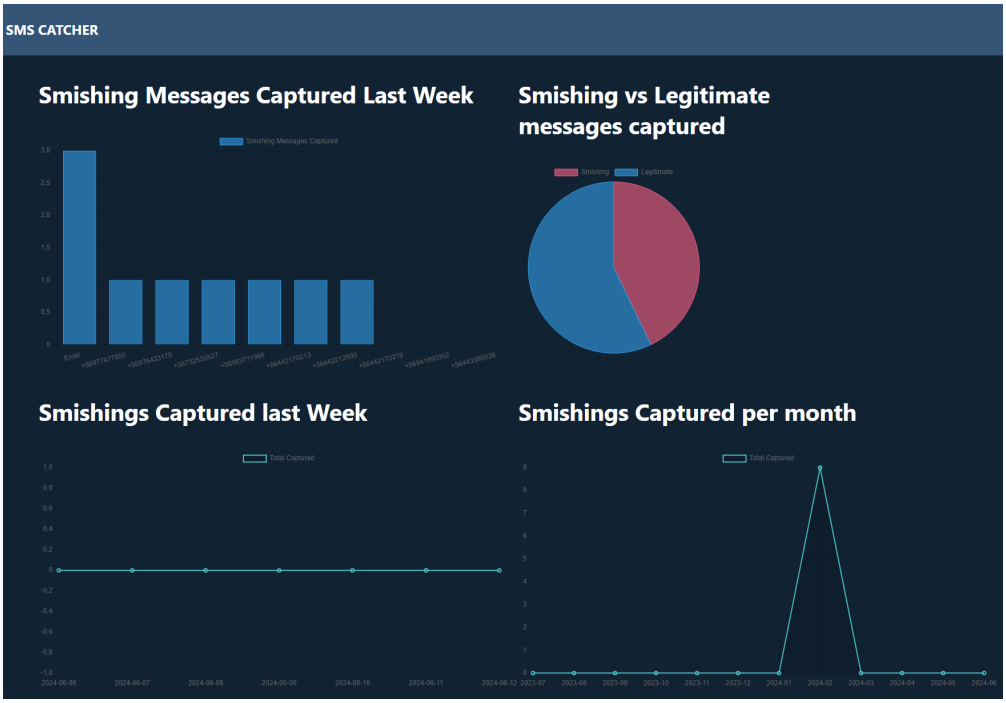


Figura 4.2: Vista de Métricas en portal de Usuario

jar eficazmente estas condiciones lo convierte en una elección ideal para la clasificación de mensajes fraudulentos.

Debido a que la cantidad de datos utilizados es relativamente pequeña, limitandose a la cantidad de mensajes obtenidos en base a la colaboración con el CSIRT de Gobierno y el CLCERT, se prefiere el uso de *Random Forest* frente a otros métodos como *Multinomial Naive Bayes* [2].

Cabe destacar qué, mediante la utilización de *Random Forest*, se permite el aprendizaje continuo del aplicativo, ya que el modelo puede ser fácilmente actualizado con nuevos datos conforme estos se van recopilando. Esto no solo mejora la precisión del modelo con el tiempo, sino que también lo hace más adaptable a nuevas tácticas de fraude que puedan surgir. Además, *Random Forest* facilita la identificación de las características más relevantes en la detección de mensajes fraudulentos, lo que contribuye a optimizar y enfocar los esfuerzos de detección en las señales más significativas.

Uno de los puntos claves a tener en cuenta en la clasificación del texto, resultan ser factores que tienen en común los mensajes utilizados para su entrenamiento, como lo son la presencia de URL's dentro de estos además de las faltas ortográficas recurrentes en mensajes de estafa.

El proceso incluye las siguientes etapas:

1. **Preprocesamiento del Texto:** Se depura el texto de cada mensaje realizando tareas como la conversión del texto a minúsculas, la eliminación de caracteres especiales, la

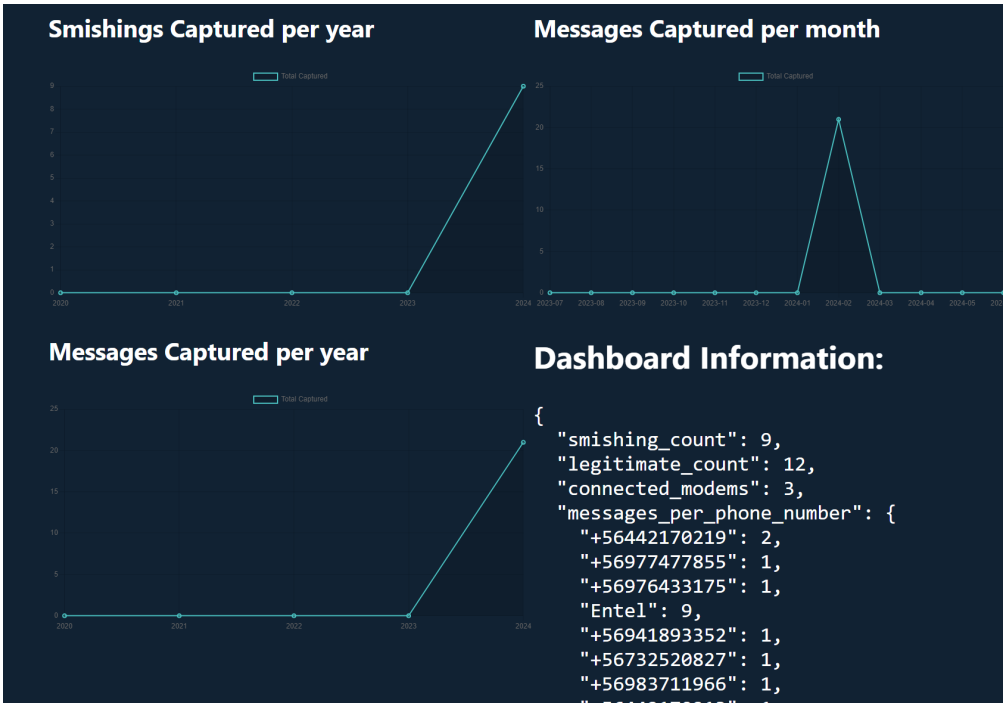


Figura 4.3: Vista de Gráficos en Portal de Usuario

supresión de espacios adicionales y la exclusión de palabras vacías tanto en inglés como en español.

2. **Carga del Conjunto de Datos:** El conjunto de datos se importó desde un archivo CSV utilizando la biblioteca `pandas`. Este archivo CSV comprendió mensajes auténticos junto con una compilación de mensajes fraudulentos extraídos de informes previos generados por el CSIRT de gobierno.

3. **Preprocesamiento de los Datos:** Se aplicó el proceso de depuración de texto a todos los mensajes del conjunto de datos, y los resultados se almacenaron en una nueva columna.

4. **Inicialización del Modelo:** Se instanció un modelo de clasificación de *Random Forest* utilizando la clase `RandomForestClassifier` de `sklearn.ensemble`.

5. **Creación del Pipeline:** Se configuró un pipeline mediante la clase `Pipeline` de `sklearn.pipeline`, que constó de tres etapas: vectorización, transformación TFIDF y el modelo de *Random Forest*.

6. **Entrenamiento del Modelo:** El conjunto de datos se dividió en conjuntos de entrenamiento y prueba utilizando la librería `scikit-learn`.

7. **Guardado y Carga del Modelo:** Finalmente, el modelo entrenado se almacenó en disco utilizando la biblioteca `joblib` para su uso futuro. Asimismo, se implementó la carga del modelo desde el disco para realizar predicciones.

4.14. Ejecución Automática

Al estar utilizando un dispositivo con sistema operativo Linux, se utiliza un *cronjob* que periodicamente ejecuta la acción de envío de información a la entidad reguladora. Para tener datos más precisos enviados, el sistema es ejecutado cada un minuto, con el fin de reportar los mensajes lo más pronto posible a medida que sean detectados, y poder identificar si distintas campañas de *smishing* llegaron al mismo tiempo.

Capítulo 5

Despliegue

5.1. Instalación de Imagen

En esta sección se explica cómo instalar una imagen de una Raspberry Pi con el proyecto realizado.

5.1.1. Requisitos

Para llevar a cabo este proceso, se necesitan los siguientes elementos:

- Una computadora con lector de tarjetas SD.
- Tarjeta SD para utilizar en una Raspberry Pi.
- La imagen de Raspberry Pi descargada.
- Software para escribir la imagen en la tarjeta SD.

5.1.2. Descargar la Imagen

Primero, es necesario descargar la imagen de la Raspberry Pi desde hackerlab.cl.

5.1.3. Escribir la Imagen a la Tarjeta SD

A continuación, se detallan los pasos para escribir la imagen a la tarjeta SD en diferentes sistemas operativos. Para este proceso, es necesario reemplazar `/path/to/imagen_raspberry_pi.img` y `/dev/sdX` con la ruta correcta de la imagen y el nombre del dispositivo.

En Windows

1. Insertar la tarjeta SD en el lector de tarjetas de la computadora.
2. Descargar e instalar *Win32 Disk Imager* desde sourceforge.net.
3. Abrir *Win32 Disk Imager*.
4. Seleccionar la unidad correspondiente a la tarjeta SD.
5. Seleccionar la imagen descargada haciendo clic en el ícono de la carpeta y buscando el archivo `imagen_raspberry_pi.img`.
6. Hacer clic en *Write* para escribir la imagen en la tarjeta SD.

En macOS

1. Insertar la tarjeta SD en el lector de tarjetas de la computadora.
2. Abrir la *Terminal*.
3. Encontrar el nombre del dispositivo de la tarjeta SD ejecutando el siguiente comando:

```
diskutil list
```

4. Desmontar la tarjeta SD ejecutando:

```
diskutil unmountDisk /dev/diskX
```

   donde `/dev/diskX` es el nombre del dispositivo de la tarjeta SD.

5. Escribir la imagen a la tarjeta SD utilizando `dd`:

```
sudo dd if=/path/to/imagen_raspberry_pi.img of=/dev/rdiskX bs=1m
```

En Linux

1. Insertar la tarjeta SD en el lector de tarjetas de la computadora.
2. Abrir una terminal.
3. Encontrar el nombre del dispositivo de la tarjeta SD ejecutando:

```
lsblk
```

4. Desmontar la tarjeta SD ejecutando:

```
sudo umount /dev/sdX*
```

   donde `/dev/sdX` es el nombre del dispositivo de la tarjeta SD.

5. Escribir la imagen a la tarjeta SD utilizando `dd`:

```
sudo dd if=/path/to/imagen_raspberry_pi.img of=/dev/sdX
bs=1M
```

5.2. Operación del Sistema

Activación de Cronjobs

Es necesario activar cron para monitorear la llegada de nuevos mensajes. Para eso, se debe ejecutar el comando:

```
sudo service cron start
```

Para dejar de obtener nuevos mensajes utilizar el comando `sudo service cron stop`

```
sudo service cron stop
```

Creación de archivo .env

Dentro de la carpeta /sms-catcher/backend se debe modificar un archivo de nombre `.config` ubicado en la dirección `boot/.config`, en el cual se debe guardar la dirección de la API receptora de mensajes de la siguiente forma y opcionalmente un `API TOKEN`. Los datos en este archivo posteriormente serán copiados y utilizados como variables de entorno para la ejecución.

```
POST_API_URL = "my_api_url"
API_TOKEN = "my_api_token"
```

Donde `my_api_url` debe reemplazarse por la dirección de la API.

Ejecución del Sistema

El sistema comienza a operar de manera autónoma al realizar un reinicio con:

```
sudo reboot
```

Para desplegar el *frontend*, es necesario ejecutar el siguiente comando:

```
cd sms-catcher/frontend/sms-catcher
serve -s build -l 3000
```

Lo anterior nos sitúa dentro de la carpeta del proyecto en *React* e inicia la interfaz de usuario, mostrando las gráficas.

5.2.1. Uso Real para Investigación

Al estar la memoria en un estado funcional, es posible realizar la públicación de números telefónicos asociados a los módems en redes de tráfico ilícito de datos o bien, fuentes de datos de uso común posiblemente usadas por ciberdelincuentes, con el fin de identificar a los actores de este tipo de fraude.

El tiempo en tener resultados reales resulta ser impredecible y dependiente de la cantidad de filtraciones, números y estándares utilizados para la públicación de números telefónicos a modo de anzuelo para delincuentes. Debido a lo anterior, una estimación posible es la de obtener resultados tangibles en más de un año.

Capítulo 6

Resultados

Los resultados de la evaluación del sistema automatizado para recolección y análisis de mensajes de texto fraudulentos son esenciales para determinar su efectividad al cumplir con los objetivos.

6.1. Resultados de evaluación del Modelo

Se llevó a cabo una evaluación del modelo utilizando los datos de prueba. Se calcularon métricas como la precisión, el recall y la puntuación F1, y se generó un informe de clasificación.

	Precisión	Recall	Puntuación F1	Soporte
Legítimo	1.00	0.97	0.99	40
Fraudulento	0.97	1.00	0.98	30
Exactitud			0.99	70
Promedio macro	0.98	0.99	0.99	70
Promedio ponderado	0.99	0.99	0.99	70

Las métricas presentadas permiten evaluar el rendimiento del modelo en diferentes aspectos:

Precisión: Refleja la proporción de mensajes clasificados como fraudulentos que realmente lo son, lo que indica la capacidad del modelo para minimizar los falsos positivos.

Recall: Indica la capacidad del modelo para identificar correctamente los mensajes fraudulentos, es decir, mide la efectividad en la detección de fraudes.

Puntuación F1: Proporciona un equilibrio entre precisión y recall, siendo una métrica clave cuando se busca mantener un buen desempeño en ambas áreas. Esta resulta ser la

métrica clave para la evaluación de los resultados obtenidos por el modelo clasificador.

Soporte: Representa el número total de instancias evaluadas en cada tipo de mensaje, en este caso legítimo y fraudulento, ayudando a entender la distribución de los datos de prueba.

Exactitud: Mide el porcentaje total de predicciones correctas, proporcionando una visión general del rendimiento del modelo.

Promedio Macro y Promedio Ponderado: El promedio macro da igual peso a cada clase, mientras que el promedio ponderado ajusta las métricas según la proporción de instancias de cada clase.

Se puede ver la efectividad del uso del método *Random Forest* debido a los altos puntajes obtenidos en el informe de clasificación. Al probar el funcionamiento de manera manual, es posible observar que la posibilidad de clasificar mensajes fraudulentos como legítimos es escasa; sin embargo, la clasificación de mensajes legítimos como fraudulentos resulta pasar ocasionalmente. Esto no afecta la funcionalidad y aplicación del sistema, ya que el usuario es el encargado de verificar la veracidad de los datos.

6.2. Métricas Mostradas en el Dashboard

El sistema mostró métricas actualizadas en tiempo real, reflejando el análisis de los mensajes recibidos. Los siguientes datos corresponden a aquellos obtenidos mediante el uso de 3 módems en el periodo de Noviembre 2023 hasta Junio 2024, se identificaron 9 mensajes de *smishing*, lo que indica la cantidad de amenazas detectadas por el sistema. En contraste, se recibieron 12 mensajes legítimos, lo que proporciona una visión del tráfico normal de mensajes.

El sistema también monitoreó que hay 3 módems USB conectados y en uso para la recolección de mensajes, coincidiendo con la cantidad físicamente contectada al dispositivo. Lo anterior permite verificar la operatividad del sistema.

El total de mensajes recibidos fue de 21. Se detalla la cantidad de mensajes recibidos por cada número de teléfono. Se identificó que de los mensajes obtenidos, “Entel” envió 9 mensajes, correspondiendo a la mayoría. Esta información ayuda a identificar los números con mayor actividad de mensajes.

En cuanto a los mensajes de *smishing*, se especificó que “Entel” envió 3 mensajes de *smishing*, lo cual resulta en falsos positivos qué, sin embargo, cumplen con el formato de los datos utilizados para entrenar el modelo. Otros números como +56977477855 y +56976433175 enviaron 1 mensaje cada uno. Esta métrica es útil para identificar las principales fuentes de amenazas.

A nivel mensual, se indicó que se recibieron 21 mensajes en febrero de 2024, mientras

que no se recibieron mensajes en otros meses del último año. Esta información es útil para analizar la actividad de mensajes a mediano plazo y detectar posibles patrones o tendencias.

Finalmente, todos los mensajes recibidos (21 en total) fueron en 2024, proporcionando una perspectiva a largo plazo sobre la actividad de mensajes. De manera similar, las métricas mostraron información específica sobre los mensajes de *smishing* recibidos diariamente, mensualmente y anualmente. En particular, se recibió un total de 9 mensajes de *smishing* en febrero de 2024, lo que resalta un período de alta actividad en términos de amenazas detectadas.

Capítulo 7

Conclusiones

7.1. Recuento de Objetivos Alcanzados y no Alcanzados

7.1.1. Objetivo General

El objetivo general de diseñar e implementar un sistema de captura de mensajes de texto capaz de identificar intentos de *Smishing* fue alcanzado con éxito. El sistema proporciona la usabilidad requerida por el CSIRT de gobierno, y es puesto en marcha en la organización.

7.1.2. Objetivos Específicos

1. **Investigar sobre acciones y conductas que atraen ataques de *smishing*:** No se ha cumplido el objetivo debido a que el tiempo requerido para obtener estafas está fuera del control del usuario.

2. **Documentar cómo operar con módems USB de telefonía celular para enviar y recibir SMS:** Se ha cumplido de manera parcial, logrando el acceso a los mensajes de manera automática.

3. **Conectar módems USB con un servidor, al cuál se deben enviar los datos de manera automática:** Se ha logrado enviar los datos cada cierto intérvalo de tiempo, predefinido como 1 minuto.

4. **Investigar y analizar el comportamiento de los mensajes de *smishing*, identificando patrones y tácticas comunes en ataques de *smishing*:** Se ha logrado el objetivo mediante el estudio de los mensajes capturados previamente por el CSIRT de gobierno.

5. **Implementar métodos de recolección de datos sobre los intentos de *smishing*:** Objetivo logrado mediante la recolección de datos tanto en la información de los mensajes, como en las páginas web fraudulentas si aplica.

6. **Realización de pruebas del sistema utilizando el reenvío campañas de *smishing* previamente capturadas por el CSIRT:** No fue necesario debido a que la cantidad de mensajes obtenidos fue suficiente para realizar pruebas.

7.2. Reflexiones

7.2.1. Gestión del Proyecto y Proceso de Desarrollo

La planificación inicial y los diseños preliminares fueron elementos cruciales en el desarrollo del proyecto. El tiempo dedicado al estudio y la adaptación a tecnologías como Raspberry Pi y módems USB para el acceso a tarjetas SIM fue esencial. Esta preparación permitió asegurar la continuidad del proyecto y mitigar problemas potenciales a lo largo del desarrollo. No obstante, a pesar de una planificación meticulosa, surgieron desafíos imprevistos que requirieron ajustes durante el proyecto. Estos cambios se realizaron en función de la información disponible en cada momento y las limitaciones de los dispositivos utilizados. La capacidad de adaptación fue fundamental para superar estos obstáculos y mantener el avance constante del proyecto.

7.2.2. Desafíos y Soluciones

Uno de los primeros desafíos fue el uso de nuevas tecnologías, como los computadores de bajo costo y la utilización de módems. En los primeros acercamientos, se consideró el uso de módulos para manejar DBUS, sin embargo, la complejidad aumentaba de manera considerable, por lo que esta solución fue descartada para la presente implementación.

El mayor desafío técnico fue la limitación del equipamiento inicial. La baja capacidad de procesamiento de la Raspberry Pi Zero 2 W hizo necesario cambiar a una Raspberry Pi 5. La baja disponibilidad y los frecuentes fallos del equipamiento inicial afectaron la capacidad de prueba del sistema, obligando a basar el desarrollo en el estado del arte para evitar retrasos en la parte técnica del proyecto. Además, el uso de múltiples tecnologías con diferentes propósitos requirió la integración de microprocesos separados, lo que hizo necesaria la verificación de compatibilidad entre librerías y dispositivos, demandando tiempo para solucionar estos problemas y en su defecto, reemplazar librerías no compatibles con el stack tecnológico.

Para la obtención de los datos provistos por el CSIRT, hubo un aplazamiento debido a la falta de un aplicativo que pudiera capturar los mensajes, lo que obligó a recurrir a la recopilación manual de reportes exportados en PDF. Este problema podría superarse

en trabajos futuros mediante la creación de conjuntos de datos de mensajes en español relacionados con fraudes, lo que facilitaría el entrenamiento de nuevos modelos.

7.2.3. Calidad y Pruebas

La realización de pruebas fue una parte crucial del desarrollo del proyecto. Se creó un servidor local para verificar el correcto envío de datos en el sistema de manera autónoma y previa a la verificación con el usuario final.

La retroalimentación continua fue esencial para asegurar la calidad del proyecto, permitiendo considerar posibles problemas y prevenirlos y mejorar la funcionalidad del sistema.

7.2.4. Lecciones Aprendidas

Una de las lecciones más importantes aprendidas es que numerosos pequeños componentes pueden unirse para formar un sistema grande y funcional. La integración efectiva de diversas tecnologías y la gestión de múltiples procesos demostraron ser esenciales para el éxito del proyecto.

7.2.5. Impacto

El sistema de recolección y análisis de mensajes de texto fraudulentos desarrollado en esta tesis tiene un impacto significativo en el bienestar de las personas en Chile. Al colaborar con el CSIRT de gobierno, este sistema ayuda a proteger a los ciudadanos de fraudes mediante mensajes de texto. Además, el proyecto contribuye a la concientización y educación del público sobre las estafas telefónicas, promoviendo una mayor seguridad y prevención ante estos delitos.

7.3. Trabajo a Futuro

7.3.1. Mejoras en el Modelo de Clasificación

El modelo de clasificación utilizado, entrenado con el Método de Random Forest, presenta con frecuencia falsos positivos, lo cual representa un punto de mejora para futuras iteraciones. El trabajo propuesto en DSmishSMS-A [23] ofrece una implementación teórica más efectiva para la detección de fraudes utilizando inteligencia artificial. Sin embargo, esta solución aún no ha sido probada en un entorno práctico.

Para mejorar la detección automatizada y confiable de fraudes, es crucial el diseño e implementación de tecnologías avanzadas que permitan una identificación más precisa. Además de analizar y proponer puntos de mejora para los algoritmos existentes, es importante integrar técnicas de *deep learning* y análisis de patrones, así como implementar mecanismos de retroalimentación continua que puedan adaptar y refinar los modelos en tiempo real. La unión entre campos disciplinarios como la inteligencia artificial, seguridad informática y ciencias del comportamiento pueden contribuir significativamente a la creación de modelos más robustos. Adicionalmente, la utilización de conjuntos de datos más amplios y diversificados para el entrenamiento puede reducir la incidencia de falsos positivos y mejorar la generalización del modelo.

7.3.2. Mejoras en el Frontend

El Frontend del proyecto solo funciona como visualización de los datos almacenados dentro del sistema, en este caso una Raspberry Pi 5. Es posible crear una interfaz para la interacción y control del comportamiento del sistema, permitiendo realizar tareas como:

- Deshabilitar y habilitar uno o varios puertos, y por consiguiente, manejar el estado de los módems.
- Visualización de información de los módems.
- Filtrar datos y sus visualizaciones.
- Interacción con la base de datos.

Para mejorar la experiencia del usuario y la funcionalidad del sistema, sería beneficioso implementar una interfaz de usuario más intuitiva y responsiva. Incorporar gráficos en tiempo real y herramientas de visualización dinámica puede ayudar a los usuarios a entender mejor los datos y tomar decisiones informadas rápidamente. Además, añadir capacidades de personalización permitirá a los usuarios ajustar la interfaz según sus necesidades específicas, mejorando la eficiencia operativa y la satisfacción del usuario final.

7.3.3. Sistema Centralizado de Reporte de Mensajes Fraudulentos

La ampliación de la captura de mensajes del dispositivo es posible mediante la conexión de múltiples módems a un solo sistema. Sin embargo, esta estrategia enfrenta limitaciones significativas en cuanto al poder de procesamiento necesario para gestionar mensajes de múltiples números simultáneamente. Además, en situaciones donde la recepción de mensajes debe ser aún mayor, la capacidad del sistema puede verse comprometida.

Para abordar estas limitaciones, se propone la implementación de un sistema centralizado de reporte de mensajes fraudulentos. Este sistema permitiría a varios cooperadores acceder

a un mismo *endpoint* para enviar mensajes de fraude, centralizando la recolección de datos y facilitando la gestión eficiente de un mayor volumen de mensajes. La centralización no solo optimiza el uso de recursos, sino que también mejora la capacidad de procesamiento y almacenamiento, asegurando una respuesta más rápida y eficaz ante el creciente volumen de datos.

En este sistema centralizado, los cooperadores, que pueden incluir entidades gubernamentales, proveedores de servicios de telecomunicaciones y otros actores relevantes, enviarían los mensajes fraudulentos detectados a través de un *endpoint* común. Esto permitiría una recolección de datos más uniforme y ordenada, facilitando el análisis y clasificación de los mensajes en un entorno controlado.

La implementación de un sistema centralizado también tendría un impacto significativo en la estructura y funcionamiento de la recolección de datos. Al consolidar los informes de fraude en un solo punto, se simplifica la infraestructura necesaria y se reduce la redundancia de datos. Esto permite una mejor gestión de los recursos y asegura que los datos sean accesibles de manera eficiente para su análisis. Además, al disponer de una base de datos centralizada, se mejora la capacidad de monitoreo y seguimiento de las tendencias de fraude, facilitando una respuesta más coordinada y efectiva ante los incidentes detectados.

7.3.4. Investigaciones pendientes sobre *Smishing*

El sistema logra ser capaz de actuar como punto de recolección de mensajes, lo cual facilita el estudio del fraude mediante mensajes de texto. Resulta posible realizar un *honeypot*, es decir, la creación de un anzuelo para obtener estafas, utilizando la publicación de los números de los módems utilizados en medios públicos con el fin de atraer campañas de mensajes de texto y por sobre todo, estafadores.

A diferencia de los reportes realizados por personas, la creación de estos anzuelos es menos aleatoria y menos dependiente de factores externos, constituyendo una medida controlada para obtener mensajes fraudulentos. Los resultados, en comparación con los reportes, dependen de los componentes utilizados (como puede ser la cantidad de números) y de las acciones efectuadas sobre cada uno para identificar la atracción de estafadores, como la separación de publicaciones de números en distintas páginas según el dispositivo. Dado lo anterior, la mayor ventaja que presenta el uso de *honeypots* es que, al ser algo controlado, permite realizar una caracterización.

El estudio de mensajes capturados mediante la publicación de números resulta ser un limitante en el intervalo de tiempo utilizado para el desarrollo de la presente memoria, sin embargo, resulta ser efectivo a largo plazo, con una estimación de al menos un año para obtener resultados que representen la realidad y den una caracterización acertada.

Los lugares en donde los números de teléfono se hagan públicos deben estar guardados con el fin de identificar de donde se pudo haber filtrado cada número una vez lleguen mensajes fraudulentos, lo cual además de identificar de donde sale la filtración, logra proporcionar

métricas sobre que tipo de páginas o servicios se ven afectados por filtraciones y que tipo de sitios suelen ser más seguros en el cuidado de la información personal.

Bibliografía

[1] A. Affandi and M. Husain. An investigation on standards and applications of signalling system no. 7. 2015.

[2] Analytics Vidhya. Randomforest classifier vs multinomial naive bayes for a multi-output natural language, 2024. Recuperado el 16/07/24 de URL: https://medium.com/analytics-vidhya/randomforest-classifier-vs-multinomial-naive-bayes-for-a-multi-output-natural-language-2426381a5217: :text=While

[3] bankmycell. HOW MANY SMARTPHONES ARE IN THE WORLD? Technical report. Recuperado el 03/12/2023 de: https://www.bankmycell.com/blog/how-many-phones-are-in-the-world.

[4] Ben Martens. 11 Facts + Stats on Smishing (SMS Phishing) in 2023. Technical report. Recuperado el 03/12/2023 de: https://www.safetydetectives.com/blog/what-is-smishing-sms-phishing-facts/.

[5] CallHub. 6 reasons why sms is more effective than email marketing. Technical report, 2020. Recuperado el 14/12/2023 de: https://callhub.io/6-reasonssms-effective-email-marketing/.

[6] K. Choi and M. Kim. A study on the modus operandi of smishing crime for public safety. *Convergence Security Journal*, 16(3_2):3–12, 2016.

[7] CSIRT. GUÍA DE NOTIFICACIÓN DE INCIDENTES PARA ORGANISMOS DE LA ADMINISTRACIÓN PÚBLICA. Technical report. Recuperado el 05/12/2023 de: https://www.csirt.gob.cl/media/2021/10/Guia-de-notificacion-ciberincidentes.pdf.

[8] CSIRT. Nueva campaña de phishing por SMS (smishing) que suplanta a Correos-Chile. Technical report. Recuperado el 06/09/2023 de: https://www.csirt.gob.cl/alertas/8fph23-00842-01/.

[9] CSIRT de Gobierno de Chile. CSIRT de Gobierno de Chile. Technical report. Recuperado el 09/09/2023 de: https://www.csirt.gob.cl.

[10] Darkcrizt. Modem Manager: Una aplicación para la gestión de Modem en Linux. Technical report. Recuperado el 05/12/2023 de: https://blog.desdelinux.net/modem-manager-una-aplicacion-para-la-gestion-de-modem-en-linux/.

[11] Simon J Delany, Mark Buckley, and Derek Greene. Sms spam filtering: methods and data. *Expert Systems with Applications*, 39(10):9899–9908, 2012.

[12] Raspberry Pi Stack Exchange. Looking for 3g usb dongle with low power consumption, 2024. Recuperado el 28/05/2024 de: https://raspberrypi.stackexchange.com/questions/15838/looking-for-3g-usb-dongle-with-low-power-consumption.

[13] FBI's 2021 Internet Crime Complaint Center (IC3). FBI's 2021 Internet Crime Complaint Center (IC3) Report. Technical report, 2021. Recuperado de: https://www.ic3.gov/Media/PDF/AnnualReport/2021_IC3Report.pdf.

[14] FBI's 2022 Internet Crime Complaint Center (IC3). FBI's 2022 Internet Crime Complaint Center (IC3) Report. Technical report, 2022. Recuperado de: https://www.ic3.gov/Media/PDF/AnnualReport/2022_IC3Report.pdf.

[15] GeeksforGeeks. Difference between random forest and decision tree, 2024. Recuperado el 16/07/24 de URL: https://www.geeksforgeeks.org/difference-between-random-forest-and-decision-tree/.

[16] Google. Report Phishing Page. Technical report. Recuperado el 05/12/2023 de: https://safebrowsing.google.com/safebrowsing/report_phish/?hl=en.

[17] IBM. Cost of a Data Breach Report 2023. Technical report. Recuperado el 03/12/2023 de: https://www.ibm.com/reports/data-breach.

[18] IBM. Smishing, 2024. Recuperado el 16/07/24 de URL: https://www.ibm.com/topics/smishing: :text=In

[19] Identity Guard. Smishing meaning, 2024. Recuperado el 16/07/24 de URL: https://www.identityguard.com/news/smishing-meaning.

[20] Internet World Stats. Internet World Stats. Technical report, s/f. Recuperado de: https://www.internetworldstats.com/stats.html.

[21] 4G LTE Mall. Huawei e353 3g umts hspa hsdpa 21mbps usb surf stick, 2024. Recuperado el 28/05/2024 de: https://www.4gltemall.com/huawei-e353-3g-umts-hspa-hsdpa-21mbps-usb-surf-stick.html.

[22] Ministerio Secretaria general de la presidencia. Ley 19628 Sobre Protección de la Vida Privada. Technical report. Recuperado el 03/12/2023 de: https://www.bcn.cl/leychile/navegar?idNorma=141599.

[23] S. Mishra and D. Soni. Dsmishsms-a system to detect smishing sms. *Neural Computing and Applications*, pages 1–18, 2021.

[24] Of Modems and Men. Power requirements, 2024. Recuperado el 28/05/2024 de: https://www.ofmodemsandmen.com/power.html.

[25] Policía de Investigaciones. Ciberdelitos continuaron al alza en 2021. Technical report. Recuperado el 06/09/2023 de: https://www.pdichile.cl/centro-de-prensa/detalle-prensa/2022/01/04/ciberdelitos-continuaron-al-alza-en-2021.

[26] PK Roy, JP Singh, and S Banerjee. Deep learning to filter sms spam. *Future Generation Computer Systems*, 102:524–533, 2020.

[27] RA Sesha, PS Avadhani, and C Nandita. A content-based spam e-mail filtering approach using multilayer perceptron neural networks. *International Journal of Engineering Trends and Technology*, 41:44–45, 2019.

[28] Sinch. Sms vs email marketing, 2024. Recuperado el 16/07/24 de URL: https://www.sinch.com/blog/sms-vs-email-marketing/.

[29] Sinch. What is smishing?, 2024. Recuperado el 16/07/24 de URL: https://www.sinch.com/blog/what-is-smishing/.

[30] SMPP. Smpp protocol: Api to enable sms messaging between applications and mobiles. Technical report, 2023. Recuperado el 14/12/2023 de: https://smpp.org.

[31] Sofía Alvarez. Casos de delitos informáticos aumentaron 61 por ciento durante 2021 y 2022. Technical report. Recuperado el 06/09/2023 de: https://chocale.cl/2023/06/casos-de-delitos-informaticos-aumentaron-un-61-durante-2021-y-2022/.

[32] SSL Insights. Ssl certificates statistics, 2024. Recuperado el 16/07/24 de URL: https://sslinsights.com/ssl-certificates-statistics/: :text=Over

[33] Joseph Steinberg. Why scammers make spelling and grammar mistakes, 2020. Recuperado el 2024-07-16 de https://josephsteinberg.com/why-scammers-make-spelling-and-grammar-mistakes/.

[34] Truecaller. Smart sms, 2024. Recuperado el 16/07/24 de URL: https://www.truecaller.com/es-la/blog/features/smart-sms.

[35] USA Gov. Where to report scams. Recuperado el 05/12/13 de https://www.usa.gov/where-report-scams#block-usagov-content.

[36] WP Funnels. Sms vs email marketing: Cost effectiveness and roi, 2024. Recuperado el 16/07/24 de URL: https://getwpfunnels.com/sms-vs-email-marketing/: :text=Cost