An Early Alert System for Software Vulnerabilities based on Vulnerability Repositories and Social Networks

Néstor Fabián Riveros
Facultad de Ciencias y Tecnologías
Universidad Católica "Nuestra Señora de la Asunción"
Asunción, Paraguay
fabian.riveros@uc.edu.py

Carlos Rodríguez
Facultad de Ciencias y Tecnologías
Universidad Católica Nuestra "Señora de la Asunción"
Asunción, Paraguay.
carlos.rodriguez@uc.edu.py

***Abstract*—The huge amount of information regarding software vulnerabilities, the multiple and heterogeneous information sources, and the lack of awareness about the dangers of software vulnerabilities, exacerbates the risks of security threats being materialized. In this complex context, this paper approaches the problem of managing early alerts for software vulnerablities by leveraging existing vulnerability information found in vulnerability repositories and social networks. To this end, we propose a solution based on techniques that stem from automated retrieval of information about vulneratilities from the above sources, user-defined preferences regarding their technological environment and intelligent vulnerability tagging. Our user studies reveal the feasibility of our approach as a tool for managing early alerts regarding software vulnerabilities and keeping security professionals aware of them.**

Keywords: Software vulnerabilities, intelligent tagging, early alerts, natural language processing.

I. Introducción

De un tiempo a esta parte, las empresas han aumentado su dependencia del software, el cual ayuda a mejorar la eficiencia de sus procesos. Esto, sin embargo, las lleva a ser más vulnerables al riesgo de ataques cibernéticos que podrían resultar en graves daños y pérdidas económicas [1]. En este contexto, el aumento del home office[1] a causa del COVID-19, trae consigo un nuevo desafío al expandirse la conectividad fuera de las empresas,[2] pues los sistemas de software no actualizados con los últimos parches de seguridad, las contraseñas débiles, y la falta de protección contra virus y malware, hacen que los dispositivos y sistemas de software utilizados para el efecto sean especialmente vulnerables[3] y podrían, por lo tanto, exponer a las empresas a algún tipo de ciberataque.

Una vulnerabilidad destacada del año 2018, conocida como EternalBlue,[4] causó graves problemas en el protocolo Server Message Block (SMB) de Microsoft.[5] Uno de los ataques más famosos de su uso fue el de Wanna Cry,[6] que afectó a más de 300.000 empresas en todo el mundo y generó unos costes totales de alrededor de 4 mil millones de dólares en pérdidas. Cabe destacar que esto se podría haber evitado, pues al momento del ataque ya existían los parches de seguridad para dichas vulnerabilidades, los cuales fueron publicados meses antes de los incidentes.

Como se pudo constatar, las consecuencias de no aplicar los parches de seguridad pueden ser catastróficas.[7] En este contexto, en un estudio realizado por Beattie et al. [2] se analizan la problemática de las actualizaciones del software y cuándo aplicar los parches de seguridad al sistema. El artículo explica la disyuntiva entre la aplicación de una actualización de seguridad que pudiera no ser aún estable y el riesgo de que el software sea comprometido debido a la espera prolongada por una actualización estable.

Según un artículo publicado por Panda Security en el año 2018 (ver nota al pie 7), se estima que para el 2021, el coste del cibercrimen alcanzará los 6 billones de dólares a nivel mundial. Concluyen en dicho artículo que, típicamente, las empresas no tienen políticas de aplicación de parches de seguridad y que las vulnerabilidades pueden pasar desapercibidas. En otras palabras, hay poca conciencia sobre la amenaza que representan las vulnerabilidades del software y la importancia de las actualizaciones de seguridad. Por otro lado, el reporte publicado por el Banco Interamericano de Desarrollo (BID), denominado "Riesgos, avances y el camino

[1]https://www.rdstation.com/mx/blog/home-office/
[2]https://mexico.unir.net/vive-unir/seguridad-informatica-home-office-mexico/
[3]https://www.3cx.es/blog/seguridad-home-office/
[4]https://www.welivesecurity.com/la-es/2018/05/10/exploit-eternalblue-registra-mayor-actividad-ahora-que-durante-brote-wannacryptor/
[5]https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0144
[6]https://en.wikipedia.org/wiki/WannaCry_ransomware_attack
[7]https://www.pandasecurity.com/es/mediacenter/seguridad/consecuencias-no-aplicar-parches/

N.F. Riveros, C. Rodriguez. "An Early Alert System for Software Vulnerabilities based on Vulnerability Repositories and Social Networks." The 47th Latin American Computing Conference (CLEI 2021), October 2021, San Jose, Costa Rica

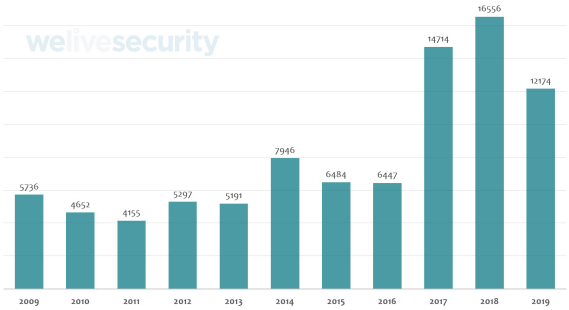

Fig. 1. Vulnerabilidades reportadas desde el 2009 hasta el 2019 (extraído de Welive security (ver nota al pie 9))

a seguir en América Latina y el Caribe",[8] se menciona el importante crecimiento en la región del interés sobre aspectos relacionados a la seguridad del software y el cibercrimen, lo cual podría interpretarse como un aumento (aunque no lo suficiente) en la consciencia de las personas y empresas sobre las consecuencias entorno a estas cuestiones.

El aspecto de la seguridad del software se ha vuelto un desafío mayor en los últimos años, ya que la cantidad de vulnerabilidades, *exploits* y ataques siguen creciendo a un ritmo constante. En este contexto, las fuentes de información de ciberseguridad son elementos fundamentales en la operación diaria de los profesionales de la seguridad [1]. Entre las fuentes más importantes se encuentran los CVE (del inglés, *Common Vulnerabilities and Exposures*) [3], mediante los cuales se documentan las vulnerabilidades del software oficialmente reportadas por profesionales de la seguridad (p.ej., hackers éticos).

En la Figura 1, se puede observar un gráfico estadístico de la cantidad de vulnerabilidades documentadas del software, reportadas desde el 2009 hasta el 2019 a nivel mundial.[9] Esta figura denota una clara tendencia hacia aumento de las vulnerabilidades reportadas anualmente, en particular, durante los últimos años. Adicionalmente a las fuentes oficiales sobre vulnerabilidades (p.ej., CVE), existe otra categoría de vulnerabilidades conocidas como vulnerabilidades del día cero (del inglés, *zero-day o 0-day vulnerability*).[10] Éstas se refieren a vulnerabilidades que son desconocidas por personas u organizaciones a quienes afectan y quienes debería estar interesadas en mitigarlas.

Independientemente de las fuentes consultadas, el descubrimiento de nuevas vulnerabilidades requiere típicamente la inversión de mucho tiempo y esfuerzo, optando típicamente por una o más de las siguientes opciones: (i) conocer la estructura del buscador de cada repositorio (p.ej., CVE o NVD (del inglés, *National Vulnerability Database*[11]) con el fin de poder encontrar información útil; (ii) estar inscripto en listas de correos dedicadas a difundir información sobre vulnerabilidades; (iii) seguir a profesionales y expertos de seguridad fiables en las redes sociales. Sumado a estos esfuerzos, se debe tener en cuenta, además, que los desarrolladores de software y otros profesional de las Tecnologías de Información y Comunicaciones (TIC) no siempre comprenden en profundidad los aspectos relativos a la ciberseguridad,[12] lo cual podría exacerbar la dificultad de acceso a la información relativa a vulnerabilidades del software.

En el flujo de pasos en un proceso de descubrimiento de alguna vulnerabilidad, detectado un fallo, el usuario debe aplicar los parches de seguridad y revisiones sugeridos según los CVE correspondientes. Muchas vulnerabilidades podrían tener un nivel de gravedad bajo, por lo que se podría retrasar la atención. En cambio, para aquellas vulnerabilidades críticas, se deben tomar las acciones pertinentes con la mayor inmediatez posible. Sin embargo, según una publicación realizada por el sitio Welive Security,[13] "la excesiva cantidad de alertas produce fatiga de seguridad", lo cual podría acarrear que aquellas vulnerabilidades verdaderamente críticas pasen desapercibidas. Este inconveniente es análogo a las fatigas por alarmas presentes, típicamente, en contextos relativos a las emergencias médicas.[14]

Por todo lo anterior, hemos verificado, que estar informados en el menor tiempo y con la mayor precisión posible sobre los nuevos descubrimientos de vulnerabilidades y que afecten a un entorno de software específico, conduciría a mejorar el conocimiento sobre la existencia de vulnerabilidades de interés para las empresas y la conciencia sobre la importancia de la actualización de los sistemas de software utilizados por las mismas.

Este artículo propone el abordaje de los problemas indicados previamente mediante la propuesta de una solución que permita generar alertas tempranas sobre vulnerabilidades del software. Tomamos como base las fuentes de datos obtenidas de registros oficiales de vulnerabilidades y, en forma complementaria, información sobre vulnerabilidades extraída de las redes sociales. La primera fuente, provee información formalmente reportada a organizaciones encargadas de gestionar la información sobre vulnerabilidades (p.ej., CVE), mientras que la segunda permite la identificación de potenciales vulnerabilidades del día cero. Para el efecto, combinamos técnicas de recuperación de la información [4], NLP (del inglés, *Natural Language Processing*) [5] y etiquetado automático e

[8]https://publications.iadb.org/publications/spanish/document/Reporte-Ciberseguridad-2020-riesgos-avances-y-el-camino-a-seguir-en-America-Latina-y-el-Caribe.pdf

[9]https://www.welivesecurity.com/la-es/2020/01/30/vulnerabilidades-exploits-reportados-2019/

[10]https://www.vicarius.io/blog/zero-day-vulnerabilities-the-basics

[11]https://nvd.nist.gov

[12]https://www.sans.org/blog/2015-state-of-application-security-closing-the-gap

[13]https://www.welivesecurity.com/la-es/2016/10/11/fatiga-de-seguridad-incidentes/

[14]https://www.philips.es/healthcare/services/clinical-services/alarm-fatigue

inteligente [6] de vulnerabilidades. El empleo de estas técnicas permite identificar, extraer y clasificar vulnerabilidades en forma conveniente de manera a permitir una búsqueda y descubrimiento más efectivos de vulnerabilidades que sean de interés para los usuarios.

Este trabajo se compone de 5 secciones distribuídas de la siguiente manera: En la sección 2 se presentan los trabajos relacionados. La sección 3 presenta conceptos y definiciones útiles que permitirán una mejor comprensión del contexto del problema. En la sección 4 presentamos la propuesta de solución, y en la sección 5 se presenta la implementación de la propuesta y evaluación de la misma. Finalmente, en la sección 6 exponemos las conclusiones y trabajos futuros.

II. Trabajos Relacionados

A continuación, discutimos los trabajos relacionados categorizándolos de la siguiente manera: (i) repositorios y servicios en línea de vulnerabilidades; (ii) descubrimiento y extracción de la información sobre vulnerabilidades; (iii) gestión y etiquetado de la información; (iv) procesamiento de Language Natural.

A. Repositorio y Servicios en Línea de Vulnerabilidades

Entre los servicios en línea sobre seguridad informática, HISPASEC[15] provee un mecanismo de suscripción a una lista de correos para el envío diario de avisos sobre vulnerabilidades informáticas, con el propósito de divulgar y concienciar a los usuarios sobre problemáticas del sector. La información proveída es muy amplia y de diversidad de contenido, y no precisamente sobre temas específicos relativos la realidad tecnológica del usuario.

Google Hacking Database[16] es un repositorio muy importante en el ámbito de la seguridad informática, el cual provee un potente motor de búsqueda para el descubrimiento de vulnerabilidades. La información contenida en este repositorio es muy relevante y la vez de alta tecnicidad. Esto implica que el usuario final deberá de conocer muy bien la terminología y los aspectos técnicos entorno a las vulnerabilidades. Debido al tecnicismo en el ámbito de las vulnerabilidades, debe invertir una cantidad de tiempo y esfuerzo significativos para encontrar vulnerabilidades y parches de seguridad de su interés (p.ej., en correspondencia con su realidad tecnológica).

En el ámbito de soluciones y servicios de información sobre vulnerabilidades integrados, TimeSys[17] provee un servicio de suscripción dirigido a profesionales del área de seguridad. El servicio se centra en el monitoreo de incidentes de seguridad enfocado exclusivamente en plataformas Linux. El servicio provee un conjunto de paneles (en inglés, *dashboards*) que permiten acceder a información sobre vulnerabilidades del tipo CVE.

Otras fuentes útiles de información sobre vulnerabilidades del software incluyen GitHub,[18] Information Security Stack Exchange[19] y Stack Overflow.[20] Los mismos proveen referencias técnicas que contienen una base de conocimientos muy importante sobre vulnerabilidades del software y ciberseguridad en general.

Las redes sociales son también fuentes importantes de información sobre vulnerabilidades del software [7]. Por ejemplo, en Twitter[21] se pueden encontrar cuentas verificadas de especialistas de seguridad informática y que día a día comparten sus conocimientos técnicos en este ámbito, por lo que actuan de sensores cruciales para estar informados sobre nuevos CVEs, así como también sobre vulnerabilidades del día cero. Sin embargo, similarmente a lo que ocurre con las listas de correos, la información no está filtrada y por ende no siempre resulta de utilidad para el contexto tecnológico del usuario que la recibe.

Finalmente, se encuentran también las fuentes de información tradicionales tales como la NVD, OWASP[22] y Mitre,[23] las cuales mantienen repositorios sobre vulnerabilidades, estándares y esquema de clasificación de debilidades del software, entre otras informaciones de utilidad.

B. Extracción de la información sobre vulnerabilidades

Para extraer la información de Twitter y desde las fuentes de información que documentan las vulnerabilidades del software, rescatamos algunos estudios relevantes tales como el trabajo de Alqahtani et al. [8]. En el mismo, los autores abordan el enfoque de modelado semántico, que aprovecha las tecnologías Web para establecer vínculos de trazabilidad entre los repositorios de avisos de seguridad y otros repositorios de software. Exploramos este estudio para aprovechar la propuesta semántica mencionada y así optimizar nuestra extracción de la información. En el mismo contexto, identificamos varias investigaciones referentes a diversas técnicas propuestas para mejorar la extracción del contenido y el aprendizaje automatizado [9], [10], [11], [12]. La más importante en el contexto de nuestro trabajo es aquella propuesta por Arnav et al. [9], en la que se menciona la manera de extracción de la información sobre vulnerabilidades desde el repositorio NVD.

[15]https://hispasec.com/es/
[16]https://www.exploit-db.com/google-hacking-database
[17]https://www.timesys.com
[18]https://github.com/
[19]https://security.stackexchange.com/
[20]https://stackoverflow.com/
[21]https://twitter.com/
[22]https://owasp.org/
[23]https://cve.mitre.org/

C. Gestión y Etiquetado de Información sobre Vulnerabilidades

En cuanto a la gestión de la información, la base de conocimientos existentes fue fundamental para abordar nuestro trabajo y tomamos como referencia la investigación [1], donde los autores abordan las fuentes de información de vulnerabilidades, la heterogeneidad de las mismas y la utilización de lenguaje natural para consultarlas. También, en el mismo contexto, en la propuesta de Atymtayeva et al. [13], se explora la construcción de una base de conocimientos para expertos en seguridad. Ambas propuestas tienen relación directa en la construcción de bases de conocimientos y que es fundamental para el descubrimiento de las vulnerabilidades. De estos trabajos rescatamos las referencias de fuentes y las técnicas de exploración de la información utilizadas. En estos trabajos profundizan en la creación de bases de datos en donde se busca facilitar la búsqueda de la información sobre vulnerabilidades conocidas. Esto implica que los usuarios finales deben dedicar tiempo y esfuerzo en la búsqueda proactiva de vulnerabilidades relacionadas a su entorno tecnológico.

El uso de información recolectada a partir de las redes sociales ha demostrado ser de gran utilidad en contextos variados tales como la detección y predicción de desastres naturales, eventos sociales globales, entre otros contextos. Por ejemplo, Sakaki et al. [14] exploran la clasificación semántica tweets para alertas tempranas en el ámbito de los terremotos. Mientras que Alexander [15] propone el uso de las redes sociales como un termómetro de variedad de eventos globales y como mecanismo de gestión de desastres y crisis. Si bien estos trabajos han sido realizados en contextos diferentes a la gestión de información sobre vulnerabilidades, las problemáticas de base y mecanismos utilizados guardan (en esencia) una relación con la gestión de alertas en el ámbito de las vulnerabilidades del software.

Finalmente, en el ámbito del etiquetado para propósitos de gestión de la información, Vig et al. [6] proponen un mecanismo de etiquetado documentos basados en técnicas de aprendizaje de máquina y VSM (del inlgés, *Vector Space Model*). Dicho mecanismo contribuye a la clasificación de contenido descubierto y una mejor presentación y filtrado de la información en base a las preferencias de los usuarios finales.

D. Procesamiento de Lenguaje Natural

El uso de técnicas de NLP ha tenido un fuerte repunte en los últimos años gracias a los importantes avances recientes en el área [16]. Por ejemplo, técnicas basadas en la representación distribuida de palabras [17] ha permitido la creación de una variedad de mecanismos de procesamiento de lenguage natural para propósitos de clasificación y compresión de lenguaje natural. *Word2vec* es una de tales técnicas [5]. El mismo utiliza un modelo de red neuronal para aprender asociaciones de palabras de un cuerpo de texto en un dominio, creando un VSM con la propiedad de que las palabras semánticamente relacionadas se cuentran cercanas unas a otras en el espacio vectorial.

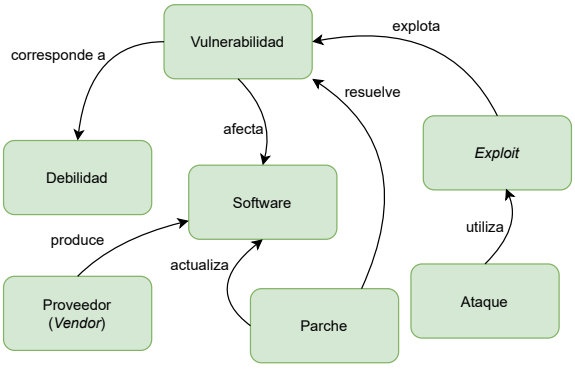

Fig. 2. Conceptos claves en el ámbito de las vulnerabilidades del software.

En el ámbito de las vulnerabilidades del software, Mumtaz et al. [10] proponen la incrustación de palabras (en inglés, *word embedding*) de dominio específico utilizando como corpus fuentes de datos conteniendo información relativa a vulnerabilidades del software. Mokhov et al. [11], sin embargo, han empleado técnicas de NLP para el análisis de código fuente con el propósito de detectar debilidades y vulnerabilidades del software. Mientras que Khazaei et al. [18] han propuesto la predicción del puntaje basado en CVSS (del inglés, *Common Vulnerability Scoring System*[24]) utilizando técnicas de NLP sobre las descripciones de vulnerabilidades.

Este artículo contribuye al avance del estado del arte en el ámbito de las alertas tempranas sobre vulnerabilidades del software mediante una propuesta que combina técnicas de recuperación de la información [4], expansión de consultas (*query expansion*) [19], categorización y etiquetado inteligente [6] de vulnerabilidades mediante técnicas de *word embeddings* [5], y presentación de la información mediante mecanismos basados en *timelines* [20] y *push notifications* [21].

III. Vulnerabilidades del Software

En esta sección, nos enfocamos en conceptos, definiciones y terminologías claves dentro de contexto de las vulnerabilidades del software. Éstos nos permitirán establecer las bases para la propuesta de nuestro trabajo introducida en la siguiente sección.

La Figura 2 presenta un modelo conceptual con los elementos claves dentro del ámbito de las vulnerabilidades del software. Una *vulnerabilidad* es una falla presente en un *software* que permite vulnerar o comprometer la seguridad del mismo. Las vulnerabilidades son categorizadas en base a la

[24]https://nvd.nist.gov/vuln-metrics/cvss

debilidad que las caracteriza. Ejemplos de debilidades muy populares son el *SQL Injection* y *Cross-site Scripting* (CSS). Las debilidades que pueden afectar al software están categorizadas y documentadas mediante un mecanismo conocido como CWE (del inglés, *Common Weakness Enumeration*[25]). Para el ejemplo anterior, la debilidad CSS es formalmente identicada en CWE como CWE-79 (*Improper Neutralization of Input During Web Page Generation*).

Las vulnerabilidades poseen diferentes niveles severidad dependiendo del daño potencial que podría en caso de que la misma sea utilizada para perpetrar un ataque. Dicha severidad es típicamente definida mediante el mecanismo CVSS, el cual asigna a las vulnerabilidades una puntuación que va del 1 (baja severidad) al 10 (alta severidad). Las vulnerabilidades podrían ser desconocidas por los *proveedores* (en inglés, *vendors*) del software, en cuyo caso son referidas como vulnerabilidades del día cero. En contrapartida, existen vulnerabilidades de público conocimiento que son formalmente reportadas a organizaciones encargadas de gestionarlas y socializarlas, tales como Mitre y la NVD. Las vulnerabilidades son típicamente individuadas mediante un identificador, tal como CVE-2014-0160. En particular, este identificador hace referencia a la vulnerabilidad conocida por el nombre de Heartbleed y la cual ha afectado a la libreria OpenSSL utilizada para propósitos de comunicación segura en redes de ordenadores. En base a la métrica de severidad CVSS, Heartbleed ha recibido un puntaje de 7.5 (alta severidad).[26]

Un *exploit* es un código, conjunto de datos o serie de comandos que explota (aprovecha) una vulnerabilidad existente en un software para forzar un comportamiento no previsto o indeseado en el mismo. Los exploits son típicamente utilizados para perpetrar *ataques* al software, los cuales podrían ser ejecutados por personas u organizaciones malintencionadas, por ejemplo, para propósitos de denegación de servicios o escalada de privilegios. Los *parches* (de seguridad), por su parte, consisten en modificaciones en el software que permiten eliminar vulnerabilidades o mitigar los efectos que las mismas podrían causar en el software.

Este artículo se enfoca primordialmente en las vulnerabilidades del software. Sin embargo, es de suma importancia tener presente los conceptos y terminología presentados en esta sección, dado que la información sobre vulnerabilidades viene típicamente enriquecida con detalles adicionales sobre la debilidad a la que está asociada, los proveedores involucrados, los parches y *exploits* existentes, los posibles ataques perpetrados aprovechando la vulnerabilidad existente, entre otra información de gran utilidad para propósitos de la ciberseguridad.

[25]https://cwe.mitre.org
[26]https://nvd.nist.gov/vuln/detail/cve-2014-0160

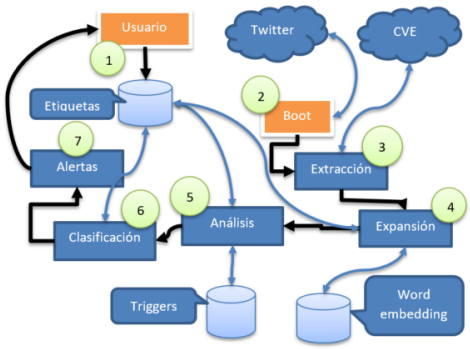

Fig. 3. Arquitectura de la solución propuesta para alertas tempranas sobre vulnerabilidades del software.

IV. Recuperación, Curación y Presentación de Información sobre Vulnerabilidades

La idea central en la cual se basa la solución propuesta en este artículo consiste en mantener al usuario al tanto de las vulnerabilidades existentes que afecten al entorno tecnológico de su interés. Abogamos por la posibilidad de que los usuarios puedan configurar preferencias sobre su entorno tecnológico, de modo a obtener alertas personalizadas. Para lograr este objetivo, nos basamos en técnicas de extración de información Web y recuperación de la información a partir de redes sociales, utilizando técnicas de expansión de consultas (en inglés, *query expansion*) [19]. La presentación de la información es realizada utilizando mecanismos basados en *timelines* [20] junto con el etiquetado automático e inteligente [6] basado en técnicas de *word embeddings* [5], mientras que la entrega de alertas la realizamos utilizando el paradigma de *push notifications* [21].

La Figura 3 presenta la arquitectura de la solución propuesta en este artículo para el acceso a alertas tempranas sobre vulnerabilidades del software. Esta solución permite a los usuarios finales la definición de preferencias sobre los productos y *vendors* que se corresponden con el entorno tecnlógico de interés del usuario. En base a dichas preferencias, nuestro sistema de alerta temprana permite identificar información relevante publicada en Twitter y fuentes oficiales de vulnerabilidades (CVE) y presentarlas convenientemente al usuario.

En la Figura 3, el *Usuario* ① define primeramente los productos (p.ej., SQL Server) y *vendors* (p.ej., Microsoft) de su interés en base a su entorno tecnológico utilizando etiquetas. Dichas preferencias son configuradas por el usuario utilizando la interfaz de usuario presentada en la Figura 4. Dichas etiquetas son utilizadas para la construcción de consultas (en inglés, *queries*) a ser utilizadas para la recuperación de información sobre vulnerabilidades. Debido a que tanto los productos como los *vendors* son expresados de múltiples formas (p.ej., “Microsoft Internet Explorer” es también expresado mediante las siglas “IE” en la jerga de las TIC), es

importante contar con la capacidad de recuperar información relevante inpedendientemente de la terminología utilizada para representar dicha información.

Para lidiar con esta variedad terminológica, el componente de *Expansión* ④ permite extender las etiquetas de preferencia del usuario mediante la utilización de técnicas de *word wmbedding* [5], las cuales permiten la representación de palabras en un espacio vectorial de manera tal que la palabras semánticamente similares se encuentren cercanas unas a otras (en este trabajo, utilizamos *word embeddings* entrenados para el dominio de la informática y la ciberseguridad, como detallamos en la siguiente sección de este artículo). Por ejemplo, si el usuario selecciona la etiqueta "Microsoft Internet Explorer", el componente de expansión genera palabras adicionales y relacionadas para enriquecer la consulta, tales como "Internet Explorer", "Explorer" y "IE". La utilización de consultas expandidas de esta manera permite el aumento del *recall* al momento de recuperar información relacionada a las preferencias del usuario [4].

Una vez expandidas las consultas de la manera indicada en el párrafo anterior, el componente de *Extracción* ③, en conjunción con el componente *Boot* ②, realizan las consultas pertinentes tanto a Twitter (mediante sus correspondientes API) como a CVE (utilizando ténicas de extracción de datos Web [12]). La información recuperada de estas fuentes es posteriormente procesada por los componentes de *Análisis* ⑤ y *Clasificación* ⑥, los cuales permiten categorizar y etiquetar la información recuperada para presentarlas al usuario final en forma de *Alertas* ⑦ personalizadas. Utilizamos el concepto de *etiquetado inteligente* [6], el cual es generado utilizando palabras semánticamente relacionadas en el espacio vectorial generado utilizando técnicas de *word embedding* [5]. Dichas alertas son presentadas al usuario utilizando el paradigma de *timelines* [20] (ver Figura 6). Hemos optado por esta opción debido a que el mismo es considerado apropiado para la presentación de información naturalmente asociada al aspecto temporal (en el dominio de la ciberseguridad, los reportes de vulnerabilidades de están naturalmente asociados a una fecha). Adicionalmente, *timelines* es un paradigma de presentación de la información actualmente muy utilizado y comprendido por los usuarios, particularmente debido a la exposición frecuente de las personas a dicho paradigma en la utilización de aplicaciones para redes sociales (p.ej., Twitter, Facebook, Instagram).

V. Implementación y Evaluación

A. Implementación de la Propuesta

La solución propuesta en este artículo es implementada como una application Web utilizando un *stack* de tecnologías apropiadas para el efecto. En esta sección, proveemos los detalles tecnológicos de la implementación de nuestra propuesta.

Fig. 4. Interfaz de usuario para la definición de preferencias del usuario en relación a productos y *vendors* de su interés.

Fig. 5. Visualización de la front-end para la definición de etiquetas de caracterización del Software.

Para la implementación del *back end* utilizamos PHP versión 7 y el motor de datos MySQL versión 5. El acceso a la información publicada en Twitter lo hacemos mediante sus API públicamente disponibles. Realizamos consultas a varias cuentas de usuarios verificados y cuyos avisos son considerados confiables. Recordamos que utilizamos esta red social como principal fuente de información sobre potenciales vulnerabilidades del día cero. En contrapartida, utilizamos los repositorios de NVD como principal fuente para la extracción de información sobre vulnerabilidades formalmente reportadas y documentadas. El acceso a dicha información la realizamos utilizando herramientas tradicionales de extración de datos Web [12], en particular, la librería Simple HTML DOM.[27]

Para propósitos de extracción de datos (particularmente, a partir de la red social Twitter), contamos con una lista de más de 120 palabras semillas [19] que permiten detectar la presencia de información relacionada a vulnerabilidades. Ejemplo de estas palabras semillas incluyen "exploit", "vulnerability", "attack", entre otras semillas. Adicionalmente, contamos también con palabras bloqueadas que frecuentemente resultan en falsos positivos [4] (p.ej., "podcast", "seminar", "webinar", "article"). De manera a lidiar con las inflexiones del lenguaje (verbos conjugados, palabras en plural, gerundios, etc.), utilizamos la raíz de las palabras arriba mencionadas, basándonos en el recurso públicamente disponible del Diccionario de

[27]https://simplehtmldom.sourceforge.io/

Fig. 6. Interfaz de usuario para la presentación de alertas sobre vulnerabilidades.

Cambridge.[28] La utilización de las palabras raíz nos permite aumentar el *recall* en la recuperación de información [4], lo cual contribuye a disminuir la omisión de información relevante relativa a las vulnerabilidades del software.

Como adelantabamos en la sección anterior, la expansión de consultas [19] es realizada utilizando técnicas de *word embedding*. En particular, para este trabajo, hemos producido nuestra propia implementación de la arquitectura de *word2vec* [5] en su variedad de *skip-gram* utilizando el lenguaje PHP y utilizando como corpus información textual del ámbito de la ciberseguridad. La utilización de *word embeddings* entrenados a partir corpus específicos del dominio (en nuestro caso, ciberseguridad) han demostrado ser más performantes respecto a la utilización de corpus de índole general (p.ej., Wikipedia) [10], [22]. Esta implementación forma parte del componente de *Expansión* de la arquitectura presentada en la Figura 3.

En cuanto a la implementación del *front end*, hemos utilizado los estándares HTML5[29] y CSS.[30] Los componentes de la interfaz de usuario fueron modelados con la librería Bootstrap,[31] la cual nos ha permitido obtener una alta y estandarizada interactividad para la construcción de una interfaz basada en el concepto de Responsive Web. Los procesos asíncronos fueron implementados utilizando Ajax[32] y JQuery,[33] mientras que JSON fue utilizado como mecanismo principal de representación de datos para la comunicación vía API. Las Figuras 4, 5 y 6 son capturas de pantallas que ejemplifican las interfaces de usuarios utilizadas en nuestra solución para definición de preferencias del usuario, definición de etiquetas y presentación de alertas sobre vulnerabilidades, respectivamente.

[28]https://dictionary.cambridge.org
[29]https://es.wikipedia.org/wiki/HTML5
[30]https://www.w3.org/wiki/Es/CSS
[31]https://getbootstrap.com/
[32]https://es.wikipedia.org/wiki/AJAX
[33]https://es.wikipedia.org/wiki/JQuery

B. Evaluación de la Propuesta

Durante la segunda mitad del mes de abril de 2021, hemos realizado la evaluación de nuestra propuesta para identificar los beneficios y limitaciones de la misma. Para el efecto, hemos llevado a cabo entrevistas contextuales [23] y pruebas de usabilidad [24] del sistema involucrando 5 personas (promedio de edad = 28 años) del ámbito de las TIC y la ciberseguridad. Este número de participantes es considerado aceptable para este tipo de estudios [24]. Los participantes fueron invitados y han proveído su consentimiento para participar en el estudio mediante correos electrónicos. Las sesiones con cada usuario fueron llevadas a cabo mediante video llamadas, en forma indepediente, con una duración de 1 hora. La participación fue voluntaria y no remunerada. El protocolo utilizado para el estudio fue el siguiente: (i) el investigador introduce el estudio y el propósito del mismo al participante; (ii) el investigador realiza una demostración de la herramienta para familiarizar al participante con la misma; (iii) el investigador provee una serie de tareas a ser realizada con la herramienta por el participante, quien las ejecuta bajo la observación del investigador; (iv) se lleva cabo una entrevista semi-estructurada para obtener una retroalimentación del participante. El audio de toda la experiencia es registrado por parte del investigador para propósitos de análisis posteriores.

Para propósitos de esta evaluación, hemos construido una base de datos con vulnerabilidades recolectadas de las fuentes listadas en la Sección IV, a partir de la segunda mitad del mes de febrero de 2021. Las vulnerabilidades utilizadas en el estudio fueron seleccionadas en base a la realidad tecnológica de los participantes, incluyendo vulnerabilidades reportadas para Wordpress, PHP, Windows, Linux, entre otras tecnologías. Para este estudio se consideraron un total aproximado de 2070 vulnerabilidades.

La entrevista semi-structurada fue realizada en base a las siguientes temáticas: (i) preguntas generales sobre percepciones y sentimientos relativos a la experiencia; (ii) preguntas sobre las percepciones en cuanto a las tareas realizadas con la herramienta; (iii) preguntas sobre la herramienta en sí. A continuación, proveemos los resultados obtenidos por este estudio.

Percepciones y sentimientos sobre la experiencia

En líneas generales, los usuarios declararon haber completado el estudio sin signos de agotamiento. Un total de 3 usuarios declaron que los ejercicios realizados en esta experiencia incidieron positivamente en sus habilidades y conocimientos sobre vulnerabilidades (p.ej., aprendieron nuevos conceptos), mientras que los restantes 2 mencionaron que la experiencia no tuvo incidencia en dichos aspectos, pero que la herramienta en sí podría ser de utilidad para sus actividades diarias.

Se pudo observar una lenta comprensión de las op-

ciones/funcionalidades del sistema por parte de los participantes, pero a medida que iban interactuando con la herramienta, se ha notado una rápida mejora en cuanto a la captación de la dinámica y las funcionalidades proveídas por la herramienta. En este sentido, un total de 3 participantes reconocieron explícitamente que todo sistema nuevo tiene su curva de aprendizaje, y por ende, es normal una cierta falta de compresión al inicio. En particular, un participante expresó cuanto sigue:

"El uso [correcto] de cualquier sistema implica [requiere] conocer el sistema, así que es normal perderse un poco con la herramienta en la primera sesión." (P3)

Los participantes P1 y P2 fueron los primeros en participar en el estudio, y en base a sus primeras experiencias y recomendaciones, el sistema fue corregido para las sesiones con los siguientes participantes. Por ejemplo, el participante P2 sugirió mejorar la iconografía de la herramienta para evitar confusiones. Como resultado de las correcciones, se ha notado una mejoría en la experiencia de los participantes P3, P4 y P5.

Percepciones sobre las tareas realizadas con la herramienta

Los participantes fueron consultados si es que sintieron una falta de habilidades y/o conocimientos para realizar las tareas con la herramienta. Todos afirmaron de que no experimentaron dicha falta debido a la familiaridad de los participantes con el dominio y con la dinámica de la herramienta en sí. En cuanto a lo último, consideramos que los usuarios no encontraron mayores dificultades debido a la familiaridad de los mismos con el paradigma de *timeline* (p.ej., debido a la familiaridad con aplicaciones de redes sociales como Facebook y Twitter), así como el etiquetado y filtrado de información.

Al ser abordados sobre la cercanía de las tareas realizadas con la herramienta respecto al día a día de sus actividades laborales (en TIC y ciberseguridad), todos coincidieron enfáticamente de que las mismas son efectivamente cercanas a dichas actividades, particularmente para especialistas del área de ciberseguridad.

Los participantes fueron además consultados sobre el nivel de conocimiento sobre ciberseguridad que se requeriría para poder utilizar el sistema de manera efectiva. Un total de 2 participantes coincidieron en que la herramienta no está orientada a usuarios sin conocimientos del dominio específico de la ciberseguridad. Los restantes observaron que podría ser de utilidad en caso de que la herramienta sea enriquecida con conceptos y definiciones fundamentales que permitan una mejor comprensión del dominio para usuarios no expertos. En este sentido, uno de los participantes afirmó:

"Para que usuarios no técnicos puedan usar la herramienta, creo que debería haber más definición de conceptos y una explicación más amplia de cuestiones relacionadas al ámbito de [la] seguridad [...]" (P5)

Al ser abordados sobre la utilidad de poder consultar y discernir entre vulnerabilidades formalmente reportadas (CVE) y aquellas del día cero, todos los participantes coincidieron en que es una funcionalidad útil. Por ejemplo, uno de los participantes agregó:

"[...] los puntos altos [altamente positivos] del sistema son los filtros y como filtro básico está la opción de listado por CVE y/o 0-day, útiles según la necesidad, [...] los usaría en mi día a día." (P4)

Percepciones sobre la herramienta

Se destacó como *ventajosa* la utilización de la herramienta como apoyo para concentrar y tener organizados los avisos sobre las vulnerabilidades del software, en relación a otros mecanismos de acceso a información sobre vulnerabilidades (p.ej., buscadores y repositorios de vulnerabilidades), y como muy importante la *inmediatez* en la entrega de las alertas. Dos de los participantes proveyeron observaciones muy interesantes en ese sentido:

"En mi día a día utilizo muchas fuentes de información [sobre vulnerabilidades] y todo es muy abrumador, esta herramienta me facilitaría la vida en la obtención rápida de la información pues tener la información oportuna en el momento oportuno me da muchas ventajas técnicamente hablando." (P4)

"En mi experiencia, cuándo recibíamos algún aviso o nos enterabamos por cualquier medio de alguna vulnerabilidad, en ese momento empezabamos a buscar en cualquier lugar [fuente], descentralizadamente, y en general, no siempre teníamos el tiempo suficiente para invertir en buscar información [sobre vulnerabilidades]. Normalmente no teníamos protocolos ni buenas fuentes de alertas." (P5)

En las primeras sesiones con los participantes P1 y P2, se presentaron dificultades en la compresión en cuanto al modo de listar las vulnerabilidades. Luego de ajustes realizados en base a las recomendaciones proveídas por estos dos participantes, las sesiones siguientes con los participantes restantes han demostrado mejoras en cuanto a la comprensión de la presentación (listado) de la información.

En cuanto a la configuración de las preferencias tecnológicas del usuario en base a etiquetas, las cuales permiten recibir altertas de vulnerabilidades en base a dichas preferencias, la mayoría de los usuarios experimentaron dificultad en configurarlas. Sin embargo, les pareció acertada la decisión de permitir filtrar la información en base a las preferencias tecnológicas del usuario, de modo a poder alivianar la carga cognitiva que implica estar proactivamente buscando nuevas vulnerabilidades. En ese sentido, un participante afirmó:

"Partiendo de la experiencia con algunos repositorios de información sobre vulnerabilidades y foros, en los que la

cantidad de información no relevante es muy grande, y que este sistema me permita aproximarme a mis preferencias tecnológicas y que me provea ya lo justo y necesario, me parece muy acertada y me permite relajarme y concentrarme en otras tareas." (P4)

Respecto al etiquetado automático de las vulnerabilidades, todos los usuarios coincidieron en que el mismo es realizado correctamente por la herramienta. El participante P2 resaltó que las etiquetas permiten comprender con un golpe de vista la categoría a la que pertenece la vulnerabilidad listada por la herramienta.

Cuando los participantes fueron consultados si es que han sido expuestos a conceptos o elementos que normalmente no tienen en cuenta en sus actividades diarias, 2 participantes resaltaron los conceptos de *niveles de gravedad* y *vulnerabilidades del día cero*. En particular, les pareció interesante que la herramienta contemple estos dos aspectos. Por otro lado, al consultarles sobre cómo compararían la herramienta en relación a soluciones similares, los 5 participantes declaron no conocer un sistema similar para el acceso a información sobre vulnerabilidades, enfatizando la relevancia de nuestra propuesta, y aclarando que sus fuentes primarias de información son los foros y repositorios de vulnerabilidades (los cuales no son directamente comparables a nuestro sistema de alertas tempranas de vulnerabilidades).

Al ser consultados sobre la funcionalidad de la herramienta para marcar una vulnerabilidad como solucionada o pendiente de seguimiento, todos los participantes coincidieron en afirmar que les resultaría muy útil. Incluso fueron más allá de esta funcionalidad y 4 participantes propusieron la posibilidad de compartir con la comunidad comentarios al respecto y la solución utilizada para abordar la vulnerabilidad. En este sentido, un participante sugirió cuanto sigue:

"Me parecen geniales y útiles las opciones complementarias [marcar una vulnerabilidad como solucionada o pendiente de ser solucionada], pero [además] me gustaría poder compartir con la comunidad comentarios y/o explicar los pasos que realicé en una vulnerabilidad solucionada." (P5)

Adicionalmente, un participante propuso la funcionalidad de poder asignar a otro usuario, mediante la herramienta, el trabajo de mitigar una vulnerabilidad:

"[...] me gustaría tener más información respecto al seguimiento en sí, y poder derivar a alguien el trabajo y saber cuándo lo solucionó." (P2)

En relación al uso del filtrado de información mediante etiquetas y nivel de gravedad, todos los participantes coincidieron en su relevancia y utilidad. Por ejemplo, un participante observó que:

"[...] puntos altos [altamente positivos] del sistema son las opciones de filtro, tanto para la búsqueda por etiquetas y más especialmente por grado de criticidad de la vulnerabilidad." (P4)

Dado que la herramienta permite el filtrado colaborativo de etiquetas utilizadas en las vulnerabilidades, permitiendo agregar/eliminar etiquetas, hemos también consultado al respecto a los usuarios. Todos los usuarios reconocieron su potencial utilidad, siempre y cuando sea correctamente utilizado. Además, todos expresaron su preocupación de un uso malintencionado (o por desconocimiento) de esta funcionalidad. Este es un problema ampliamente reconocido en sistemas de *crowdsourcing* [25], [26]. Por ejemplo, un participante afirmó:

" [...] particularmente me preocupa un tanto que cualquiera pueda eliminar las etiquetas, más que nada por que podría eliminarse alguna útil por desconocimiento o malintencionadamente, pero usándola bien me parece una buena opción colaborativa en la que nos beneficiemos todos en la comunidad. Podría haber una especie de usuarios colaboradores y que no todos puedan tener la opción de borrar [etiquetas] ." (P5)

La propuesta presentada arriba por el participante P5 se encuentra alineada con mecanismos utilizados en sistemas de *crowdsourcing* existentes. En particular, con la comunidad de preguntas y respuestas Stackoverflow,[34] donde usuarios con cierta reputación ganada en la comunidad acceden a permisos especiales de curación de contenido en la plataforma.

Finalmente, al consultar a los participantes si recomendaría a otros la utilización de la herramienta, todos los coincidieron que sí la recomendarían. Uno de los participantes agregó:

" [...] porque es útil y facil de usar para gente que esté en el ámbito de las vulnerabilidades, porque facilita tener a mano la información relevante. Rescatando todas sus funcionalidades: definición de preferencias, tagging de la información, filtros y avisos por email me parece una herramienta genial." (P1)

VI. Conclusión

Este artículo presenta una propuesta de alertas tempranas sobre vulnerabilidades formalmente documentadas en repositorios oficiales, así como también sobre potenciales vulnerabilidades del día cero detectadas a partir de las redes sociales. La propuesta apunta a abordar las necesidades crecientes y urgentes de las organizaciones de mantenerse al tanto de las debilidades del software, fenómeno siempre creciente particularmente en los últimos años. Nuestra propuesta aborda esta problemática mediante una solución que combina técnicas de recuperación de la información [4], expansión de consultas (*query expansion*) [19], categorización y etiquetado inteligente [6] de vulnerabilidades mediante técnicas de *word embeddings*

[34] https://stackoverflow.com

[5], y presentación de la información mediante mecanismos basados en *timelines* [20] y *push notifications* [21].

Los estudios realizados con usuarios representativos en forma de entrevistas contextuales [23] y pruebas de usabilidad [24] han revelado la viabilidad, utilidad y potencialidad de la solución propuesta. Los resultados arrojados por el estudio indican que éstos usuarios representativos encuentran en la solución propuesta una herramienta útil que emplearían en el día a día de sus operaciones diarias. Aspectos altamente positivos, en la perspectiva de los participantes, incluyen la posibilidad de establecer preferencias tecnológicas sobre las cuales recibir información sobre vulnerabilidades (permitiendo lidiar con la sobrecarga de información en este dominio), la precisión de la categorización y etiquetado de las vulnerabilidades, la posibilidad de filtrar información en base a criterios como vulnerabilidades del día cero y vulnerabilidades documentadas, criticidad de las vulnerabilidades, etc. En cuanto a las limitaciones, se ha visto con preocupación la posibilidad de que cualquier usuario pueda modificar las etiquetas (p.ej., elimándola). En consecuencia, se ha propuesto acceder a dicha funcionalidad basada en privilegios ganados por el usuario en base a su reputación.

Como trabajo futuro, proponemos incorporar las mejoras sugeridas por los participantes del estudio y desplegar la solución en un entorno organizacional real para su utilización en operaciones diarias, con propósitos de llevar a cabo un caso de estudio longitudinal sobre los beneficios y limitaciones de la propuesta en la creación de conciencia sobre las vulnerabilidades del software, la influencia de la propuesta en la mitigación de las mismas y el impacto en la mejora de la ciberseguridad de la organización.

AGRADECIMIENTO

El trabajo de los autores ha sido financiado parcialmente por PROCIENCIA y PRONII (Res. 148/2020) del CONACYT (Paraguay).

REFERENCES

[1] C. Rodriguez, S. Zamanirad, R. Nouri, K. Darabal, B. Benatallah, and M. Al-Banna, "Security vulnerability information service with natural language query support," in *International Conference on Advanced Information Systems Engineering*. Springer, 2019, pp. 497–512.

[2] S. Beattie, S. Arnold, C. Cowan, P. Wagle, C. Wright, and A. Shostack, "Timing the application of security patches for optimal uptime." in *LISA*, vol. 2, 2002, pp. 233–242.

[3] Y.-Y. Chang, P. Zavarsky, R. Ruhl, and D. Lindskog, "Trend analysis of the cve for software vulnerability management," in *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*. IEEE, 2011, pp. 1290–1293.

[4] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge university press, 2008.

[5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[6] J. Vig, S. Sen, and J. Riedl, "The tag genome: Encoding community knowledge to support novel interaction," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 2, no. 3, pp. 1–44, 2012.

[7] C. Sabottke, O. Suciu, and T. Dumitraș, "Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits," in *24th {USENIX} Security Symposium ({USENIX} Security 15)*, 2015, pp. 1041–1056.

[8] S. S. Alqahtani, E. E. Eghan, and J. Rilling, "Tracing known security vulnerabilities in software repositories–a semantic web enabled modeling approach," *Science of Computer Programming*, vol. 121, pp. 153–175, 2016.

[9] A. Joshi, R. Lal, T. Finin, and A. Joshi, "Extracting cybersecurity related linked data from text," in *2013 IEEE Seventh International Conference on Semantic Computing*. IEEE, 2013, pp. 252–259.

[10] S. Mumtaz, C. Rodriguez, B. Benatallah, M. Al-Banna, and S. Zamanirad, "Learning word representation for the cyber security vulnerability domain," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[11] S. A. Mokhov, J. Paquet, and M. Debbabi, "The use of nlp techniques in static code analysis to detect weaknesses and vulnerabilities," in *Canadian Conference on Artificial Intelligence*. Springer, 2014, pp. 326–332.

[12] E. Ferrara, P. De Meo, G. Fiumara, and R. Baumgartner, "Web data extraction, applications and techniques: A survey," *Knowledge-based systems*, vol. 70, pp. 301–323, 2014.

[13] L. Atymtayeva, K. Kozhakhmet, and G. Bortsova, "Building a knowledge base for expert system in information security," in *Soft computing in artificial intelligence*. Springer, 2014, pp. 57–76.

[14] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 851–860.

[15] D. E. Alexander, "Social media in disaster risk reduction and crisis management," *Science and engineering ethics*, vol. 20, no. 3, pp. 717–733, 2014.

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[18] A. Khazaei, M. Ghasemzadeh, and V. Derhami, "An automatic method for cvss score prediction using vulnerabilities description," *Journal of Intelligent & Fuzzy Systems*, vol. 30, no. 1, pp. 89–96, 2016.

[19] R. P. Khandpur, T. Ji, S. Jan, G. Wang, C.-T. Lu, and N. Ramakrishnan, "Crowdsourcing cybersecurity: Cyber attack detection using social media," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1049–1057.

[20] O. Alonso, R. Baeza-Yates, and M. Gertz, "Exploratory search using timelines," in *Proceedings of the ACM SIGCHI 2007 Workshop on Exploratory Search and HCI*, 2007, pp. 23–26.

[21] J. D. Falcão, J. Krebs, S. Kumar, and H. Erdogmus, "Openalerts: A software system to evaluate smart emergency alerts and notifications," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, 2018, pp. 1250–1255.

[22] S. Mumtaz, C. Rodriguez, and B. Benatallah, "Expert2vec: Experts representation in community question answering for question routing," in *International Conference on Advanced Information Systems Engineering*. Springer, 2019, pp. 213–229.

[23] K. Holtzblatt and S. Jones, "Conducting and analyzing a contextual interview (excerpt)," in *Readings in Human–Computer Interaction*. Elsevier, 1995, pp. 241–253.

[24] J. Lazar, J. H. Feng, and H. Hochheiser, *Research methods in human-computer interaction*. Morgan Kaufmann, 2017.

[25] M. Allahbakhsh, B. Benatallah, A. Ignjatovic, H. R. Motahari-Nezhad, E. Bertino, and S. Dustdar, "Quality control in crowdsourcing systems: Issues and directions," *IEEE Internet Computing*, vol. 17, no. 2, pp. 76–81, 2013.

[26] R. Sumi, T. Yasseri *et al.*, "Edit wars in wikipedia," in *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*. IEEE, 2011, pp. 724–727.