Tácticas arquitectónicas de seguridad para un sistema de e-voting

Francisco Javier Zambrano-Santacruz[1]
Daniel Esteban Burbano[2]
Giovanni Albeiro Hernández-Pantoja[3]
Alexander Barón-Salazar[4]

Resumen

En una sociedad se debe buscar procesos electorales democráticos seguros y transparentes. Actualmente la democracia colombiana no brinda estas garantías. Esta es una de las razones por las que existe un abstencionismo alto. En los últimos años, el voto electrónico (*e-voting*) se presenta como alternativa para evitar fraude electoral; sin embargo, no es utilizado por problemas en la seguridad. El objetivo de esta investigación fue identificar tácticas arquitectónicas de seguridad utilizadas en la construcción de software para *e-voting* e implementarlas en un producto software. La metodología utilizada para identificar las tácticas fue la revisión sistemática de literatura. A partir de las tácticas identificadas se procedió a construir una solución software que implemente tácticas pertinentes para e-voting. Los resultados indican que las tácticas arquitectónicas de seguridad se orientan a detectar, resistir, reaccionar y recuperarse de ataques, junto con auditar, autenticar y establecer sesiones seguras. Los campos de mayor aplicación son la industria, la academia y el gobierno.

[1]Ingeniero de Sistemas. Integrante del grupo de investigación Galeras.net, Universidad de Nariño, Pasto, Nariño, Colombia. Correo electrónico: franciscozambrano@udenar.edu.co.
[2]Ingeniero de Sistemas. Integrante del grupo de investigación Galeras.net, Universidad de Nariño, Pasto, Nariño, Colombia. Correo electrónico: danielburbano19@udenar.edu.co.
[3]Magíster en Docencia Universitaria, Especialista en Gerencia Informática, Ingeniero de Sistemas. Docente asociado tiempo completo e integrante del grupo de investigación GISMAR, Universidad Mariana, Pasto, Nariño, Colombia. Correo electrónico: gihernandez@umariana.edu.co
[4]Doctor en Ingeniería, Magíster en Ingeniería Informática, Especialista en Ingeniería de Software, Especialista en Docencia Universitaria, Especialista en Desarrollo de Software, Ingeniero de Sistemas. Docente tiempo completo e integrante del grupo de investigación Galeras.Net, Universidad de Nariño, Pasto, Nariño, Colombia. Correo electrónico: abaron@udenar.edu.co

Con las tácticas identificadas se construyó Kybernan utilizando Scrum. Se concluye que las principales tácticas arquitectónicas de seguridad para *e-voting* se centran en detectar, resistir, reaccionar y recuperarse de los ataques. Kybernan es un software que soporta un proceso electoral e implementa ocho tácticas arquitectónicas de seguridad.

Palabras clave: E-voting; ingeniería de software; táctica arquitectónica; táctica arquitectónica de seguridad.

Architectural security tactics for an e-voting system

Abstract

Society must seek safe and transparent democratic electoral processes; currently, Colombian democracy does not offer these guarantees, generating as a consequence, a high level of abstention. In recent years, electronic voting has been presented as an alternative to avoid electoral fraud; however, it is not used, due to security problems. The objective of this research was to identify security architectural tactics used in the construction of e-voting software and implement them in a software product. The methodology used to identify the tactics was the systematic literature review. Based on the identified tactics, a software solution was built that implements pertinent tactics for e-voting. The results indicate that the architectural security tactics are geared towards detecting, resisting, reacting, and recovering from attacks, along with auditing, authenticating, and establishing secure sessions. The fields of greatest application are industry, academia, and government. With the identified tactics, Kybernan was built using Scrum. It is concluded that the main architectural security tactics for e-voting focus on detecting, resisting, reacting, and recovering from attacks. Kybernan is software that supports an electoral process and implements eight security architectural tactics.

Keywords: E-voting; Software Engineering; architectural tactic; architectural security tactic.

Táticas de segurança arquitetônica para um sistema de votação eletrônica

Resumo

A sociedade deve buscar processos eleitorais democráticos seguros e transparentes; atualmente, a democracia colombiana não oferece essas garantias, gerando como consequência um alto nível de abstenção. Nos últimos anos, o voto eletrônico (e-votação) tem se apresentado como uma alternativa para evitar fraudes eleitorais; no entanto, ele não é usado, devido a problemas de segurança. O objetivo desta pesquisa foi identificar táticas arquitetônicas de segurança utilizadas na construção de softwares de votação eletrônica e implementá-las em um produto de software. A revisão sistemática da literatura foi a metodologia utilizada para identificar as táticas. Com base nas táticas identificadas, foi construída uma solução de software que implementa táticas pertinentes para votação eletrônica. Os resultados indicam que as táticas de segurança arquitetônica são voltadas para detectar, resistir, reagir e se recuperar de ataques, junto com a auditoria, autenticação e estabelecimento de sessões seguras. Os campos de maior aplicação são indústria, academia e governo. Com as táticas identificadas, Kybernan foi construído usando Scrum. Conclui-se que as principais táticas de segurança arquitetônica para e-votação se concentram em detectar, resistir, reagir e se recuperar de ataques. Kybernan é um software que suporta um processo eleitoral e implementa oito táticas de arquitetura de segurança.

Palavras-chave: Votação eletrônica; Engenharia de software; tática arquitetônica; tática de segurança arquitetônica.

INTRODUCCIÓN

En la actualidad las instituciones y organizaciones colombianas continúan desarrollando procesos tradicionales de votación, es decir, votaciones democráticas manuales, se usa las boletas tradicionales de papel, las cuales deben ser marcadas y depositadas en una urna y posteriormente contabilizadas por los jurados de votación. El proceso electoral convencional no brinda las garantías necesarias, y siempre ha existido incertidumbre por parte del elector en cuestiones de alteración, privacidad y errores de conteo.

En este sentido, existen organizaciones que tratan de garantizar y velar por la transparencia de dichos procesos electorales (tradicionales) en Colombia, pero la historia ha mostrado que los resultados de las elecciones no son los más confiables, independientemente de la organización encargada de velar y dar garantías en las elecciones democrática. Esto radica en que el problema está enfocado en el sistema y no en las organizaciones encargadas de dar las garantías, puesto que el sistema tradicional es difícilmente auditable, lo cual genera que pueda presentarse alteración en los resultados electorales.

Otro de los inconvenientes que presenta el sistema actual de votación es que el reporte de resultados suele requerir mucho tiempo y el escrutinio de los votos lo hacen los jurados de votación, esto genera aún más desconfianza para el sistema, dado que el ser humano no está exento de cometer errores, teniendo en cuenta que las actividades que se desarrollan son completamente manuales. Además, actualmente no existe un proceso de verificación al momento en que el votante ingresa a la urna, esto deja una posibilidad para suplantación de identidad.

Cabe resaltar que estos inconvenientes se concluyen en una deserción de votantes masiva, por ejemplo, en las elecciones presidenciales de Colombia, llevadas a cabo el 28 de mayo 2018, el 46,62 % de la población no ejerció su derecho al voto, y del 100 % de los votos escrutados por la Registraduría Nacional, solo ejercieron su derecho al voto 19’636.714 personas de las 36’783.940 habilitadas (Alvarado, 2018). Además, en el año 2016, la abstención al voto del plebiscito por el proceso de paz fue de un 62 %, una cifra bastante alta que no se presentaba en los últimos 24 años en Colombia (BBC Mundo, 2016).

Por otra parte, la tasa de votos nulos en Colombia, en el año 2014, para las elecciones del Senado fue de 10,38 % y de 12,23 % para la Cámara de Representantes (Registraduría Nacional del Estado Civil, 2014). Estos índices se deben a la complejidad a la hora de votar en los tarjetones tradicionales de elección, porque hay saturación de información que confunde al elector.

Otra razón importante tiene que ver con la población de adultos mayores, que superan los 60 años en Colombia, la cual comprende un 11 % (Quiñones, 2017). Este grupo de personas se abstiene de votar, debido a la dificultad que les representa este proceso de elección.

Teniendo en cuenta que todos los ciudadanos tienen el derecho a elegir a sus representantes, se debe contar con un sistema que resuelva estas problemáticas e incentive a la comunidad a ejercer este derecho.

En los últimos años, el voto electrónico (*e-voting*) ha tomado fuerza en diferentes países como Brasil, Bélgica, India y Argentina (Estudiantes de Comunicación Social y Periodismo - Universidad los Libertadores, 2018); sin embargo, en Colombia han fracasado los intentos de implementación, ya que no existen las garantías necesarias en los tópicos de seguridad en sistemas *e-voting*.

Mientras que en muchos países se quiere implementar el voto electrónico, en naciones con democracias consolidadas, como Alemania, Finlandia, Holanda, Irlanda y el Reino Unido, argumentan que no lo hacen por problemas de seguridad (Estudiantes de Comunicación Social y Periodismo - Universidad los Libertadores, 2018), es decir, sostienen que el porcentaje de fraude puede ser muy alto. Cabe destacar que Holanda tuvo un sistema de votación electrónica, pero decidió volver al sistema de votación tradicional. El problema de base en Holanda no fue utilizar tecnología en entornos electorales, sino la falta de mantenimiento y actualización de los sistemas que utilizó para las elecciones (Ramos, 2017). En una auditoría de seguridad, las autoridades determinaron que tanto el software como las computadoras que se estaban utilizando para enviar esa información utilizaban un sistema operativo Windows XP y un software que no se había actualizado desde hace casi 10 años. Tampoco implementaba medidas criptográficas para proteger los datos que se transmitían, como firmas electrónicas o cifradas. Por lo tanto, sin mantenimiento de seguridad, desde hace años, y sin medidas de protección de los datos transmitidos, existe un riesgo evidente de seguridad.

A pesar de que se han desarrollado proyectos para dar respuesta a los atributos de calidad en la construcción de software (Hernández et al., 2015, 2019a), la dificultad principal que se encuentra para diseñar e implementar un producto software para e-voting se centra en el atributo de calidad de seguridad. El error más grave, por parte de algunos de los desarrolladores e investigadores, es considerar que el nivel secreto de una votación electrónica es similar al de una entidad financiera, cuando en ésta la operación puede ser conocida por terceros autorizados; en cambio en el voto electrónico el anonimato es parte esencial del mismo (Panizo, 2015).

Por todo lo anteriormente descrito, nace del interés por indagar sobre tácticas arquitectónicas de seguridad que se puedan implementar en el desarrollo de un producto software para voto electrónico (e-voting).

Metodología

Para lograr el objetivo de identificar las tácticas arquitectónicas de seguridad, que son utilizadas en la construcción de productos software, se utilizó el protocolo de revisión

sistemática de literatura descrito en Kitchenham et al. (2015). Además, se adoptó algunos elementos de la propuesta metodológica realizada por Hernández et al. (2019b), la cual consta de las siguientes etapas: elaborar las preguntas de investigación, realizar un proceso de búsqueda, llevar a cabo un proceso de selección y hacer un proceso de evaluación de la calidad.

Preguntas de investigación

Las preguntas de investigación formuladas se basan en la pregunta principal: ¿Cuáles son las tácticas arquitectónicas de seguridad que aportan a la construcción de un producto software para *e-voting*?

A partir de la pregunta principal se derivaron las preguntas secundarias, que permitieron analizar y categorizar los estudios primarios:

RQ1. ¿Qué tácticas arquitectónicas de seguridad son utilizadas en la construcción de software?

RQ2. ¿Qué tácticas arquitectónicas de seguridad son utilizadas en *e-voting*?

Proceso de búsqueda

Como parte del protocolo para buscar estudios primarios, se identificó las fuentes de información y se definió la cadena de búsqueda, según la pregunta de investigación principal. Las siguientes bases de datos se utilizaron como fuentes de información: ACM digital library, IEEE explorer y Springer.

La cadena general creada para establecer los criterios de búsqueda se configura a partir de los datos mostrados en la Tabla 1.

Tabla 1

Cadena de búsqueda tipo seguridad

Concepto	Palabras relacionadas
Tactic	Strategy OR Technique
Architectural	Model OR Architecture
Security	Protection OR Defense OR Prevention
Software development	(Software OR System) AND (Development OR Creation OR Construction)

La cadena de búsqueda fue elaborada construyendo expresiones que utilizan los operadores booleanos OR y AND. El operador OR se lo utilizó para incorporar sinónimos del concepto de búsqueda; mientras que el operador AND permite agregar las palabras relacionadas en la cadena de búsqueda.

Para la ejecución de las búsquedas se examinó: título, resumen y palabras clave, dentro de los resultados obtenidos a través del uso de los motores de búsqueda de cada fuente de datos seleccionada. La revisión de artículos se limitó a aquellos que se encuentren escritos en inglés y dentro de una ventana de observación entre 2014 y 2019.

Proceso de selección

Después de realizar el proceso de búsqueda, se procedió a seleccionar los estudios relevantes, es decir, aquellos artículos que permiten dar respuesta a las preguntas de investigación planteadas. Para determinar la relevancia de los estudios, se establecieron unos criterios de inclusión y exclusión, como se pueden observar en la Tabla 2.

Tabla 2

Criterios de selección de estudios

Criterios de inclusión	Criterios de exclusión
1. Artículos que se relacionan con tácticas arquitectónicas de seguridad	1. Artículos que no se relacionen con tácticas arquitectónicas de seguridad
2. Artículos publicados en una ventana de observación entre 2014 y 2019	2. Reportes técnicos
3. Artículos escritos en inglés	3. Estudios duplicados
4.Artículos resultados de estudios primarios	

Para la aplicación de los criterios de inclusión y exclusión, como parte de la selección, se definieron los filtros que se muestran en la Tabla 3.

Tabla 3

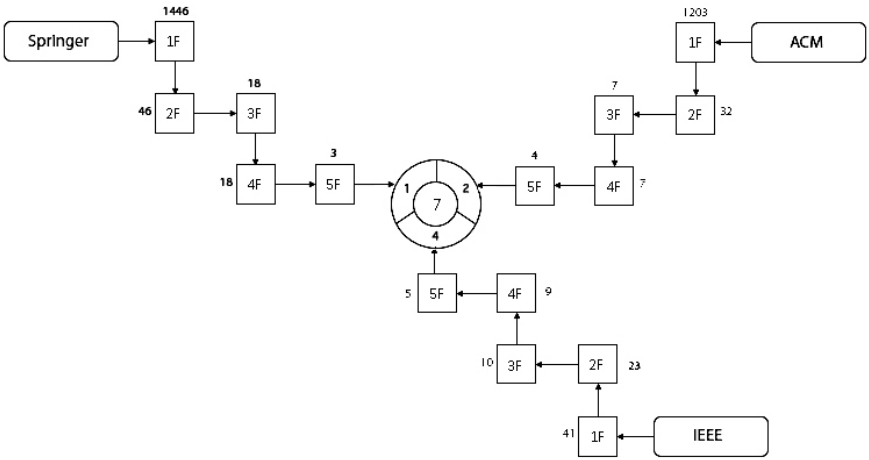
Estrategia de selección

Filtro	Descripción	Criterio aplicado	
		Inclusión	Exclusión
1F	Buscar los artículos mediante la cadena de búsqueda en los motores de las fuentes seleccionadas	3 y 4	
2F	Leer el título, palabras clave y resumen del artículo y aplicar los criterios de inclusión y exclusión	1, 2 y 5	1, 2 y 3
3F	Leer los resultados y conclusiones del artículo y aplicar los criterios de inclusión y exclusión	1, 2 y 5	1, 2 y 3
4F	Eliminar los estudios duplicados		4
5F	Leer el artículo completo y aplicar los criterios de inclusión y exclusión	1, 2 y 5	1, 2 y 3

En la Figura 1 se puede observar los resultados obtenidos de la aplicación de los filtros, definidos en la Tabla 3.

Figura 1

Resultado de la revisión sistemática



Proceso de evaluación de la calidad

Posterior al proceso de selección, se realizó una nueva tarea para asegurar la calidad de los artículos encontrados, que corresponde a inspeccionar un conjunto de criterios, que se muestran en la Tabla 4, en los artículos seleccionados.

Tabla 4

Criterios de evaluación de la calidad

Criterio	Descripción	Categoría
C1	Los objetivos y preguntas de investigación se describen de forma explícita, son claros y relevantes	Calidad del reporte
C2	La investigación presenta un diseño metodológico que le permite alcanzar los objetivos	Rigor
C3	El procedimiento de recopilación de datos es coherente con el diseño metodológico	Rigor

Criterio	Descripción	Categoría
C4	Los resultados presentados son claros y coherentes con el diseño metodológico propuesto	Credibilidad
C5	El estudio es valorado por otros investigadores	Relevancia

Para evaluar la calidad de los artículos se estableció una escala para inspeccionar el nivel de cumplimiento de los criterios de la siguiente manera: Alto (2 puntos), Medio (1 punto) y Bajo (0 puntos).

Tabla 5

Matriz de evaluación de la calidad de los artículos

Título del artículo	C1	C2	C3	C4	C5	$\sum_{i=1}^{5} Ci$	% Ci
Mitigating Security Threats Using Tactics and Patterns: A Controlled Experiment (Pedraza-García et al., 2016)	2	2	2	1	0	7	70 %
Security Tactics Selection Poker (TaSPeR) (Osses et al., 2018)	2	2	2	1	0	7	70 %
A Methodological Approach to Apply Security Tactics in Software Architecture Design (Pedraza-Garcia et al., 2014)	2	2	2	1	2	9	90 %
Building Sustainable Software by Preemptive Architectural Design Using Tactic-Equipped Patterns (Kim et al., 2014)	2	2	2	1	2	9	90 %
Detecting, Tracing, and Monitoring Architectural Tactics in Code (Mirakhorli y Cleland-Huang, 2015)	2	2	2	2	2	10	100 %
Understanding Software Vulnerabilities Related to Architectural Security Tactics (Santos et al., 2019)	2	2	2	2	2	10	100 %
Revisiting architectural tactics for security (Fernandez et al., 2015)	2	2	2	2	1	9	90 %

Los artículos que cumplieron con un 80 % o más del total de los puntos posibles son lo que finalmente se eligieron. Al finalizar el proceso de evaluación de calidad de los artículos solo se incluyeron 5.

Resultados

Los artículos que finalmente fueron seleccionados se los puede observar en la Tabla 6. El 80 % de los artículos corresponden a actas que se generan como memorias presentadas en eventos. Estos estudios están en un rango de años entre 2014 a 2017, además, IEEE es la principal fuente de información.

Tabla 6

Estudios seleccionados

ID	Estudio	Tipo de artículo	Año	Fuente
S1	A Methodological Approach to Apply Security Tactics in Software Architecture Design (Pedraza-Garcia et al., 2014)	Acta de congreso	2014	IEEE
S2	Building Sustainable Software by Preemptive Architectural Design Using Tactic-Equipped Patterns (Kim et al., 2014)	Acta de congreso	2014	IEEE
S3	Detecting, Tracing, and Monitoring Architectural Tactics in Code (Mirakhorli y Cleland-Huang, 2015)	Artículo publicado en revista	2015	IEEE
S4	Understanding Software Vulnerabilities Related to Architectural Security Tactics (Santos et al., 2019)	Acta de congreso	2017	IEEE
S5	Revisiting architectural tactics for security (Fernandez et al., 2015)	Acta de congreso	2015	Springer

Tácticas arquitectónicas de seguridad

Las tácticas arquitectónicas de seguridad identificadas al inspeccionar los artículos seleccionados se pueden observar en la Tabla 7. Los tipos de tácticas de seguridad más frecuentes en los estudios seleccionados son los siguientes: detección, resistencia y recuperación de ataques. Además, los campos de uso principales son industria y academia, a su vez, los recursos generalmente empleados para las tácticas son algoritmos de cifrados, copias redundantes de datos, sistemas biométricos y certificados digitales.

Tabla 7

Tácticas arquitectónicas de seguridad identificadas

Estudio	Tipo de táctica de seguridad	Campo de uso	Práctica en la táctica de seguridad
S1	Detectar ataques Resistir ataques Reaccionar a los ataques Recuperarse de los ataques	Gobierno	Evalúa casos de uso Especifica la vista funcional en capas Uso de modelo de datos Identifica recursos sensibles Define políticas de seguridad Uso de modelo de amenaza (árbol)
S2	Resistencia a ataques Detección de ataques Recuperación de ataques	Academia	Identificación de antipatrones Identificación patrones compatibles Visualización de la arquitectura Administración de requerimientos Control de la seguridad Reutilización
S3	Auditoría Autenticación HMAC Sesiones seguras Administración y RBAC	Industria	Verificación permanente de la calidad Visualización de la arquitectura Administración de requerimientos Pruebas unitarias Control de la seguridad
S4	Auditoría Verificar la integridad del mensaje Detectar intrusiones Detectar ataque de denegación de servicio Informar a los actores Cambiar la configuración predeterminada Separar entidades Cifrar los datos Limitación/Restricción de servicios Limitar acceso Autorización de usuarios Autenticación de usuarios Administrar sesiones de usuarios Validar entradas Identificar usuarios	Industria	Visualización de la arquitectura Administración de requerimientos Pruebas unitarias Control de la seguridad

Estudio	Tipo de táctica de seguridad	Campo de uso	Práctica en la táctica de seguridad
S5	Detectar ataques Resistir ataques Reaccionar a los ataques Recuperarse de los ataques	Academia	Uso de políticas de seguridad Jerarquía de políticas Seguridad aplicada a todo el sistema Detectar intruso Integridad de servicios Integridad de mensajes Identificar, autenticar, autorizar autores Limitar acceso Cifrar datos Informes del sistema Auditoría Mantener disponibilidad Complementar la seguridad del sistema con patrones de seguridad

Tácticas arquitectónicas de seguridad en *e-voting*

Las tácticas arquitectónicas de seguridad en *e-voting,* identificadas al inspeccionar los artículos seleccionados, se pueden observar en la Tabla 8.

Tabla 8

Tácticas arquitectónicas de seguridad para e-voting identificadas

Estudio	Tipo de táctica de seguridad en *e-voting*	Recurso para la táctica
S1	Resistiendo ataques Reaccionando a ataques Recuperarse de ataques	Experto seguridad Responsable de tomar decisiones Componentes de redes para interoperabilidad Bases de datos
S2	Resistencia a ataques Detección de ataques Recuperación de ataques	Firewalls y DMZ Sistemas biométricos Certificados o firmas digitales Sistemas de cifrado Parámetros de acceso o niveles tipo Grant Controles de cifrado (DES, 3DES, AES) Algoritmos de cifrado (simétricos, cuánticos, asimétricos, de curva elíptica o híbridos) Mecanismos de Checksum y códigos Hash Copias redundantes de los datos Uso de patrones Tunneling y configuración de VPN

Estudio	Tipo de táctica de seguridad en *e-voting*	Recurso para la táctica
S3	Auditoría Autenticación HMAC Sesiones seguras Administración y RBAC	Certificados o firmas digitales Parámetros de acceso o niveles tipo Grant Listas de usuarios Roles del sistema Dominios Controles de cifrado (DES, 3DES, AES) Copias redundantes de los datos Tunneling y configuración de VPN kernels y shells de seguridad
S4	Auditoría Verificar la integridad del mensaje Detectar intrusiones Detectar ataque de denegación de servicio Cifrar los datos Limitación/ Restricción de servicios Limitar acceso Autorización de usuarios Autenticación de usuarios Administrar sesiones de usuarios Validar entradas Identificar usuarios	Monitores de red y registro de eventos Sistemas biométricos Certificados o firmas digitales Sistemas de cifrado Controles de cifrado (DES, 3DES, AES) Mecanismos de Checksum y códigos Hash Copias redundantes de los datos
S5	Detectar ataques Resistir ataques Reaccionar a los ataques Recuperarse de los ataques	Sistemas de control y autenticación Sistemas de cifrado de datos Algoritmos de cifrado Red con canales seguros Patrones de seguridad de apoyo Hardware de nueva generación

De la Tabla 8 se puede deducir que el tipo de tácticas de seguridad en *e-voting*, que más se mencionan, se relacionan con detectar, resistir, reaccionar y recuperarse de ataques. Con respecto a los recursos para la táctica, entre los principales están: las copias redundantes de los datos, sistemas de cifrado y roles en el equipo de trabajo.

Existen métodos que permiten pensar en cómo lograr que un sistema cumpla con los atributos de seguridad; uno de ellos es pensar en la seguridad física. Las instalaciones seguras tienen acceso limitado (por ejemplo, mediante el uso de puntos de control de seguridad), medios para detectar intrusos (por ejemplo, al exigir a los visitantes legítimos que usen insignias), mecanismos de disuasión (como guardias armados), mecanismos de reacción (como el bloqueo automático de puertas) y mecanismos de recuperación (copia de seguridad fuera del sitio). Estos conducen a cuatro categorías de tácticas: detectar, resistir, reaccionar y recuperarse (Bass et al., 2013).

Para Bass et al. (2013), detectar ataques consiste en 3 tácticas: detectar intrusos, detectar ataques de denegación de servicio y verificar integridad de mensajes. Igualmente, Bass et al. (2013) plantean que en la categoría para resistir ataques existen varios medios bien conocidos como: identificar, autenticar y autorizar actores, limitar el acceso, también la exposición, encriptar datos y administrar sesiones a usuarios. En la categoría de reacción ante ataques existe una táctica denominada informar actores, destinada a responder a un posible ataque (Bass et al., 2013). Finalmente, los mismos autores plantean que para la recuperación de ataques, una vez un sistema ha detectado e intentado resistir un ataque, necesita recuperarse. Parte de la recuperación es la restauración de los servicios y datos, por ejemplo, servidores adicionales o conexiones de red pueden mantenerse en reserva para tal fin, dado que un ataque exitoso puede considerarse un tipo de falla.

En el presente estudio se excluyó un gran número de artículos, debido a que no cumplían con los requisitos necesarios que se evaluaban en cada filtro, pero, aun así, llama la atención el número de estudios sobre tácticas arquitectónicas de seguridad, sin embargo, estos estudios no están enfocados específicamente en *e-voting.* En ausencia de estudios orientados a estos sistemas, se debería tener en cuenta fuentes de información diferentes a las tres usadas en este estudio, por ejemplo, bases de datos genéricas que brinden información acerca de *e-voting*, su implementación y un seguimiento en diferentes instituciones, esto permitiría que se pueda evidenciar las tácticas mencionadas en estudios anteriores. Hay que mencionar, además, que en los estudios evaluados hay una gran similitud en las tácticas arquitectónicas de seguridad, que están de acuerdo con autores como Bass et al. (2013), donde se mencionan características del software teniendo en cuenta la seguridad de estos, lo cual concuerda, en gran parte, con otros estudios realizados. Dentro de algunos estudios realizados sobre las tácticas arquitectónicas varían atributos dentro de las tácticas, sin embargo, se centran en detectar, resistir, reaccionar y recuperarse de ataques.

Construcción del sistema Kybernan

Como parte de la arquitectura de software se implementaron algunas de las tácticas arquitectónicas de seguridad identificadas para *e-voting.*

En la gestión del desarrollo del producto software se utilizó algunos elementos del *framework Scrum* (Hernández et al., 2015). Los artefactos utilizados fueron el *product*

backlog y el *sprint backlog*. Los eventos correspondieron con el *sprint planning meeting*, *daily Scrum, review* y las *retrospective* (Hernández et al., 2019b). Para los lineamientos se utilizaron métricas con el fin de medir el desempeño y velocidad del equipo y para los roles se empleó el producto *owner, development team* y *Scrum master*.

En total se realizaron 6 sprints, cada uno tuvo una duración de 15 días. Por cada sprint se presenta información en relación con: necesidades priorizadas (*sprint backlog*), gráfico de objetivos suaves para la definición de las decisiones de arquitectura, resultados del *review* e información relacionada con la medición de velocidad y desempeño.

Para establecer las decisiones de diseño implementando tácticas arquitectónicas de seguridad, se utilizó como técnica el árbol de objetivos suaves (SIG, por sus siglas en inglés *Softgoal Interdependency Graph*). El SIG se elaboró informalmente, tiene objetivos y subobjetivos de calidad representados como objetivos flexibles y soluciones candidatas de diseño representadas como operacionalizaciones (Kobayashi et al., 2016). En la Figura 2 se observa las decisiones de diseño adoptadas para cada táctica arquitectónica de seguridad en el producto software para *e-voting*.

Figura 2

Árbol de objetivos suaves

Al finalizar los 6 *sprints, s*e logró construir un producto software para *e-voting* funcional llamado “Kybernan”, que implementa tácticas arquitectónicas de seguridad para la gestión de procesos democráticos mediante la realización de 27 historias de usuario, con un porcentaje de 74,07, según el nivel de dificultad Media.

De acuerdo con el atributo de calidad de seguridad, se implementaron 8 tácticas arquitectónicas. Dentro de las decisiones de diseño más importantes se encontró que la implementación de las tecnologías *JWT* y el uso de roles y permisos tipo *grant* fueron las que aportaron con mayor frecuencia al sistema.

De los 6 *sprints*, en un 83,3 % se logró cumplir con el objetivo planeado. Entre los principales impedimentos para no cumplir con los objetivos definidos para los *sprints* están los siguientes: dar respuesta al manejo de sesiones y autorización de actores, estas actividades se incorporaron en el *sprint backlog*, pero no se estimaron, por esta razón, se avanzaron dos de las cinco historias de usuario planeadas para el *sprin*t 1.

Adicionalmente, se extendió la implementación de la táctica de autenticar actores, debido a un inconveniente de compatibilidad entre las versiones de angular y *face API*.

En relación con la productividad en el desarrollo del proyecto, se puede afirmar que la etapa que mayor porcentaje de tiempo requirió fue la de diseño, porque el desarrollo del sistema estuvo centrado en implementar tácticas arquitectónicas. En total se trabajaron 337 horas y 34 minutos en el proyecto. De acuerdo con la planeación, se presentó un porcentaje de desperdicio de 6.

Conclusiones

La mayor fuente de información para identificar tácticas arquitectónicas de seguridad fue *IEEE*, en el rango de 2014 a 2017, brindó en sus artículos una parte importante para desarrollar este estudio, sin embargo, los artículos evaluados no se centran específicamente en *e-voting*.

Las tácticas arquitectónicas de seguridad identificadas se orientaron principalmente a detectar, resistir, reaccionar y recuperarse de ataques, junto con auditar, autenticar y establecer sesiones seguras. Los campos de mayor aplicación para estas tácticas fueron la industria, la academia y el gobierno. Además, se logró sistematizar las tácticas al identificar las principales prácticas, efectos y los recursos utilizados para implementarlas.

Las tácticas arquitectónicas de seguridad pertinentes para un producto software de *e-voting*, identificadas de manera sistemática, son aquellas que permitieron detectar, resistir, reaccionar y recuperarse de ataques.

Se construyó Kybernan, un producto software para *e-voting*, que implementa ocho tácticas arquitectónicas de seguridad para la gestión de procesos democráticos. Dentro de las decisiones de diseño más importantes incorporadas está la implementación de tecnologías *JWT* (por sus siglas en inglés *Json Web Token*) y el uso de roles y permisos tipo *grant*.

Kybernan, en un ambiente experimental, no permitió exponer en su totalidad las pruebas de seguridad que se pueden presentar en un ambiente de producción, esto se debe a que bajo este entorno de desarrollo se expuso al sistema en escenarios concretos y controlados.

Referencias

Alvarado, A. (2018, 28 de mayo). El 46,6 por ciento de los colombianos no votaron. *El Tiempo.* https://www.eltiempo.com/elecciones-colombia-2018/presidenciales/colombianos-que-no-votaron-a-la-presidencia-223064

Bass, L., Clements, P., & Kazman, R. (2013). *Software architecture in practice*. Addison-Wesley Professional.

BBC Mundo. (2016, 3 de octubre). Qué dice de Colombia que haya habido 62 % de abstención en el histórico plebiscito por el proceso de paz. *BBC News*. http://www.bbc.com/mundo/noticias-america-latina-37539590

Estudiantes de Comunicación Social y Periodismo - Universidad los Libertadores. (2018, 3 de abril). *¿Es viable y necesario el voto electrónico en Colombia? Las 2 Orillas*. https://www.las2orillas.co/es-viable-y-necesario-el-voto-electronico-en-colombia/

Fernandez, E. B., Astudillo, H., & Pedraza-García, G. (2015). Revisiting Architectural Tactics for Security. In: Weyns D., Mirandola R., Crnkovic I. (eds.), *Software Architecture. ECSA 2015. Lecture Notes in Computer Science* (Vol. 9278; pp. 55-69). Springer, Cham. https://doi.org/10.1007/978-3-319-23727-5_5

Hernández, G., Martínez, Á., Argote, I. y Coral, D. (2015). Metodología adaptativa basada en Scrum : Caso empresas de la Industria de Software en San Juan de Pasto - Colombia. *Revista Tecnológica ESPOL*, *28*(5), 211–223. http://www.rte.espol.edu.ec/index.php/tecnologica/article/view/435

Hernández, G., Martínez, Á., Jiménez, R., & Jiménez, F. (2019b). Métricas de productividad para equipo de trabajo de desarrollo ágil de software: una revisión sistemática. *TecnoLógicas*, *22*, 63–81. https://doi.org/10.22430/22565337.1510

Hernández, G., Martínez, Á., Jiménez, R. y Jiménez, F. (2019a). Scrum y Peopleware : elementos clave para la gestión en la construcción de software. *Revista Ibérica de Sistemas e Tecnologias de Informacao, E19*(4), 265–277.

Kim, D.-K., Ryoo, J., & Kim, S. (2014). Building sustainable software by preemptive architectural design using tactic-equipped patterns. En IEEE (Ed.), *2014 Ninth International Conference on Availability, Reliability and Security* (pp. 484–489).

Kobayashi, N., Morisaki, S., Atsumi, N., & Yamamoto, S. (2016). Quantitative Non Functional Requirements evaluation using softgoal weight. *Journal of Internet Services and Information Security (JISIS)*, *6*(1), 37–46.

Mirakhorli, M., & Cleland-Huang, J. (2015). Detecting, tracing, and monitoring architectural tactics in code. *IEEE Transactions on Software Engineering*, *42*(3), 205–220.

Osses, F., Márquez, G., Villegas, M. M., Orellana, C., Visconti, M., & Astudillo, H. (2018). Security tactics selection poker (TaSPeR) a card game to select security tactics to satisfy security requirements. *Proceedings of the 12th European Conference on Software Architecture: Companion Proceedings*.

Panizo, L. (2015). *Desarrollo de una metodología para el análisis y la clasificación de los sistemas de voto electrónico* (tesis de doctoral, Universidad de León). Buleria. https://buleria.unileon.es/handle/10612/4237

Pedraza-Garcia, G., Astudillo, H. & Correal, D. (2014). A methodological approach to apply security tactics in software architecture design. *2014 IEEE Colombian Conference on Communications and Computing (COLCOM)*. 10.1109/ColComCon.2014.6860432

Pedraza-García, G., Noël, R., Matalonga, S., Astudillo, H., & Fernandez, E. B. (2016). Mitigating security threats using tactics and patterns: a controlled experiment. *Proccedings of the 10th European Conference on Software Architecture Workshops*.

Quiñones, J. (2017). Sabe Colombia 2015: Estudio Nacional de Salud, Bienestar y Envejecimiento. *Carta Comunitaria*, *25*(144), 24–35.

Ramos, D. (2017, 29 de marzo). A fondo: ¿Es seguro el e-voting? (II). *Silicon.es*. https://www.silicon.es/a-fondo-seguro-e-voting-ii-2332352

Registraduría Nacional del Estado Civil. (2014). Así participan los colombianos en las elecciones presidenciales. https://wsr.registraduria.gov.co/Asi-participan-los-colombianos-en.html

Santos, J. C., Peruma, A., Mirakhorli, M., Galster, M., Vidal, J. V., & Sejfia, A. (2019). Understanding Software Vulnerabilities Related to Architectural Security Tactics. An Empirical Investigation of Chromium, PHP and Thunderbird. En *2017 IEEE International Conference on Software Architecture (ICSA)* (pp. 69-78). 10.1109/ICSA.2017.39