TELEMÁTICA
REVISTA DIGITAL de las Tecnologías de la Información y las Telecomunicaciones
Revista Telemática. Vol. 17. No. 2, mayo-agosto, 2018, p. 60- 70 ISSN 1729-3804

SISTEMA PARA LA DETECCIÓN DE ATAQUES PHISHING UTILIZANDO CORREO ELECTRÓNICO

Antonio Hernández Dominguez [1], Sergey Storchak[2]

[1] Centro de Telemática Facultad 2. Universidad de las Ciencias Informáticas, Carretera a San Antonio de los Baños, Km. 2 ½. Torrens, La Lisa, La Habana, Cuba. CP: 19370, [2] Instituto de Tecnología Informática y Seguridad de la Información, Ingeniería y Academia Tecnológica de la Universidad Federal del Sur, Taganrog, Rusia, 347922, Federación de Rusia, Taganrog, Chekhova str.2
[1]e-mail: ahdominguez@uci.cu

RESUMEN

Los ataques phishing constituyen un mecanismo criminal que utiliza técnicas de ingeniería social y subterfugio técnico para robar a los usuarios datos de identidad personal y credenciales de sus cuentas financieras. Desafortunadamente, ninguna entidad es totalmente inmune a estos ataques, lo que induce a que todas las organizaciones deben implementar un plan ordenado de medidas de prevención de phishing, con el objetivo de reducir los riesgos ante una exposición directa. Generalmente una de las medidas que toman las empresas y que en principio puede resultar bastante eficaz, es la de capacitar, entrenar y preparar a los empleados que utilicen las TIC, con el fin de que estos sean capaces de reconocer oportunamente este tipo de ataques. No obstante, debido a la creciente innovación empleada por los perpetradores y que cada vez utilizan nuevas técnicas, apelar solo a la conciencia no es suficiente. De ahí que el objetivo de la presente investigación sea el desarrollo de un sistema basado en una arquitectura orientada a servicios que integre y correlacione patrones de comportamientos existentes en bases de datos internacionales para la detección de ataques phishing utilizando correo electrónico.

PALABRAS CLAVES: phishing, ingeniería social, ciberataques, email.

SYSTEM FOR THE DETECTION OF PHISHING ATTACKS USING EMAIL

ABSTRACT

Phishing attacks are a criminal mechanisms that uses social engineering techniques and technical subterfuge to steal users' personal identity data and credentials from their financial accounts. Unfortunately, any entity is totally immune to those attacks, which means that all organizations must implement a proper plan of phishing prevention measures, in order to reduce the risks to direct exposure. Generally, one of the measures taken by companies, which in principle can be quite effective, is to train and prepare employees who use ICT, in order that they be able to recognize those type of attacks in a timely manner. However, due to the increasing innovation employed by the perpetrators and who are increasingly using new techniques, appealing only to self-responsibility of users is not enough. Hence, the objective of this research is conducted on developing a system based on a service-oriented architecture to integrate and correlate patterns of common behaviors from international databases for the detection of phishing attacks using e-mail.

KEY WORDS: phishing, social engineering, cyberattacks, email.

1. INTRODUCCIÓN

En la actualidad, con el desarrollo vertiginoso de las Tecnologías de la Información y Comunicación (TIC), se ha manifestado una tendencia hacia el crecimiento del desarrollo de aplicaciones. Paralelo al desarrollo de las TIC surge una de las principales preocupaciones del hombre dado por la seguridad de la información que

transita a través de Internet. Actualmente las tendencias mundiales revelan un crecimiento exponencial de acciones malignas encaminadas a poner en riesgo la integridad de la información. Un ciberataque consiste en cualquier acción tomada para socavar las funciones de una red informática con fines políticos o de seguridad nacional. [1]

Estudios recientes publicados según el Informe de Amenaza de Seguridad de Internet (ISTR, por sus siglas en inglés), emitido por la corporación multinacional estadounidense Symantec Corporation, arrojan que, durante el año 2017, las tácticas más sencillas y los delincuentes informáticos más innovadores consiguen resultados sin precedentes en el panorama de las amenazas mundiales [2]. Los ataques dirigidos que moldean gobiernos, el sabotaje y la manipulación con fines económicos; las organizaciones criminales, enfocadas cada vez más hacia el sector financiero; el *ransomware*, como variante moderna para la extorsión digital y las vulnerabilidades en la nube como próxima frontera de los delitos informáticos, son algunos ejemplos citados por Symantec. Estos evidencian la necesidad de tomar acciones proactivas, que permitan reducir el riesgo ante la exposición directa ante alguna amenaza en específico.

Por otra parte, este estudio hace referencia al uso del correo electrónico como un medio de infección. Según estadísticas, este comportamiento ha ido creciendo hasta llegar a convertirse hoy día en el arma preferida de los "*hackers*". Los fraudes relativos al correo electrónico corporativo estafaron más de 3000 millones de dólares a empresas en los últimos tres años, con ataques diarios a más de 400 compañías [3]. La compañía estadounidense Digital Guardian, utilizando una investigación realizada por un panel de 28 expertos en seguridad de la información, asegura que el 97% de los ataques informáticos no aprovechan una falla en el software, sino que usan técnicas de ingeniería social para conseguir las credenciales necesarias y así mediante estas vulnerar la seguridad informática. De acuerdo a Christopher Hadnagy en su libro "Ingeniería Social el Arte del Hacking Personal" se define a la Ingeniería Social como: "El acto de manipular a una persona para que lleve a cabo una acción que –puede ser o no- lo más conveniente para cumplir con cierto objetivo. Este puede ser la obtención de información, conseguir algún tipo de acceso o lograr que se realice una determinada acción" [4].

Los ataques de ingeniería social más comunes provienen de phishing o *spear phishing* y pueden variar según los eventos actuales, los desastres o la temporada de impuestos. Debido a que la ingeniería social involucra un elemento humano, prevenir estos ataques puede ser complicado para las empresas. Con este propósito y enfocados en el ámbito académico se fundó en el año 2003 en San Francisco, Estados Unidos, el Grupo de Trabajo Anti-phishing (APWG, según sus siglas en inglés). En la Fig. 1 extraída del reporte sobre ataques de phishing emitido por APWG durante el primer semestre del año 2017, se puede apreciar cómo se mantuvo prácticamente constante la cantidad de reportes de correo electrónicos de phishing, a excepción de un pico que se produjo durante el mes de marzo dado por 21 unidades porcentuales por encima de la media semestral.

A pesar de tener un comportamiento similar de un mes a otro, es importante destacar que las cifras de ataques a nivel mundial son considerablemente elevadas. Además, la tasa de crecimiento anual promedio del número de sitios web de phishing únicos detectados por dicho grupo fue de 36.29% durante los últimos seis años y un 97.36% durante los últimos dos años [6]. De acuerdo con el *Mobile World Congress* del año 2016, los ataques phishing pueden ser de distintos tipos: phishing engañoso, phishing basado en *malware*, *keyloggers* y *screenloggers*, secuestro de sesión, ataques de reconfiguración del sistema, robo de datos, phishing basado en DNS, phishing por inyección de contenido, phishing del hombre en el medio, phishing en buscadores, entre otros [7]. Esta diversidad hace más compleja la situación debido a que estos ataques pueden propagarse continuamente.

Un estudio reciente publicado en la revista *Computer Science Review* de la editora ELSEVIER resume que el ciclo de vida de un ataque phishing está compuesto por los siguientes elementos que se describen en la Fig. 2.

1. Se envía un enlace utilizando uno de los canales siguientes (correo electrónico, mensajería instantánea, blogs y foros en línea, servicios de mensajes cortos, servicios de intercambio de archivos de igual a igual y sitios web de redes sociales) o simplemente se solicitan en el cuerpo del mensaje las credenciales de acceso a las posibles víctimas.
2. Al hacer clic, el enlace redireccionará a las víctimas potenciales a un sitio o servicio web malicioso.
3. Los usuarios se vuelven vulnerables a medida que intentan iniciar sesión usando su credencial en el sitio web sospechoso o al enviar sus propias credenciales en el cuerpo del mensaje.
4. Las credenciales de inicio de sesión se transfieren a un servidor, o se instala un registrador de claves en el dispositivo de cómputo del usuario.
5. El phisher puede utilizar las credenciales para realizar delitos adicionales [8].

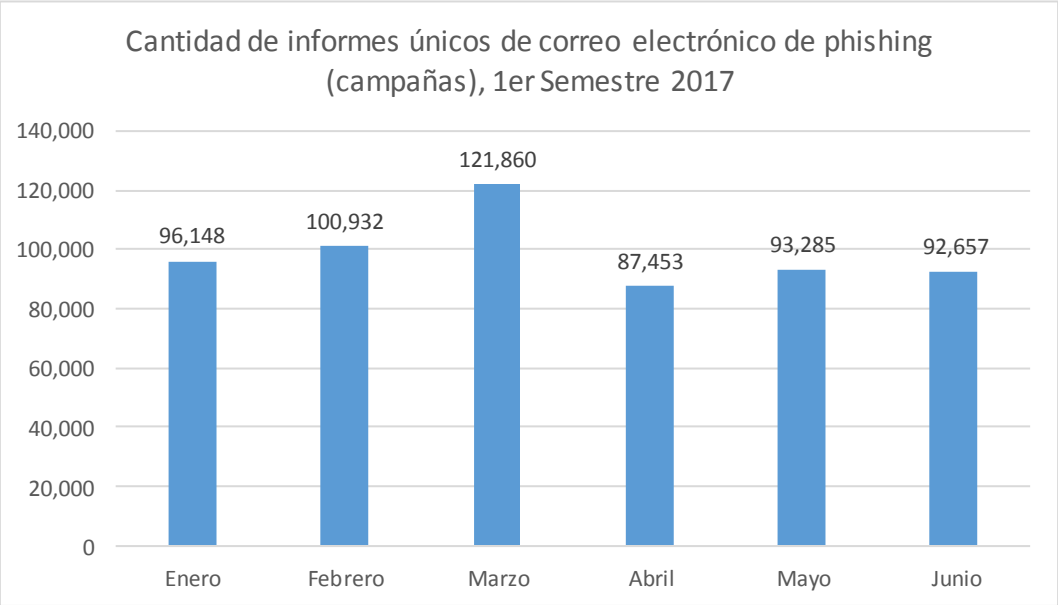

Figura 1: Informes únicos de correo electrónico de phishing, 1er Semestre 2017. [5]

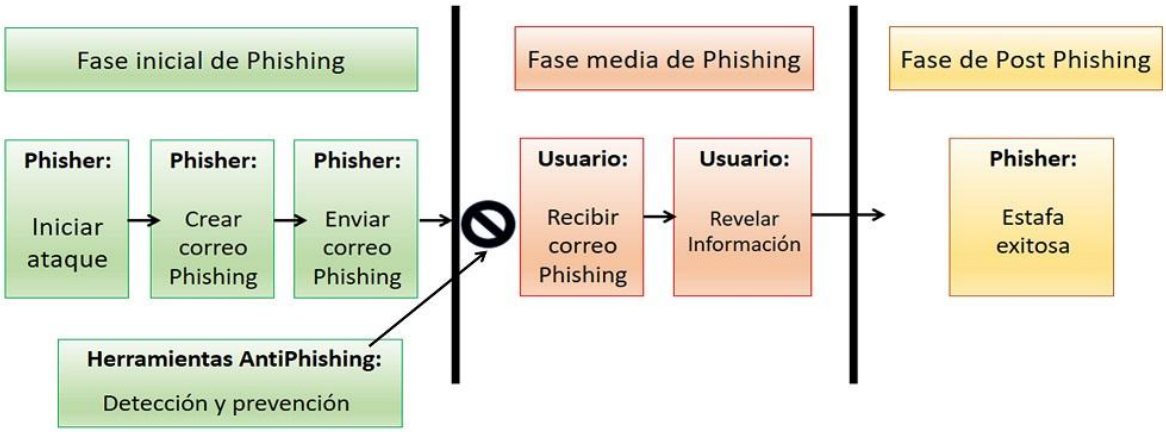

Figura 2: Ciclo de vida de un ataque de Phishing [8].

Para combatir el Phishing a nivel internacional se han desarrollado algunas herramientas informáticas que generalmente se utilizan para la detección y prevención ante la ocurrencia de dichos ataques. El estudio de trabajos precedentes permite agrupar estas soluciones en tres grandes grupos:

1. **Educacional y legal:**

   - Una de las primeras soluciones que surgieron para frenar y condenar este tipo de ataques fue la de aprobar desde el marco legal, un conjunto de leyes que reconocen como delitos al fraude electrónico y a la suplantación de identidad. Por ejemplo, en el estado de California en los Estados Unidos, desde el año 2005, se aprobó una legislación en contra del phishing, que tipifica como ilegal el hecho de utilizar cualquier medio electrónico para solicitar información a los usuarios en línea de una cierta empresa sin la autoridad de la misma.
   - Paralelo a esto existen soluciones encaminadas a la formación de los usuarios en identificar este tipo de ataques a través de un entorno de entrenamiento simulado de situaciones reales pasadas. Por ejemplo, el sitio web de APWG se especializa en este tipo de entrenamientos. Por otra parte, *Cofense* anteriormente *PhishMe*, es el proveedor líder de soluciones de defensa contra el phishing, cuenta con un alcance internacional y ofrece un enfoque de

colaboración a la ciberseguridad al permitir la participación de toda la organización ante amenazas de correo electrónico activas.

- Otras soluciones en el ámbito educacional son basadas en la experiencia del usuario, de esta forma se crean comunidades en línea anti-phishing para monitorear las actividades de phishing recientes y proporcionar noticias a los diferentes grupos de interés. Dichas experiencias constituyen buenas prácticas y se basan en casos reales relacionados con diferentes tipos de phishing, los cuales pueden ser posteriormente estudiados con el objetivo de hacer Internet un lugar más seguro. Por ejemplo, existen varias comunidades públicas anti-phishing, como son: APWG, *PhishTank*, *Millersmiles* y Symantec [9] [10] [11] [12].
- Estos métodos aunque ayudan a combatir estas malas prácticas no logran disminuir los ataques de phishing debido a que en el caso de las acciones legales no llegan a ser efectivas. Esto es debido a la breve vida útil del ataque en sí (generalmente 2 días) lo que ayuda esto al phisher a desaparecer una vez cometido el fraude haciendo la aplicación de la ley un poco difícil. A su vez el enfoque educacional demanda altos costos y requiere que los usuarios estén equipados con conocimientos de seguridad informática para estar mejor preparados ante la evolución acelerada de las técnicas de phishing. Estas técnicas se han evolucionado a procesos más sofisticados en los últimos años debido a los esfuerzos grupales de los phishers que emplean estrategias sistemáticas de ataque, lo que dificulta que incluso los expertos en seguridad y los agentes especializados de la ley mantengan sus habilidades actualizadas. Esto hace que los usuarios comunes sean vulnerables, incluso si estuvieran equipados con conocimientos básicos sobre phishing.

2. **Computacional utilizando métodos semi-automáticos:** Actualmente existen herramientas de software antispam que pueden bloquear correos electrónicos sospechosos. Sin embargo, estos programas bloquean constantemente una gran cantidad de correos electrónicos genuinos y los clasifican como correos electrónicos no deseados [13]. Los correos electrónicos mal clasificados como spam son simplemente falsos ejemplos positivos. Por tanto, uno de los objetivos finales de la herramienta anti-phishing computarizada es reducir los falsos positivos y aumentar los verdaderos positivos para que los usuarios puedan estar seguros de los resultados del filtro de su buzón sin tener que revisar manualmente su carpeta de correo no deseado.
   - Bases de datos (lista negra y lista blanca): Este enfoque se basa en el uso de una lista predefinida que contiene nombres de dominio o direcciones de correo electrónico que han sido reconocidos como dañinos. Debido a que cualquier usuario u organización ya sea esta pequeña a grande puede crear listas negras, las disponibles actualmente y públicas tienen diferentes niveles de efectividad de seguridad, particularmente con respecto a dos factores:
     a) Total de veces que se actualiza la lista negra y su disponibilidad constante.
     b) Calidad de los resultados con respecto a la tasa precisa de detección de phishing.

     En este sentido, el estudio en [14] acerca del análisis de listas negras refleja que la efectividad de detección de ataques de phishing en encuentra como promedio entre un 47% a un 83%. Algunos ejemplos son: *MXToolBox Blacklist Check, Barracuda Blacklist, SpamCop Blacklist, Spamhaus Blocklist*, etc. Similar a las listas negras se encuentran las listas blancas pero estas bases de datos almacenan los dominios y direcciones de correo legítimas con el objetivo de que estas acepten de la recepción de los correos electrónicos recibidos. Por ejemplo: *GoodMailSystems's Certified Email, Return Path Certification* y *Spamhaus Whitelist*.

3. **Métodos inteligentes de Maching Learning:** Teniendo en cuenta que phishing es un problema típico de clasificación, las técnicas de Aprendizaje Automático (ML) y Minería de Datos (DM) resultan apropiadas para obtener conocimiento a partir de las características del correo electrónico que dan origen al ataque. Ante dichos ataques, el éxito generalmente se encuentra en desarrollar sistemas automatizados de clasificación anti-phishing. Con el objetivo de mejorar el rendimiento del sistema predictivo se aplica un preprocesamiento del conjunto de características que presenta el mensaje para poder seleccionar el método "más" efectivo a utilizar. La efectividad puede medirse utilizando diferentes métodos de inteligencia artificial, tales como la ganancia de información, el análisis de correlación, entre otros. Algunos de los algoritmos [8] que pudieran utilizarse son:
   - Árboles de decisión [15].
   - Modelos probabilísticos (Naïve Bayes y redes bayesianas) [16].
   - Clasificación basada en reglas.

a) Clasificación asociativa (CA):
   I. Clasificación basada en la asociación (CBA) [17].
   II. Clasificación basada en asociación múltiple (CMAR) [18].
   III. Asociación basada en la clasificación multiclases (MCAR) [19].
b) Reglas de inducción ales como FOIL y RIPPER [20].
c) Cobertura o codiciosos, como el PRISM [21] y eDRI [22].

- Métodos de redes neuronales (NN) [23].
- Máquina de vectores de soporte (SVM) [24].
- Lógica difusa (FL) [25] [26].
- Métodos de impulso y paginación [27].
- Métodos de búsqueda tales como Algoritmos Genéticos (GA) [28] [29].

De cada uno de estos algoritmos se derivan diversos enfoques que son aplicados en los sistemas de clasificación anti-phishing, de ahí que se discutan en estos trabajos sus principales ventajas y debilidades. Fundamentalmente todos se basan en la aplicación de un conjunto de técnicas de inteligencia artificial, por lo que uno de los factores analizados siempre es el nivel de efectividad en la detección de ataques de phishing y en cuán medida son capaces de reducir la cantidad de falsos positivos. Por lo general todos estos estudios no rebasan el 85% en cuanto al nivel de efectividad.

Por otra parte, en un estudio diagnóstico preliminar realizado por los autores de esta investigación se identificaron dificultades y problemas asociados a la detección de ataques de phishing, formulándose las siguientes limitaciones:

- Estadísticas globales ofrecidas por la compañía *Phishme*, afirman que entre los 500 correos electrónicos sospechosos que puede recibir una empresa determinada, las tecnologías de filtrado de contenidos a menudo pasan por alto ataques de este tipo. Lo cual puede ser considerado como una brecha de seguridad potencialmente costosa.
- Un trabajador de oficina promedio recibe 125 correos electrónicos diariamente, lo que justifica que el phishing sea precisamente el principal medio de ataque en las infracciones informáticas.
- La gran dependencia de la tecnología para la protección contra amenazas cibernéticas en constante evolución ignora muchas veces el elemento más crítico: el factor humano.
- El phishing es dirigido hacia las personas, hacia los empleados de una empresa. Al enfocarse hacia los empleados, los atacantes están jugando con las probabilidades y esperando una marca fácil.
- Se producen demoras a la hora de identificar y clasificar direcciones de correo electrónico que representan amenazas reales, provocadas por el trabajo manual realizado por el personal encargado de la atención de incidentes de seguridad informática en las empresas.
- No existe una política de revisión de situaciones pasadas y análisis histórico en las decisiones tomadas a la hora de ejecutar la misma, elemento que pudiera ser utilizado como método para agilizar dicho proceso.
- Los especialistas de seguridad informática hacen uso en todo momento de sus expertas memorias y pericia en la detección de ataques de phishing, provocando en algunos momentos demoras a la hora de enfrentarse y reducir los riesgos ante amenazas reales.

Por tanto, surge la necesidad de resolver la siguiente interrogante: ¿Cómo disminuir el tiempo y la capacidad de detección de falsos positivos en la detección de ataques phishing a través de correo electrónico?

2. METODOLOGÍA COMPUTACIONAL

Para el desarrollo del sistema se emplearon los siguientes **métodos científicos**:

Métodos teóricos:

- **Analítico – sintético:** A partir de la documentación encontrada se seleccionaron los elementos claves que permiten comprender y diseñar el sistema para la detección de ataques phishing. De ellos se identificaron las ideas fundamentales y al mismo tiempo se detalló la información necesaria para el modelado correcto del negocio.
- **Histórico-lógico:** En la primera parte de la investigación se desarrolló un estudio del estado del arte de la problemática; así como se analizaron las ventajas y desventajas de cada una de las bases de datos internacionales para la detección de ataques y amenazas informáticas que se utilizaron para integrarlas en el sistema.

- **Método sistémico:** Para el desarrollo de la herramienta informática que se desarrolló y lograr que los módulos y submódulos que forman parte de la misma funcionen como un todo de manera armónica.

Métodos empíricos:

- **Entrevista:** Se utilizó la entrevista como una conversación planificada con especialistas en el área de la seguridad informática, para obtener información acerca del problema en cuestión.

Como metodología de desarrollo se utilizó el Proceso Unificado de Rational (RUP por sus siglas en inglés). Este es un proceso de desarrollo de software que unido al Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos [30]. Como herramienta de modelado se utilizó Visual Paradigm, herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software y está dotada de una buena cantidad de módulos para facilitar el trabajo durante la confección de un software [31].

Entre las tecnologías de desarrollo se seleccionó como lenguaje de programación Python en su versión 3.6, lenguaje multiplataforma[1], de código abierto, utilizado actualmente por programadores profesionales y que soporta la programación orientada a objetos [32]. Se utilizó además como marco de trabajo minimalista Flask en su versión 0.12.2. El mismo fue desarrollado basado en Python y permite crear aplicaciones web rápidamente y con la ejecución de un mínimo número de líneas de código [33]. También se utilizó se utilizó el gestor de base de datos NoSQL líder en el mercado, mongoDB en su versión 3.6, el cual permite a las empresas ser más ágiles, creando esquemas completamente flexibles que pudieran cambiar rápidamente cuando las aplicaciones evolucionan y proporcionando siempre funcionalidades que los desarrolladores esperan de las bases de datos tradicionales [34].

3. RESULTADOS Y DISCUSIÓN

Para el modelado de negocio pueden utilizarse técnicas y notaciones. Esto permite conocer los objetivos del negocio y plasmarlos en un modelo. A continuación, las siguientes figuras muestran la descripción del proceso Detección de ataques phishing a través de correo electrónico utilizando para ello el Modelo y Notación de Procesos de Negocio (BPMN, según siglas en inglés).

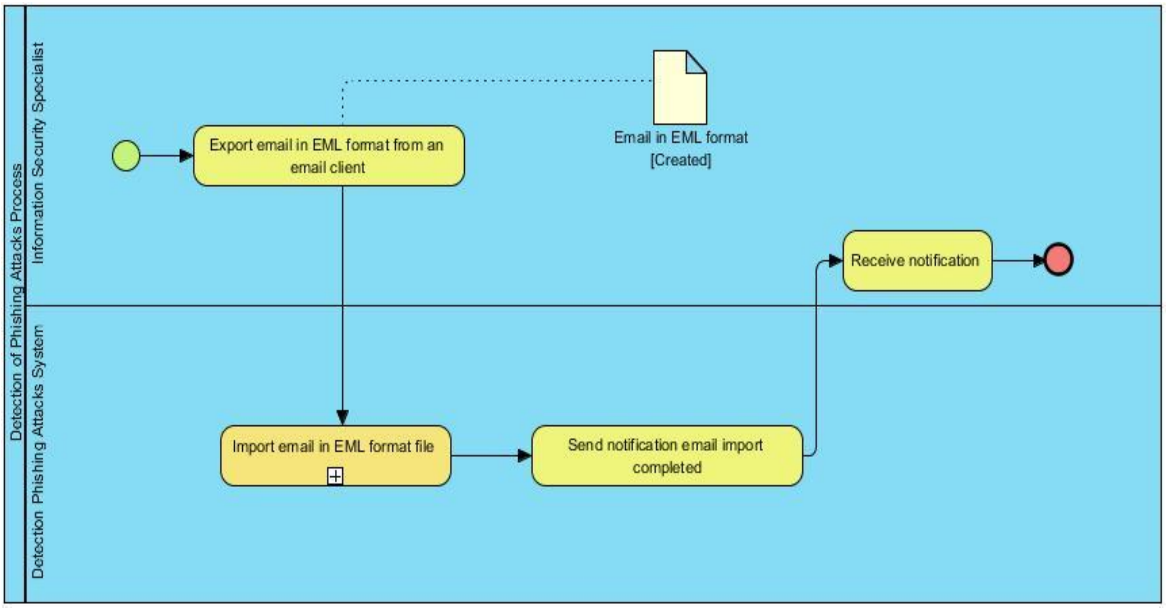

Figura 3. Descripción del proceso Detección de Ataques Phishing a través de correo electrónico utilizando BPMN.

[1] Es aplicable en varios Sistemas Operativos como: GNU/Linux, Windows, Mac, PalmOS, etc.

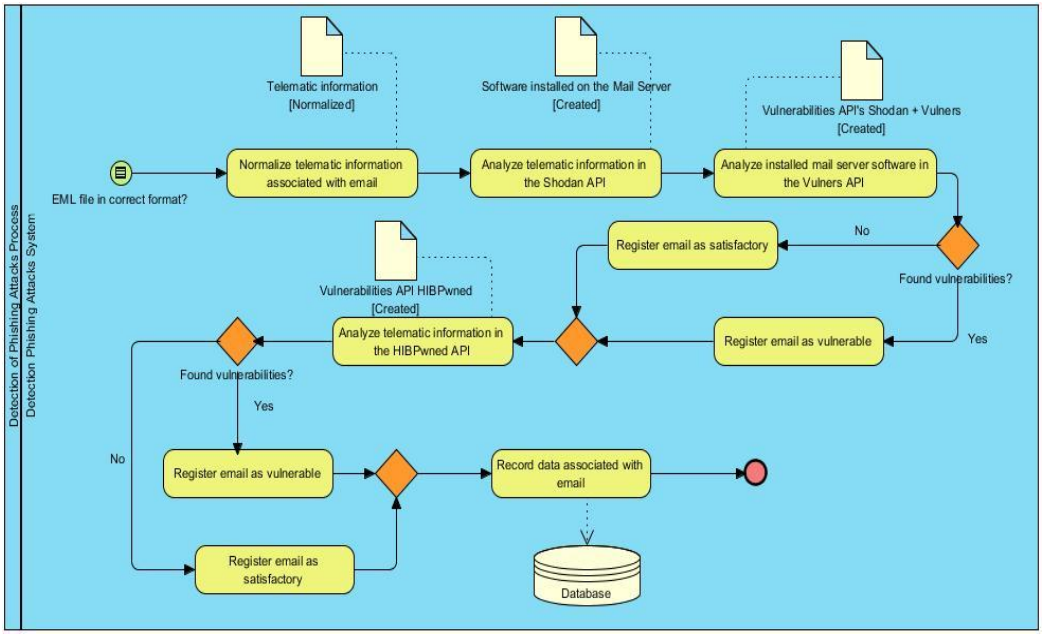

Figura 4. Descripción del subproceso Importar correo electrónico en formato EML utilizando BPMN.

Requerimientos del Sistema

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. De forma general estos requisitos indican lo que debe hacer el sistema. Los requerimientos funcionales que se identificaron para el desarrollo del sistema fueron los siguientes:

- **R1- Importar correo electrónico en formato EML:** Esta funcionalidad constituye la principal entrada de datos al sistema. Dado un fichero de correo electrónico sospechoso exportado anteriormente en formato EML, utilizando para ello un cliente de correo que incluya esa funcionalidad (Mozilla Thunderbird, Zimbra Web, Aplicación Correo Microsoft Windows 10, etc.). El sistema debe ser capaz de normalizar toda la información presente en la estructura del mismo para luego ser importada en el sistema. Se necesita importar: fecha de envío, asunto, ficheros adjuntos, dirección de correo del remitente y destinatario, así como los datos asociados al DNS[2] inverso del remitente (dirección IP[3] y hostname[4]).
- **R2- Clasificar vulnerabilidades de phishing presentes en un correo electrónico:** A través de esta funcionalidad se aplica directamente la investigación realizada, utilizado para clasificar y detectar los correos electrónicos sospechosos en posibles vulnerables a ataques de phishing o no.
- **R3- Gestionar correo electrónico:** Se pueden gestionar los correos electrónicos registrando de forma manual los datos asociados al mismo, para la posterior detección de las vulnerabilidades que puedan estar asociadas al mismo. Además, el sistema es capaz de modificar y eliminar los datos asociados a los insertados o importados anteriormente.
- **R4- Realizar búsquedas:** Se pueden realizar búsquedas simples y avanzadas filtrando la información por cada uno de la información asociada al correo electrónico.
- **R5- Exportar listado de correos electrónicos:** Se permite exportar el listado de correos que han sido analizados a diferentes formatos estándar con el fin de que puedan ser utilizados por otros sistemas. Los formatos establecidos son: PDF, CVS, TXT, XLS.
- **R6- Imprimir listado de correos electrónicos:** Con esta funcionalidad se permite imprimir el listado de correos electrónicos analizados.

[2] Servidor de Nombres de Dominio, según siglas en idioma ingles

[3] Protocolo de Internet, según siglas en idioma ingles

[4] Nombre de Equipo

Como parte del resultado se logró implementar un sistema web capaz de realizar análisis detección de ataques phishing a través de correo electrónico. En este sistema se encuentra incorporado el siguiente algoritmo para la clasificación de los ataques de phishing, el cual se describe en la Fig. 5. El algoritmo tiene como entrada una lista normalizada con la información telemática asociada a un conjunto de mensajes de correo electrónico que se desea analizar. Específicamente se utilizan los parámetros dirección IP del DNS inverso y la dirección electrónica del remitente, con el objetivo de realizar un análisis de clasificación a través de bases de datos de vulnerabilidades internacionales. Además de clasificar, el algoritmo es capaz de evaluar la efectividad del resultado, en dependencia de la cantidad de veces con que fue reportado "vulnerable" un elemento determinado y la ponderación que tiene cada base de datos. De ahí se obtiene en gran medida una idea de cuan sospechoso veraz puede resultar un determinado mensaje clasificado como phishing.

```
def classifyPhishing(emailList):
    classify_email_list = []
    for item in emailList:
        weight = 0
        state = "non-phishing"

        stateShodan = DefineStateByShodan(ip=item.get('reverse_dns_ip'))
        statePwned = DefineStateByPwned(sender=item.get('sender_mail_address'))

        if stateShodan == "vulnerable":
            weight+=2
            state = "phishing"
        if statePwned == "vulnerable":
            weight+=1
            state = "phishing"

        classify_email_list.append(state, weight)

    return classify_email_list
```

Figura 5: Fragmento de Código del Algoritmo para la clasificación de los mensajes de correo electrónico.

Pruebas del software

Se les denomina pruebas de software al conjunto de técnicas experimentales que se le aplican a los sistemas de software; son un elemento crucial para garantizar la calidad del producto y permiten validar las especificaciones, el diseño y la programación. Estas tienen como objetivo, además de descubrir errores, medir el grado en que el software cumple con los requerimientos definidos [35]. La fase de pruebas es una de las más valiosas del ciclo de vida de un software. En ese sentido, deben evaluarse todos los artefactos generados, lo que incluye especificaciones de requisitos, diagramas de diversos tipos, el código fuente y el resto de productos que forman parte de la aplicación, ejemplo: la base de datos. La prueba no puede asegurar la ausencia de errores; sólo puede demostrar que existen defectos en el software. [35].

Desde las primeras fases del desarrollo se diseñó una correcta estrategia de pruebas, que permitió durante todo el ciclo de vida del software la prueba constante de las funcionalidades y artefactos generados. Esta estrategia incluyó diversos tipos de prueba, la integración de los niveles de prueba: Pruebas Unitarias y Pruebas de Sistema; así como los métodos de prueba: Pruebas de Caja Blanca y Pruebas de Caja Negra. Para el caso del primer método se utilizó una técnica automatizada de software, la cual consiste en realizar pruebas al código del sistema utilizando *pyUnit*, *framework* de prueba de unidad estándar para Python. El segundo método se ejecutó utilizando la técnica de Partición Equivalente.

Resultado de las Pruebas

Para realizar las pruebas de caja blanca se implementó una clase *TestPhishingProject*, la cual importa y hace uso de las librerías implementadas en el módulo unittest (pyUnit). La Fig. 6 muestra un fragmento de dicha clase.

A medida que se fueron realizando las pruebas correspondientes se detectaron un total de 16 no conformidades, las cuales se resolvieron mientras se desarrollaban las funcionalidades. Una vez realizada la corrección de los errores detectados en pequeñas iteraciones, los métodos de cada clase se ejecutaron de forma correcta, obteniendo de ellos los resultados esperados.

```python
import unittest
from phisingMess import views

class TestPhishingProject(unittest.TestCase):
    def test_DefineStatusPwned(self):
        self.assertEqual(views.DefineStatusPwned('ahdominguez@uci.cu'), "OK")

    ...

if __name__ == "__main__":
    unittest.main()
```

Figura 6. Fragmento de código de la clase utilizada para la realización de las Pruebas de Unidad al sistema.

Las pruebas de caja negra se centraron en el cumplimiento de las funcionalidades descritas en el listado de requerimientos. Las no conformidades se clasificaron en Alta, Media o Baja en dependencia del impacto que tuvieran, generalmente las altas responden a errores técnicos relacionados directamente con alguna funcionalidad interna y las bajas tienden a ser errores ortográficos, validaciones, etc. La figura 7 muestra en una gráfica de barras los principales resultados de las pruebas de caja negra realizadas. Como se puede apreciar, se realizaron dos iteraciones de pruebas, detectando en la primera y segunda iteración un total de 48 y 26 no conformidades respectivamente. Comparando los resultados entre dichas iteraciones se evidencia una reducción considerable (alrededor de un 54.16%) de los principales errores detectados en el sistema una vez ejecutada la 2da iteración. Finalmente se realizó una última prueba donde se corrigieron todas las no conformidades.

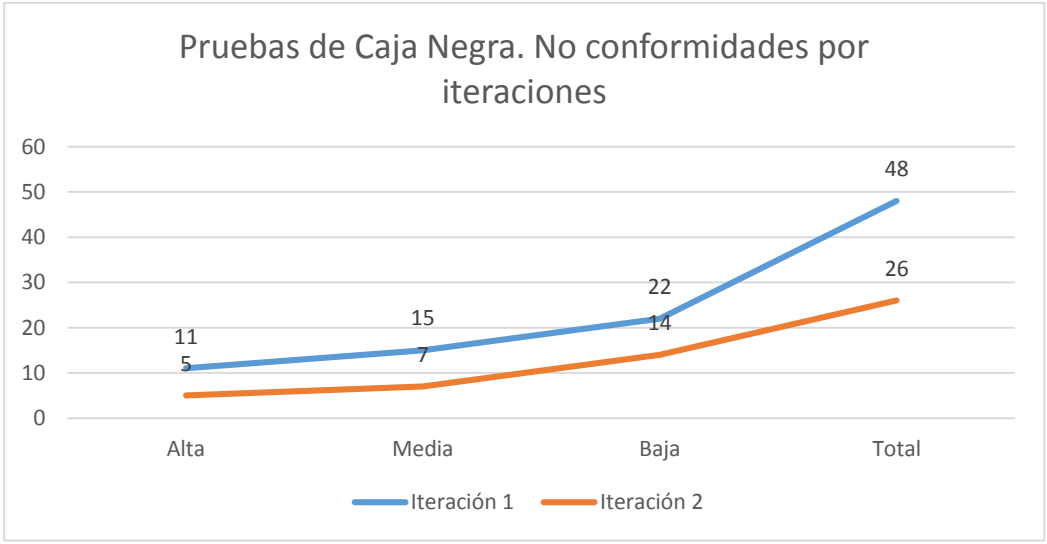

Figura 7: Impacto de las no conformidades detectadas contra cantidad de iteraciones de prueba.

4. CONCLUSIONES

Se concluye este artículo dando cumplimiento al objetivo propuesto para la realización del mismo. Mediante el presente documento, se ilustra al lector, acerca del proceso de desarrollo del Sistema para la Detección de Ataques Phishing a través de correo electrónico. Además, esta investigación arrojó los siguientes resultados:

- Se formalizó un estudio acerca del proceso de Detección de Ataques Phishing a través de correo electrónico.
- Se describió la metodología, el lenguaje de modelado, tecnologías y herramientas utilizadas para el desarrollo de dicho sistema.
- Se evaluó la viabilidad del sitio a través de pruebas de software efectuadas para el nivel de unidad, las cuales arrojaron resultados favorables posibilitando dar cumplimiento a las funcionalidades previstas para el mismo.

- Disponer de un sistema que permita la integración y la posterior correlación de patrones de comportamientos obtenidos a través de bases de datos internacionales de seguridad como: "*Shodan*", "*Vulner*" y "*Have I Been Pwned?*" permitió una mejor detección de ataques phishing.

Utilizar los resultados teóricos de detección de ataques phishing de forma automatizada hizo su aplicación más fácil y cómoda para los investigadores y especialistas del área de seguridad informática.

RECONOCIMIENTOS

Los autores desean reconocer al trabajo de colaboración existente entre la Universidad de Ciencias Informáticas de Cuba y la Universidad Federal del Sur de Rusia. Se agradece específicamente a profesores del Instituto de Tecnología Informática y Seguridad de la Información, Ingeniería y Academia Tecnológica con los cuales se llegó a estos resultados que se encuentran en este artículo.

REFERENCIAS

[1] O. A. Hathaway, R. Crootof, P. Levitz, H. Nix, A. Nowlan, W. Perdue, *et al.*, "The Law of Cyber-Attack," vol. 100, pp. 817-885, 2012.
[2] Symantec, "Official website of the Symantec corporation," ed, 2018.
[3] N. Lord. (2018). *Official website of the Digital Guardian company*. Available: https://digitalguardian.com/blog/social-engineering-attacks-common-techniques-how-prevent-attack [Accessed: 04/02/3018]
[4] C. Hadnagy, *Social engineering: the art of personal hacking*: Ediciones Anaya Multimedia, 2011.
[5] APWG. Phishing Attack Trends Report - 1 Half 2017. 2017. [Online]. Available: http://docs.apwg.org/reports/apwg_trends_report_h1_2017.pdf [Accessed: 01/02/2018]
[6] Z. Dou, I. Khalil, A. Khreishah, A. Al-Fuqaha, and M. Guizani, "Systematization of Knowledge (SoK): A Systematic Review of Software-Based Web Phishing Detection," vol. 19, pp. 2797-2819, 2017.
[7] K. F. Mbah, "A phishing e-mail detection approach using machine learning techniques," Master's Thesis, University of New Brunswick, Fredericton and Saint John, NB, Canada, 2017.
[8] I. Qabajeh, F. Thabtah, and F. Chiclana, "A recent review of conventional vs. automated cybersecurity anti-phishing techniques," *Computer Science Review*, vol. 29, pp. 44-55, 2018/08/01/ 2018.
[9] D. Jevans. (2003). *Anti-Phishing Working Group (APWG)*. Available: http://www.antiphishing.org/ [Accessed: 02/04/2018]
[10] PhishTank. (2011). *PhishTank*. Available: http://www.phishtank.com/ [Accessed: 21/03/2018]
[11] M. Bright. (2011). *MillerSmiles*. Available: http://www.millersmiles.co.uk/ [Accessed: 10/04/2018]
[12] B. Nahorney. (2015). *The MessageLabs Intelligence Annual Security Report: 2009 Security Year in Review*. Available: http://www.symantec.com/content/en/us/enterprise/other_resources/intelligence-report-06-2015.en-us.pdf [Accessed: 22/01/2018]
[13] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Tutorial and critical analysis of phishing websites methods," *Computer Science Review*, vol. 17, pp. 1-24, 2015.
[14] S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs, "Who falls for phish?: a demographic analysis of phishing susceptibility and effectiveness of interventions," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2010, pp. 373-382.
[15] C. Gutierrez, T. Kim, R. Della Corte, J. Avery, M. Cinque, D. Goldwasser, *et al.*, "Learning from the Ones that Got Away: Detecting New Forms of Phishing Attacks," *IEEE Transactions on Dependable and Secure Computing*, 2018.
[16] S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning," *Decision Support Systems*, vol. 107, pp. 88-102, 2018/03/01/ 2018.
[17] B. L. W. H. Y. L. Ma, Bing, "Integrating classification and association rule mining," in *Proceedings of the fourth international conference on knowledge discovery and data mining*, 1998, pp. 80-86.
[18] W. Li, J. Han, and J. Pei, "CMAR: Accurate and efficient classification based on multiple class-association rules," in *icdm*, 2001, pp. 369-376.
[19] F. Thabtah, P. Cowling, and Y. Peng, "MCAR: multi-class classification based on association rule," in *Computer Systems and Applications, 2005. The 3rd ACS/IEEE International Conference on*, 2005, pp. 1-7.
[20] W. W. Cohen, "Fast effective rule induction," in *Machine Learning Proceedings 1995*, ed: Elsevier, 1995, pp. 115-123.

[21] J. Cendrowska, "PRISM: An algorithm for inducing modular rules," *International Journal of Man-Machine Studies*, vol. 27, pp. 349-370, 1987.
[22] F. Thabtah, R. M. Mohammad, and L. McCluskey, "A dynamic self-structuring neural network model to combat phishing," in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 4221-4226.
[23] S. Grossberg, "Nonlinear neural networks: Principles, mechanisms, and architectures," *Neural networks*, vol. 1, pp. 17-61, 1988.
[24] T. Joachims, "Making large-scale SVM learning practical," Technical Report, SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund1998.
[25] M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah, "Intelligent phishing detection system for e-banking using fuzzy data mining," *Expert Systems with Applications*, vol. 37, pp. 7913-7921, 2010/12/01/ 2010.
[26] A. Almomani, B. Gupta, T.-c. Wan, A. Altaher, and S. Manickam, "Phishing dynamic evolving neural fuzzy framework for online detection zero-day phishing email," *arXiv preprint arXiv:1302.0629*, 2013.
[27] R. Islam and J. Abawajy, "A multi-tier phishing detection and filtering approach," *Journal of Network and Computer Applications*, vol. 36, pp. 324-335, 2013/01/01/ 2013.
[28] V. Shreeram, M. Suban, P. Shanthi, and K. Manjula, "Anti-phishing detection of phishing attacks using genetic algorithm," in *2010 INTERNATIONAL CONFERENCE ON COMMUNICATION CONTROL AND COMPUTING TECHNOLOGIES*, 2010, pp. 447-450.
[29] H. S. Hota, A. K. Shrivas, and R. Hota, "A Proposed Bucket Based Feature Selection Technique (BBFST) for Phishing e-Mail Classification," Singapore, 2018, pp. 189-194.
[30] A. P. Hernández, Annia, "Sistema Informático de Auditoría y Control (SIGAC). Módulo de Planificación," Ingeniería en Ciencias Informáticas Tesis de Pregrado, Facultad 2, Universidad de las Ciencias Informáticas, La Habana, Cuba, 2009.
[31] V. Paradigm, "Visual paradigm for uml," *Visual Paradigm for UML-UML tool for software application development*, vol. 72, 2013.
[32] I. Challenger, Y. Díaz, and R. A. Becerra, "The Python programming language," vol. 20, pp. 1-12, 2014.
[33] A. Ronacher. (2018). *Flask framework Official Web Portal*. Available: http://flask.pocoo.org/ [Accessed: 20/03/2018]
[34] D. Merriman, E. Horowitz, and K. Ryan. (2018). *Official website of the Database Management System mongoDB*. Available: https://www.mongodb.com/es [Accessed: 20/01/2018]
[35] R. S. Pressman, *Software engineering: a practitioner's approach*, 7th ed.: Palgrave Macmillan, 2009.

SOBRE LOS AUTORES

Antonio Hernández Dominguez

Graduado de Ingeniero en Ciencias Informática en el año 2009. Actual profesor de la Facultad 2 y director del Centro de Telemática de la Universidad de las Ciencias Informáticas. Imparte docencia de pregrado en dicha universidad en las asignaturas Matemática 1 y 2 y Sistemas de Bases de Datos 1 y 2. Aspirante a Doctor en Ciencias Técnicas. Categoría docente: Profesor Asistente.

Sergey Storchak (Сергей Сторчак)

Especialista en Seguridad de la Información y Aspirante a Doctor en Ciencias del Instituto de Tecnología Informática y Seguridad de la Información, Ingeniería y Academia Tecnológica de la Universidad Federal del Sur de Rusia.