Estudio de las propiedades criptológicas de los Tokens No Fungibles Inteligentes para asegurar dispositivos de Internet de las Cosas

Jorge Eterovic; Marcelo Cipriano; Edith García; Luis Torres;

Instituto de Investigación en Ciencia y Tecnología
Dirección de Investigación Vicerrectorado de Investigación y Desarrollo.
Universidad del Salvador.
Lavalle 1854 – C1051AAB -Ciudad Autónoma de Buenos Aires - Argentina

{marcelo.cipriano; jorge.eterovic; edith.garcia; luisantonio.torres}@usal.edu.ar

RESUMEN

El Token No Fungible (NFT) es una solución creada para permitir representar objetos con cualidades únicas, irrepetibles e indivisibles dentro de una blockchain. Es un activo digital que representa objetos del mundo real como arte, música y videos. Se compran y venden en línea con criptomonedas y además se construyen utilizando el mismo tipo de programación, como Bitcoin o Ethereum.

Los datos únicos de los NFT facilitan la verificación de su propiedad y la transferencia de tokens entre propietarios. Un token no fungible inteligente (Smart NFT), podría representar dispositivos de Internet de las cosas (IoT), que son activos físicos inteligentes y lo identificaría como de propiedad de un usuario. Le asignaría una Dirección de Cuenta de Blockchain (BCA) para participar activamente en las transacciones pudiendo establecer canales de comunicación seguros entre propietarios y usuarios y operar de manera segura.

Con esta solución, los dispositivos IoT podrían probar no solo que su hardware es de confianza sino también su software, porque ejecutarían un arranque seguro y llevarían a cabo procesos de autenticación mutua con los propietarios y usuarios.

Este proyecto de investigación se centra en analizar las posibilidades de asegurar los datos de los dispositivos IoT usando los Smart NFT.

Palabras Clave:
Seguridad en IoT. Blockchain. Contrato Inteligente. Ethereum. Tokens No Fungibles.

CONTEXTO

El Vicerrectorado de Investigación y Desarrollo (VRID), perteneciente a la Universidad del Salvador (USAL), dicta las políticas referidas a la investigación, concibiéndola como un servicio a la comunidad y entendiendo que los nuevos conocimientos son la base de los cambios sociales y productivos. Con el impulso de las propias Unidades Académicas se han venido desarrollando acciones conducentes a concretar proyectos de investigación uni/multidisciplinarios, asociándolos a la docencia de grado y postgrado y vinculando este accionar, para potenciarlo, con otras instituciones académicas del ámbito nacional e internacional.

La Dirección de Investigación, dependiente del VRID, brinda soporte a las distintas Unidades de Investigación y a sus investigadores para el desarrollo de Proyectos y Programas de Investigación, nacionales e internacionales, como así también, apoyo y orientación de recursos para la investigación.

A ella pertenece el Instituto de Investigación en Ciencia y Tecnología (RR 576/12) en el cual se enmarca este proyecto denominado "Estudio de las propiedades criptológicas de los Tokens No Fungibles Inteligentes para asegurar dispositivos de Internet de las Cosas.", con una duración de 2 años (2023-

2024) y que ya ha sido evaluado y aprobado para su realización.

1. INTRODUCCIÓN

Si buscamos una tecnología que impactará y beneficiará nuestras vidas en los próximos años, es el Internet de las cosas. Los automóviles, electrodomésticos, teléfonos inteligentes, medidores de servicios públicos, sensores incorporados al cuerpo, indumentaria y casi cualquier cosa que podamos imaginar estarán conectados a Internet y serán accesibles desde cualquier parte del mundo [1]. La revolución que generará IoT será inigualable, algunos autores dicen que será similar a la construcción de carreteras y ferrocarriles que impulsaron la Revolución Industrial de los siglos XVIII al XIX [2], y será transversal a todos los sectores de la sociedad y todas las industrias, desde educación, salud, hogar y ciudad inteligente, hasta manufactura, minería, comercio, logística y vigilancia, solo por mencionar algunas [3].

Internet de las cosas implica directa o indirectamente la generación de cantidades significativas de información y un grupo dinámico de partes interesadas con distintos niveles de derechos de acceso a ella. Además, el alcance de la información relacionada con IoT variará según los requisitos del dominio de la aplicación y el contexto de los dispositivos.

Las aplicaciones de IoT involucran a muchas partes interesadas, con diferentes roles y funcionalidades que acceden a distintos tipos de información con distintos niveles de acceso, múltiples identidades y condiciones particulares de seguridad para cada una de ellas. Administrar todos estos activos de manera eficiente, segura e interoperable es un problema desafiante. Se analizará si la tecnología Blockchain y los contratos inteligentes de NFT pueden desempeñar un papel importante en este sentido [4].

Una cadena de bloques mantiene una colección, o libro mayor, de transacciones de manera descentralizada y distribuida. El libro mayor es inmutable e irreversible, lo que significa que las transacciones pasadas no pueden ser modificadas por ninguna entidad que registre transacciones en la Blockchain, y se comparte y sincroniza en todos los nodos participantes. De esta manera, la cadena de bloques garantiza que el libro mayor no puede ser manipulado, y que todos los datos que posee la Blockchain son confiables [5].

Una cadena de bloques puede ser pública [6] o estar restringida solo a usuarios autorizados [7]. La Blockchain se considera una forma democrática de mantener transacciones [8] y se prevé que proporcione mecanismos de seguridad novedosos, que contribuyan a la sostenibilidad de las aplicaciones de IoT y permitan nuevos modelos de confianza [9].

Un contrato inteligente es una aplicación distribuida que vive en la cadena de bloques [6]. Esta aplicación es, en esencia, una clase de lenguaje de programación con campos y métodos. Los usuarios pueden interactuar con los campos y métodos públicos de esta clase enviando transacciones a su dirección en la cadena de bloques.

Cada vez que un usuario interactúa con un contrato inteligente, todos los nodos de la red Blockchain ejecutan todas las operaciones de manera determinista y confiable y uno de estos nodos se selecciona para almacenar el resultado de la ejecución de los contratos, si corresponde, en la cadena de bloques. Los contratos inteligentes pueden verificar las identidades y firmas digitales de los usuarios de la Blockchain, realizar cálculos de propósito general e invocar a otros contratos [10].

El código de un contrato inteligente es inmutable y no puede ser modificado ni siquiera por su propietario [11]. Además, todas las transacciones enviadas a un contrato se registran en la cadena de bloques, por lo que es posible obtener todos los valores históricos de una variable del contrato.

2. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

La línea de investigación propone analizar las posibilidades de asegurar los datos de los

dispositivos de Internet de las Cosas usando los Tokens No Fungibles Inteligentes.

Internet de las cosas (IoT) es el paradigma en el que cualquier cosa (dispositivos, objetos, sujetos, etc.) puede interconectarse a través de Internet con la capacidad de interactuar, recopilar, procesar y compartir datos de manera inteligente. El rápido crecimiento en la cantidad de cosas conectadas a Internet ha llevado a la necesidad de buscar soluciones de seguridad.

La mayoría de las soluciones existentes se centran en el manejo y el intercambio seguro de datos, afrontando riesgos que van desde el uso de datos personales sin el consentimiento o conocimiento del propietario hasta el acceso o manipulación de datos por parte de terceros no autorizados [1].

Las tecnologías Blockchain se han utilizado recientemente con IoT ya que proporcionan una cadena de bloques de datos distribuida y criptográficamente segura, lo que permite una trazabilidad de los datos inmutable, que garantiza la propiedad de los mismos y la privacidad del usuario [2]. Cada bloque de una cadena de bloques se identifica unívocamente y se vincula con el anterior mediante una función hash criptográfica. Se agrega un nuevo bloque a la cadena si los participantes en la cadena de bloques con el rol de mineros demuestran que el nuevo bloque es seguro y la mayoría de los mineros lo validan, aplicando un algoritmo de consenso.

Si un atacante intenta cambiar los datos de un bloque, por ejemplo, los datos capturados por los sensores de un dispositivo IoT, el hash del bloque cambia y los datos del siguiente bloque no coinciden. El atacante tendría que cambiar todos los bloques posteriores de la cadena para no ser detectado [3].

Los datos en los bloques pueden ser datos capturados por los dispositivos IoT, así como datos sobre transacciones y los participantes involucrados en las transacciones. El intercambio de moneda digital o activos genéricos generalmente involucra transacciones entre participantes de blockchain. En cadenas de bloques públicas, como Ethereum, los acuerdos entre los participantes se formalizan a través de contratos inteligentes representados por scripts.

Estos scripts son validados como parte de las transacciones por el algoritmo de consenso y, por lo tanto, una vez validados, los contratos inteligentes son inviolables como los demás datos registrados en la cadena de bloques [4]. Además, se aplican protocolos para lograr la ejecución segura de los contratos inteligentes [5]. Todos los participantes ejecutan el código de un contrato inteligente de la misma manera y pueden comprobar si se cumplen las condiciones establecidas por el mismo.

Por lo tanto, además de la integridad, blockchain permite la transparencia porque cualquier participante puede consultar los datos registrados, incluidos los contratos inteligentes. Esto se ha aprovechado para integrar blockchain con la gestión de la cadena de suministro, lo que, en el caso del ecosistema IoT, permite garantizar el origen y la confiabilidad de los datos sin necesidad de intermediarios [6]. Los dispositivos IoT pueden ser totalmente autónomos para intercambiar directamente sus datos con terceros a través de contratos inteligentes que rigen todas las políticas de comercio de datos y derechos de propiedad [7].

La información de propiedad se puede incluir en contratos inteligentes por medio de tokens. Dado que un token es la representación digital de un activo en la cadena de bloques, existen dos tipos principales de tokens, tokens fungibles y no fungibles (NFT), según el activo representado.

Los tokens fungibles son tokens idénticos e intercambiables, como una moneda fiduciaria, que permiten transacciones de contabilidad y facturación. Los tokens no fungibles son tokens únicos y no intercambiables, como instrumentos notariales o coleccionables de obras de arte, que permiten la trazabilidad de posesiones físicas o no físicas únicas. Un ejemplo de posesión no física en el contexto de IoT es la capacidad de acceder a recursos o servicios.

Un dispositivo IoT que actúe como proveedor de datos puede representar mediante NFT los recursos o servicios proporcionados. [8, 9]. Otro uso de los NFT es para representar productos físicos. Cada producto fabricado por un servicio en la nube puede representarse mediante un NFT, de modo que su precio, procedencia y propiedad pueden rastrearse mediante una cadena de bloques [10].

Aplicando una tokenización más fina, cada componente de un producto puede representarse mediante un NFT de modo que la creación de un producto a partir de sus componentes se registre en la cadena de bloques como la creación de un nuevo token a partir de los tokens que representan sus componentes. De esta forma, no solo se puede rastrear el origen sino también la posterior transformación de un producto [11].

La tecnología blockchain se aplica no solo para garantizar la confiabilidad de la fabricación del dispositivo de hardware, sino también para garantizar la seguridad de su software. De lo contrario, toda la seguridad falla [12]. En cuanto al software, se presta especial atención al arranque seguro del microcontrolador en el hardware del dispositivo IoT, cuyo objetivo es comprobar que el código a ejecutar es el esperado, y a la comunicación segura con propietarios y usuarios, que pueden llegar a cambiar el software del dispositivo. En éste trabajo de investigación usaremos la cadena de bloques Ethereum ya que es una de las cadenas de bloques públicas más extendidas.

En la comunidad Ethereum, se desarrollaron los estándares ERC (Ethereum Request for Comments) [13], el estándar ERC-20 define los tokens fungibles y el estándar ERC-721 que define los tokens no fungibles. Un atributo importante de un ERC-721 NFT es su propietario, que se identifica únicamente por su Dirección de Cuenta de Cadena de Bloques (BCA). Un participante de la cadena de bloques interactúa con la cadena de bloques a través de una BCA, que se compone de un par de claves criptográficas (pública y privada) y una dirección única derivada de la clave pública. Además de un propietario, hay un identificador único que está asociado con un ERC-721 NFT.

En las propuestas donde los tokens ERC-721 representan bienes físicos, los identificadores de los tokens suelen estar relacionados con los atributos lógicos del producto, como su código de barras o QR [10], o a los datos de información del producto, como las instrucciones de eliminación y las fechas de caducidad [11]. El problema es que estos identificadores no están relacionados con algo intrínseco del producto, es decir, no son verdaderos identificadores de producto ya que pueden ser modificados, copiados o transferidos a otro producto.

Para evitar este problema, se propusieron los anclajes criptográficos como un vínculo estrecho entre los dominios digital y físico [14]. Las Funciones Físicas no Clonables (PUF) de silicio, que representan las variables del proceso de fabricación de los semiconductores [15], pueden tomarse como anclajes criptográficos para productos electrónicos, como los dispositivos IoT.

El enlace de ERC-721 NFT con dispositivos IoT físicos que usan PUF se propuso en un trabajo presentado en la Conferencia Internacional sobre Criptografía Aplicada y Seguridad de Redes realizada en Alemania en el año 2020 [16]. Ese trabajo incluyó, como un nuevo atributo en ERC-721 NFT, la dirección de cuenta de blockchain (BCA) asociada con el dispositivo IoT. Se propuso el uso de PUF en el hardware del dispositivo IoT para reconstruir la clave privada de la que se deriva la dirección BCA del dispositivo en su token asociado.

Además, ese trabajo también incluyó, como un nuevo atributo en ERC-721 NFT, la dirección de la cuenta blockchain (BCA) asociada con el usuario del dispositivo IoT, de modo que no solo se puede rastrear la propiedad sino también el uso del dispositivo mediante la cadena de bloques con el mismo token.

Sin embargo, ese trabajo no permitió detectar si el dispositivo no está funcionando correctamente, si el vínculo entre el dispositivo y el NFT está roto, o si el compromiso con el

propietario y el usuario se pierde en algún momento. Para evitar estas fallas de seguridad, un trabajo reciente presenta un nuevo NFT, llamado NFT inteligente (Smart NFT). [17].

Este proyecto de investigación se centra en la búsqueda, estudio y análisis de distintas posibilidades de combinar los dispositivos de Internet de las cosas (IoT) con las tecnologías blockchain de NFT para hacer que los dispositivos IoT sean seguros, desde el punto de vista del hardware y del software, de modo que los datos que proporcionan sean confiables.

3. RESULTADOS OBTENIDOS/ESPERADOS.

El Objetivo General del proyecto de investigación es analizar las distintas posibilidades de asegurar los datos de los dispositivos de Internet de las cosas (IoT) usando la tecnología blockchain de los Tokens No Fungibles (NFT).

Los Objetivos Específicos son relevar y Estudiar en detalle la tecnología Blockchain de los Tokens No Fungibles; analizar las propuestas de soluciones que garanticen la confiabilidad del hardware y del software de los dispositivos IoT desde su fabricación hasta la operación del usuario final; analizar como los tokens no fungibles inteligentes (Smart NFT), podrían representar dispositivos de Internet de las cosas (IoT) y proponer soluciones usando Smart NFT para asegurar los dispositivos IoT.

4. FORMACIÓN DE RECURSOS HUMANOS.

El equipo de investigadores pertenece al cuerpo docente de Tecnologías Aplicadas de la Facultad de Ingeniería, específicamente al área de Seguridad Informática, de la Universidad del Salvador.

A este proyecto, se incorporaron tres docentes investigadores con amplia trayectoria académica y un docente investigador con muchos años de desempeño en la industria de TI.

Esto redundará en un aumento del activo académico e investigativo representado por su cuerpo de docentes investigadores, como así también conformará las bases para la investigación a futuro desarrollando una línea de trabajo en la Facultad de Ingeniería.

5. BIBLIOGRAFÍA.

[1] Pal, S.; Rabehaja, T.; Mukhopadhyay, S. Requisitos de seguridad para Internet de las cosas: un enfoque sistemático. Sensores 2020, 20, 5897.

[2] Novo, O. Blockchain se encuentra con IoT: una arquitectura para la gestión de acceso escalable en IoT. IEEE Internet Things J. 2018, 5, 1184–1195.

[3] Dhananjay, S.; Jong-Hoon, K.; Madhusudan, S. Tecnologías de cadena de bloques; Springer Nature: Nueva York, NY, EE. UU., 2020.

[4] Longo, R.; Podda, AS; Saia, R. Análisis de un protocolo de consenso para extender subcadenas consistentes en la cadena de bloques de Bitcoin. Cómputo 2020, 8, 67.

[5] Sol, T.; Yu, W. Un marco de verificación formal para cuestiones de seguridad de los contratos inteligentes de Blockchain. Electronica 2020, 9, 255.

[6] Al-Rakhami, MS; Al-Mashari, M. Un modelo de confianza basado en blockchain para la gestión de la cadena de suministro de Internet de las cosas. Sensores 2021, 21, 1759.

[7] Nawaz, A.; Peña Queralta, J.; Guan, J.; Awais, M.; Gia, TN; Bashir, Ak; Kan, H.; Westerlund, T. Edge Computing para asegurar la propiedad de datos de IoT y el comercio con Ethereum Blockchain. Sensores 2020, 20, 3965.

[8] Omar, AS; Basor, O. Enfoque de tokens no fungibles basado en capacidad para un marco AAA descentralizado en IoT. Ciberseguridad de cadena de bloques. Confianza. priv. 2020, 79, 7–31.

[9] Fotiou, N.; Pittaras, I.; Siris, VA; Siris, VA; Voulgaris, S.; Polyzos, autorización de GC

OAuth 2.0 usando tokens basados en blockchain. arXiv 2020, arXiv:2001.10461.

[10] Hasan, M.; Starly, B. Arquitectura de plataforma descentralizada de fabricación en la nube como servicio (CMaaS) con activos digitales configurables. J. Manuf. sist. 2020, 56, 157–174.

[11] Westerkamp, M.; Víctor, F.; Küpper, A. Seguimiento de los procesos de fabricación utilizando composiciones de fichas basadas en blockchain. Dígito. común Neto. 2020, 6, 167–176.

[12] Jesús, V. Raíces de confianza mejoradas con Blockchain. En Actas de la Conferencia internacional IEEE sobre redes y comunicaciones inteligentes (SmartNets), Yasmine Hammamet, Túnez, 16 y 17 de noviembre de 2018; pp. 1 a 7.

[13] Propuestas de mejora de Ethereum. Disponible en línea: https://eips.ethereum.org/erc (consultado en abril de 2022).

[14] Balagurusamy, VS; Cabral, C.; Coomaraswamy, S.; Coomaraswamy, S.; Delamarche, E.; Dillenberger, D.; Friedman, D.; Gökce, O.; Hinds, N.; Jelito, J.; et al. Anclajes criptográficos. IBM J.Res. desarrollo 2019, 63, 4:1–4:12.

[15] Gao, Y.; Al-Sarawi, SF; Abbott, D. Funciones físicas no clonables. Nat. Electrón. 2020, 3, 81–91.

[16] Arcenegui, J.; Arjona, R.; Baturone, I. Gestión segura de dispositivos IoT basada en tokens no fungibles de cadena de bloques y funciones físicas no clonables. En Actas de la Conferencia Internacional sobre Criptografía Aplicada y Seguridad de Redes; Springer: Berlín/Heidelberg, Alemania, 2020; Volumen 12418, págs. 24–40.

[17] Javier Arcenegui; Rosario Arjona; Roberto Román; Iluminada Baturone. Secure Combination of IoT and Blockchain by Physically Binding IoT Devices to Smart Non-Fungible Tokens Using PUFs. Sensors 2021, 21(9), 3119.

Análisis de las Herramientas para Realizar Pruebas Estáticas de Seguridad de las Aplicaciones

Jorge Eterovic; Valeria Silvestri; Andrea Vera; Martin Zeballos; Alesio Esteban Sinopoli
Departamento de Ingeniería e Investigaciones Tecnológicas Universidad Nacional de La Matanza
Florencio Varela 1903 (B1754JEC), San Justo, (5411) 4480-8900

{eterovic; vsilvestri; avera; mzeballos}@unlam.edu.ar;
asinopoli@alumno.unlam.edu.ar

RESUMEN

Las pruebas estáticas de seguridad de las aplicaciones que se utilizan para proteger el software se denominan SAST (Static Application Security Testing) y consisten en la revisión automática del código fuente para identificar patrones vulnerables.

Las herramientas SAST permiten automatizar la detección de vulnerabilidades y se pueden integrar al sistema de CI/CD (integración continua / distribución continua) para que detecten vulnerabilidades en etapas tempranas del ciclo de vida. Esto ayuda al equipo de Seguridad de las Aplicaciones a implementar un ciclo de vida del desarrollo de software seguro.

CI/CD es un método para distribuir las aplicaciones a los clientes mediante el uso de la automatización en las etapas del desarrollo de las aplicaciones. En este contexto, cualquier equipo de Seguridad de las Aplicaciones se enfrentará al desafío de automatizar los chequeos de seguridad y encontrará la solución en herramientas SAST.

Integrar una herramienta SAST al proceso de CI/CD permite detectar vulnerabilidades en la etapa de desarrollo, en vez de esperar a la etapa de prueba o que se detecten directamente en producción.

Hay disponibles herramientas SAST gratuitas para los repositorios open-source y pagas para los repositorios privados. Algunas son open-source, otras usan un motor privado pero las reglas son open-source y algunas pocas son totalmente privadas.

Este proyecto de investigación propone hacer un análisis de estas herramientas SAST y aportar los resultados obtenidos a la comunidad open-source para mejorar la seguridad de los repositorios de proyectos de desarrollo de software que las utilizan.

Palabras Clave: *Herramienta SAST; Detección de Vulnerabilidades; Análisis Estático; Seguridad de las Aplicaciones..*

CONTEXTO

Este proyecto de investigación se desarrolla en el marco de un Programa de Incentivos a Docentes Investigadores de la Secretaría de Políticas Universitarias (PROINCE) del Ministerio de Educación, y se ejecuta en el Departamento de Ingeniería e Investigaciones Tecnológicas de la Universidad Nacional de La Matanza.

El proyecto es financiado por el propio Departamento y es del tipo investigación aplicada. El mismo propone hacer un análisis de las herramientas SAST y aportar los resultados obtenidos a la comunidad open-source. Los trabajos de campo y relevamientos realizados aportaron información valiosa y sirvieron como base para el presente trabajo.

1. INTRODUCCIÓN

Las herramientas SAST permiten automatizar la detección de vulnerabilidades y se pueden integrar al sistema de CI/CD para que detecten vulnerabilidades en etapas tempranas del ciclo de vida del desarrollo del software. Esto ayuda al equipo de Seguridad de las Aplicaciones a implementar un ciclo de vida seguro.

CI/CD es un método para distribuir las aplicaciones a los clientes mediante el uso de la automatización en las etapas del desarrollo de las aplicaciones. Los principales conceptos que se le atribuyen son la integración, la distribución y la implementación continuas. Se trata de una solución para los problemas que la integración de código nuevo puede generar a los equipos de desarrollo y de operaciones.

El proceso de integración y distribución continuas incorpora la automatización y la supervisión permanentes en todo el ciclo de vida de las aplicaciones, desde las etapas de integración y prueba hasta las de distribución e implementación. En este contexto, cualquier equipo de Seguridad de las Aplicaciones se enfrentará al desafío de automatizar los chequeos de seguridad y encontrará la solución en herramientas SAST.

Integrar una herramienta SAST al proceso de CI/CD permite detectar vulnerabilidades en la etapa de desarrollo, en vez de esperar a la etapa de prueba o que se detecten directamente en producción. Una vulnerabilidad en producción implica un riesgo constante, cuesta mucho esfuerzo de los expertos en seguridad detectarla y para los desarrolladores es difícil de corregir. En cambio, si se detecta durante la etapa de desarrollo, nunca generó un riesgo real, no requirió esfuerzo de personas de seguridad para detectarla y es mucho más fácil de corregir.

Hay muchas herramientas SAST y el análisis estático está muy vinculado al lenguaje de programación. Una herramienta puede analizar varios lenguajes, pero en realidad agregar un lenguaje nuevo a la herramienta, es desarrollar un producto nuevo. Es decir, una misma herramienta va a tener distintos grados de madurez, distintas características y distintas limitaciones según el lenguaje a analizar.

Un criterio de comparación entre distintas herramientas SAST es el grado de complejidad. Una herramienta simple se ejecuta rápidamente, soporta código que no compila y es fácil de aprender a usar, pero no es precisa, da muchos falsos positivos y falsos negativos incorregibles porque no maneja la información necesaria para refinar los resultados.

Por otro lado, una herramienta compleja se ejecuta más lentamente, tiene requisitos extras como por ejemplo que el código sea compilable y completo y lleva tiempo aprender a usarla, pero a su vez como maneja mucha más información y esta se puede usar para evitar falsos positivos y falsos negativos.

Hay bastante variedad entre las herramientas SAST. Algunas son open-source, otras usan un motor privado pero las reglas son open-source y algunas pocas son totalmente privadas. Algunas incluso distinguen su sistema de precios según el código a analizar, siendo gratuitas para los repositorios open-source y pagas para los repositorios privados. Y por último hay sistemas que simplemente se encargan de integrar y ofrecer varias herramientas. Por ejemplo, GitHub tiene una sección de escaneo de código denominada “Code Scanning” con muchas herramientas SAST.

2. LÍNEAS DE INVESTIGACION Y DESARROLLO

Se desarrollan a continuación los aspectos teóricos de este proyecto de investigación:

- DevOps
- DevSecOps
- Pruebas Estáticas de la Seguridad de la Aplicación (SAST)
- Integración Continua y Despliegue Continuo (CI/CD)

DevOps

Existen muchas definiciones diferentes de DevOps disponibles en libros, en artículos de revistas o en Internet. A raíz de esta disparidad en las definiciones, surgen diversos estudios que tratan de darle una descripción académica [1] [2]. Según estos estudios, podemos definir DevOps como una cultura que trata de aunar a los equipos de desarrollo y operaciones, basándose en una serie de principios y prácticas [2] que pretenden acelerar las entregas del producto mejorando el feedback de los clientes y la capacidad de reacción ante los cambios [3].

El término DevOps surge a finales de los 2000, en un contexto en el que las metodologías de desarrollo ágiles cada vez tomaban mayor relevancia en la industria del desarrollo de software. La velocidad a la que se desarrollaban nuevas características o se corregían bugs distaba mucho de la velocidad a la que se realizaban los despliegues de estos cambios, con lo que el ciclo

de desarrollo del software se veía ralentizado. Esta ralentización era resultado de la falta de comunicación existente entre el equipo de desarrollo y el de operaciones.

Fue Patrick Debois quien, en 2007, tras una experiencia frustrante trabajando en la migración de un gran centro de datos, se percató de cómo esta falta de comunicación entre desarrolladores y administradores de sistemas afectaba al flujo de trabajo [4]. En 2009, John Allspaw y Paul Hammond, ingenieros de Flickr, presentaron su charla "10 Deploys a Day: Dev and Ops Cooperation at Flickr" [5] donde propusieron integrar desarrollo y operaciones en un flujo automatizado. Patrick Debois, tras esta conferencia, decidió organizar una similar en Bélgica, a la que llamó DevOpsDays, de donde surge el término DevOps [6].

Un aspecto importante para seguir exitosamente la cultura DevOps es la automatización de procesos, al ser lo que permite mantener la agilidad durante el desarrollo del software. La importancia de la automatización de procesos ha hecho surgir una gran cantidad de herramientas que contemplan la construcción del software, la integración y el despliegue continuos, la gestión de logs y la monitorización [7].

DevSecOps

El crecimiento de las metodologías ágiles de desarrollo del software y la acogida de la cultura DevOps por parte de las organizaciones ha incrementado la velocidad a la que las aplicaciones reciben actualizaciones. Esto tiene grandes ventajas, pues permite tener feedback temprano del cliente para mejorar el producto desde las primeras fases del desarrollo, mejorando la adaptabilidad del producto con el entorno y permitiendo marcar la diferencia con la competencia gracias a la implementación de nuevas funcionalidades y la mejora del funcionamiento de las ya existentes [8]. Sin embargo, a veces esta agilidad se consigue a costa de sacrificar otros aspectos del producto final, como puede ser la seguridad [9]. Los estudios muestran que menos de un 20% de las compañías que siguen la cultura DevOps tienen en cuenta la seguridad como parte del ciclo de desarrollo del software [5].

DevSecOps surge para incluir la seguridad en DevOps, alineando los equipos de desarrollo, de operaciones y de seguridad durante todo el ciclo de desarrollo. Esto se consigue desplazando la seguridad a la izquierda, es decir, considerándola desde las primeras etapas del desarrollo [9]. De esta manera, al tenerla en cuenta desde el diseño de la aplicación, es posible realizar los controles de seguridad necesarios a lo largo del ciclo de desarrollo del software y automatizarlos para que sean rápidos, escalables y efectivos [10]. De esta manera se mantiene la agilidad en el desarrollo del software y se detectan desde fases tempranas los fallos de seguridad que, de llegar al cliente, conllevarían grandes pérdidas de tiempo y dinero.

Al igual que en DevOps, las herramientas juegan un papel de gran importancia. Existen una gran cantidad de herramientas para llevar a cabo DevSecOps. Se ha tomado como referencia la guía OWASP DevSecOps Guideline [11] para establecer los aspectos más importantes relativos a la seguridad: la detección de secretos, las Pruebas Estáticas de la Seguridad de la Aplicación (SAST), las Pruebas Dinámicas de la Seguridad de la Aplicación (DAST), el escaneo de la infraestructura y la comprobación del cumplimiento normativo.

Pruebas Estáticas de la Seguridad de la Aplicación (SAST)

Las pruebas estáticas de seguridad analizan el código fuente de la aplicación sin ejecutarlo, tratando así de encontrar vulnerabilidades o bugs [12]. Los análisis estáticos se pueden realizar de diferentes formas, desde las más sencillas y rápidas que contemplan sólo un análisis del código fuente en base al árbol sintáctico, hasta las más complejas que combinan diversas representaciones del código como grafos de control y flujo de datos para realizar un análisis semántico en busca de patrones vulnerables [12].

Los factores a tener en cuenta a la hora de decantarse por una herramienta u otra son la velocidad a la que se quiere realizar el análisis y la profundidad del mismo, pues a más profundidad, más se tardará en realizar y viceversa. Además, se ha de considerar el porcentaje de falsos positivos que puede señalar la herramienta y el lenguaje de programación de la aplicación. Las herramientas SAST ayudan a detectar vulnerabilidades como inyecciones SQL, cross-site scripting (XSS) y problemas de gestión de memoria, entre otras.

Algunos ejemplos de estas herramientas son: Semgrep, CodeSonar o CodeQL[13] [14].

Integración Continua y Despliegue Continuo (CI/CD)

La integración continua (CI) surge como una de las prácticas de la metodología ágil Extreme Programming (XP), en la cual se propone que los desarrolladores publiquen sus cambios varias veces al día en el repositorio de código. De esta forma pueden encontrarse problemas de compatibilidad entre estos cambios en etapas más tempranas del desarrollo, y se evitan complejos y largos procesos de integración en los días anteriores de la fecha de entrega del proyecto o del hito [15]. Gracias a estas ventajas, la integración continua se utiliza como práctica independiente de XP, respaldada por referentes en el mundo del desarrollo del software como Martin Fowler [16].

El despliegue continuo (CD) amplía las bases propuestas por la integración continua, proponiendo no solo la integración automatizada del código en el repositorio, sino también el despliegue automatizado del mismo en el entorno de producción [17]. La automatización del despliegue es especialmente beneficiosa cuando existen varios entornos en los que se ha de desplegar el software cuando se genera una nueva versión, y cuando este proceso de despliegue ocupa mucho tiempo [18]. Cabe clarificar las diferencias entre la entrega continua y el despliegue continuo, conceptos que en ocasiones se confunden. Mientras la entrega continua tiene como objetivo mantener siempre el software en un estado que permite su despliegue inmediato [19], el despliegue continuo implica el despliegue de las nuevas versiones del software de forma automática.

Los procesos automatizados de integración y de despliegue pueden tardar varios minutos en ejecutarse, ralentizando el flujo de desarrollo. Además, requieren de una gran coordinación por parte de los desarrolladores para evitar conflictos en los cambios y en el orden en que se realizan las integraciones. Por estos motivos, desde la metodología XP se propone el uso de un servidor dedicado a realizar las integraciones. Este servidor, denominado servidor de integración continua, asegura la realización de los procedimientos necesarios para integrar los nuevos cambios de manera ordenada y evitando conflictos [20] y su elección es clave para llevar a cabo con éxito estos procesos.

Existen varias alternativas en el mercado, siendo GitLab CI, Jenkins y GitHub Actions los servidores de integración continua más destacables. Tanto GitLab CI como GitHub Actions forman parte del ecosistema de los gestores de repositorios GitLab y GitHub respectivamente, por lo que son los que se tendrán en cuenta en este proyecto, evitando así la dependencia de una herramienta adicional para este propósito.

3. RESULTADOS OBTENIDOS/ESPERADOS

El objetivo general es analizar las herramientas open-source disponibles para realizar las pruebas estáticas de seguridad de las aplicaciones (SAST).

Los objetivos específicos son:

- Comparar las herramientas SAST open-source para determinar sus fortalezas y debilidades.
- Desarrollar casos de ejemplo para analizar qué vulnerabilidad encuentra cada herramienta SAST y establecer sus limitaciones. Esta tarea se realiza para cada lenguaje de programación.
- Analizar las vulnerabilidades para determinar si hay patrones detectables en el código o no.
- Clasificar las vulnerabilidades para asistir a los expertos de seguridad sobre cuándo usar herramientas SAST y cuándo no.
- Desarrollar una regla en una herramienta SAST para un lenguaje específico para detectar una vulnerabilidad determinada.

4. FORMACIÓN DE RECURSOS HUMANOS

El equipo de trabajo de este proyecto está formado por dos ingenieras y un licenciado en informática, un especialista en seguridad teleinformática y un alumno avanzado de la carrera. Este trabajo se desarrolló en el marco del proyecto de investigación: “Análisis de las Herramientas SAST”.

Dada la complejidad del desarrollo del proyecto de investigación, fue necesaria la colaboración de

varios expertos con amplia experiencia en la industria y la investigación académica.

5. BIBLIOGRAFIA

[1] De Franca, B. B. N., Jeronimo, H., & Travassos, G. H. (2016). Characterizing DevOps by Hearing Multiple Voices. Proceedings of the 30th Brazilian Symposium on Software Engineering - SBES '16. Published. https://doi.org/10.1145/2973839.2973845

[2] Jabbari, R., bin Ali, N., Petersen, K., & Tanveer, B. (2016). What is DevOps? Proceedings of the Scienti_c Workshop Proceedings of XP2016. Published. https://doi.org/10.1145/2962695.2962707

[3] Virani, M. (2015). Understanding DevOps & bridging the gap from continuous integration to continuous delivery. Fifth International Conference on the Innovative Computing Technology (INTECH 2015). Published. https://doi.org/10.1109/intech.2015.7173368

[4] Watts, S. (2019, 29 March). A Brief History of DevOps. BMC Blogs. https://www.bmc.com/blogs/ devops-history/

[5] Allspaw, J., & Hammond, P. (2009, Jun 22). 10+ Deployes per Day: Dev and Ops Cooperation at Flickr [Talk]. O'Reilly Velocity Conference, San Jose, California. https://www.youtube.com/ watch?v=LdOe18KhtT4

[6] Debois, P. (s. f.). About devopsdays. DevOpsDays. Recuperado 18 de enero de 2022, de https://devopsdays.org/about/

[7] Akshaya, H. L., Nisarga Jagadish, S., Bidya, J., & Veena, K. (2015). A Basic Introduction to DevOps Tools. International Journal of Computer Science and Information Technologies, 6(3). http://ijcsit.com/docs/Volume%206/vol6issue03/ijcsit2015060382.pdf

[8] Beck, K., Fowler, M., Martin, R. C., Beedle, M., Cockburn, A., Cunningham, W., Thomas, D., Mellor, S., Schwaber, K., Sutherland, J., Bennekum, A. V., Grenning, J., Highsmith, J., Hunt, A., Je_ries, R., Kern, J., & Marick, B. (2001, 13 February). Principios del Manifiesto Ágil. Agile Manifesto. http://agilemanifesto.org/iso/es/principles.html

[9] Shackleford, D. (2016, 8 March). A DevSecOps Playbook. SANS. https://www.sans.org/webcasts/devsecops-playbook-101472

[10] Amazon Web Services. (2016, 9 November). Introduction to DevSecOps on AWS. https://www.slideshare.net/AmazonWebServices/introduction-todevsecops-on-aws-68522874

[11] The OWASP Foundation. (s. f.). OWASP DevSecOps Guideline. OWASP. Recuperado 24 de enero de 2022, de https://owasp.org/www-projectdevsecops-guideline/

[12] Adkins, H., Beyer, B., Blankinship, P., Lewandowski, P., Oprea, A., & Stubble _eld, A. (2020). Building Secure and Reliable Systems: Best Practices for Designing, Implementing, and Maintaining Systems (Illustrated ed.). O'Reilly Media. https://sre.google/books/building-secure-reliable-systems/

[13] Peterson, J. (2020, 19 November). Software Composition Analysis Explained. WhiteSource. https://www.whitesourcesoftware.com/resources/blog/softwarecomposition-analysis/

[14] Weerasinghe, M. (2019, 24 December). NodeJS Security Tools – Manjula Weerasinghe. Medium. https://medium.com/@manjula.aw/nodejs-securitytools-de0d0c937ec0

[15] Wells, D. (1999). Continuous Integration. Extreme Programming. http://www.extremeprogramming.org/rules/integrateoften.html

[16] Fowler, M. (2000, 10 September). Continuous Integration (original version). martinfowler.com. https://www.martinfowler.com/articles/originalContinuousIntegration.html

[17] Rahman, A. A. U., Helms, E., Williams, L., & Parnin, C. (2015). Synthesizing Continuous Deployment Practices Used in Software Development. 2015 Agile Conference. Published. https://doi.org/10.1109/agile.2015.12

[18] Humble, J., Read, C., & North, D. (2006). The Deployment Production Line. AGILE 2006 (AGILE'06). Published. https://doi.org/10.1109/agile.2006.53

[19] Fowler, M. (2013, 30 May). Continuous Delivery. martinfowler.com. https://martinfowler.com/bliki/ ContinuousDelivery.html

[20] Wells, D. (1999). Dedicated Release Computer. Extreme Programming. http://www.extremeprogramming.org/rules/dedicated.html