Indonesian Journal of Electrical Engineering and Computer Science
Vol. 28, No. 2, November 2022, pp. 1020~1027
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v28.i2.pp1020-1027

Threat modeling in application security planning citizen service complaints

Agus Tedyyana[1], Fajar Ratnawati[1], Elgamar Syam[2], Fajri Profesio Putra[1]
[1]Department of Informatics Engineering, Politeknik Negeri Bengkalis, Riau, Indonesia
[2]Department of Informatics, Universitas Islam Kuantan Singingi, Riau, Indonesia

Article Info

Article history:

Received Jan 16, 2022
Revised Jul 23, 2022
Accepted Aug 29, 2022

Keywords:

Abuse case
Application security
Citizen services
STRIDE
Threat modeling

ABSTRACT

The mobile-based service complaint application is one way to implement good governance today. Public facilitated to make complaints without going through a complicated process. Security aspects must be considered to protect user privacy. The security design must be considered so that no one is harmed by the application's users damaged in the application's use. This study used threat modeling during the planning stage of developing a citizen service complaint application to obtain information about vulnerabilities. The researcher uses the threat modeling process that the open web application security project (OWASP) organization has formulated as a framework. The researchers took steps to describe application information, determine and rank threats, countermeasures, and mitigation. In the final stage, the spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege (STRIDE) threat modeling methodology is used to analyze and assess mitigation actions against threats in the application. The researcher gets a defense strategy to reduce the danger based on the threat analysis results. Threat modeling in the early phase software development life cycle process is constructive in ensuring that software is developed with adequate security based on threat mitigation from the beginning.

Corresponding Author:

Agus Tedyyana
Department of Informatics Engineering, Politeknik Negeri Bengkalis
Bengkalis, Riau, Indonesia
Email: agustedyyana@polbeng.ac.id

1. INTRODUCTION

The use of mobile platforms to date is increasingly being used in application development. With so many applications that are customized in such a way, there is the possibility of a significant security risk to the application. Abundant features and rich functionalities have the opportunity to have sensitive data from application users stolen by attackers [1]. The development of smartphone technology and increasing internet activity have made digital data more diverse. Photos, videos, text, IP addresses, cookies in the browser, and global positioning system (GPS) coordinates are digital data types [2]. For example, attackers' data is taken if the application is not designed correctly from a security point of view name, places of birth, addresses, and telephone numbers. Examples of vulnerabilities that can occur, for example, applications that are designed not to pay attention to the encryption of user data, the possibility that can happen is that user data is stolen when users use connections on public Wi-Fi, attackers can easily snoop on the data that users send [3].

According to the announcement of the Cyber Operation Security Center for the Indonesian National Cyber and Crypto Agency, during 2019, the point-of-view monitoring system detected around 290.3 million cyberattacks (intrusions) into the Indonesian internet network. The largest was a data leak test attack, followed by malware attacks. Compared to many cyber-attacks, the number of public complaints regarding incidents that occur is relatively minimal; cyber-attacks spiked sharply in September, October and decreased sharply in November. In November and December, this figure is still much higher than the first six months of 2019 [4].

The fact that is happening today, fraud is a type of digital crime that often occurs compared to physical fraud. Cyber-attacks are a threat to large and small companies. According to a report from the federal bureau of investigation (FBI) internet crime report, the estimated loss from digital crime in 2020 alone is nearly 40 billion [5].

Cyber-attacks or digital attacks occur on small companies because usually, small companies do not have a high-security infrastructure like large companies. However, it is possible that large companies can also become targets of cyberattacks. The more data large companies have the more profits cybercriminals will get. That puts the company at legal risk if customer information is leaked due to the company's negligence. In addition, the leakage of personal data will reduce customer trust or disappear altogether [6].

Security in the use of application data has become a major concern of the user it self [7]. Users feel worried about the vulnerabilities in the application that can cause losses to users. For example, the existence of malicious code that allows user activity tracked and sensitive data theft. At least about 52% of users remove applications and 40% of these qualities stop using it because of the user data privacy security problem [8]. This is a challenge for application developers to have to pay attention to user data security in maintaining user loyalty in using the application [9]. With the advancement of information technology development, the government is required to make innovations that realize the participation of its citizens in supporting the open government. In addition, many reasons need the government to transform information technology (IT) services, such as cutting the budget, increasing productivity without increasing the budget, and increasing the quality of products and services [10].

The citizen service complaint application is an application developed to make it easier for citizens to make complaints about any service discrepancies they receive from the government [11]. The data consumed in this application is of course very sensitive considering that to guarantee a valid pair of data required such as ID number, name, location, and phone number of the user. Based on these researchers then perform security design on the application by applying threat modeling in the application development process.

As an important part of application security, threat modeling has become widespread in application development and system evaluation [12]. This study presents a review of threat modeling in a security case study for a public service complaint application that the researcher will build. The purpose of this work is to make it easier for researchers and other practitioners to get an idea of how threat modeling is applied to application development with security in mind, as well as find possible directions for further research.

According to research [13], the solution to the problem to assist citizens in submitting a report to the government is to build an effective and efficient application to improve government services and performance (good governance). However, in developing applications that are related to government and require sensitive citizen data. User data security must be the main focus that urgently needs to be considered in the heterogeneous and interrelated application architecture between services [14].

Threat modeling is now a growing trend and much discussed in the cybersecurity domain because it can help in several aspects to make the system more secure from a cyber threat [15]. From research [12] which describes and discusses various theories about threat modeling, interpreting threat modeling as a first step can be useful for exploring potential attacks from hackers. Similarly, threat modeling provides a structured way to secure a software design, which involves understanding the objectives of the adversary in attacking the system based on the assets present in the system.

Threat modeling is best used early in the development cycle. This means potential problems can be caught early and fixed and can prevent much more expensive app fixes from happening. Thinking about security requirements with threat modeling can lead to proactive architectural decisions that enable threat reduction from the outset [16].

Several methodologies can be used in threat modeling, one of which is the spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege (STRIDE) [17]. STRIDE is a threat modeling classification developed by Microsoft. STRIDE is an acronym that contains the following concepts, namely spoofing, tampering, repudiation, information disclosure, denial of service, and elevation [18]. Our main goal in this study was to identify and categorize threats to a public service complaint application from an attacker's point of view, and the STRIDE model fits this goal.

Threat modeling methodologies, such as STRIDE, start with data flow diagrams (DFD) [15], which are system-level abstractions representing external entities that interact with systems, processes, data flows, and data stores. Based on the DFD, systematic iteration of all the model elements will produce potential security threats that need to be assessed. In the next step, these threats are documented and prioritized and further guide the process of determining appropriate security solutions to mitigate them [19].

2. RESEARCH METHOD

In this study, researchers used a threat modeling process formulated by the open web application security project (OWASP) organization [20]. The steps taken are to describe application information, determine and rank threats, countermeasures, and mitigation. From the final results of this research, it will be used as a security guide in developing applications.

a) Describing application information

The first step in the threat modeling process is concerned with gaining an understanding of the application and how it interacts with external entities. The information from the application is written as the result of the document and is also used to describe the data flow diagram (DFD) of the application being analyzed. The DFD shows the different steps traversed through the system and describes the boundaries of access rights.

b) Threats ranking

It is essential to identify threats by applying a threat modeling methodology. Threat classification like STRIDE may be used to define threat categories such as exception management, configuration management, auditing and logging, data validation, authentication, authorization also data protection in storage and transit. The purpose of threat classification is to help identify threats using (STRIDE). The DFD generated in the previous step helps identify potential threat targets from the hacker's side of view, such as data flows, interactions with users, data sources, and processes. Use and abuse cases can illustrate how Threats can easily bypass existing protective measures or where such protection is not in place [21].

c) Countermeasures and mitigation

Vulnerabilities can be reduced by implementing countermeasures [22]. These countermeasures can be recognized using a list of threat countermeasures mapping. Once a risk rating has been set to threats in the previous step, it is still possible to prioritize mitigation efforts and rank threats known from highest to lowest risk.

A risk mitigation strategy may cover evaluating the threat from the resulting business model impact [23]. When the possible impact is discovered, options for addressing risks are identified and include:

- Accept: determine whether the impact on the business is acceptable or not.
- Eliminate: remove components that might cause security vulnerabilities.
- Mitigation: adding checks and controls that reduce the impact of risks or possible causes.

3. RESULTS AND DISCUSSION

In this section, the results of the methodological steps that have been carried out in the previous chapter are presented. The results section begins with the threat model information which contains basic application information. Then it ends with the STRIDE Threat and mitigation stage which is the final result of this research.

a) Threat model information

The first step is to develop a threat model information. This information is in the form of basic information about the application to be tested. The following is the threat model information obtained in the first step shown in Table 1.

Table 1. Threat model information

App Name	PDAM Service
App Version	1.00.a
Description	This application is a means intended for customers and non-customers to report disturbances or complaints related to services through the Android application. Users are required to have an account and then log in first to report and change their profiles.
	Agus Teddyana
	Asep Subandri
Document Owner	Fajar Ratnawati
Participant	
Reviewer	

The Table 1 shows several indicators in the table, namely application name, which contains the name of the application to be checked; application version, which is an application; a description which includes an application’s high-level description, then owner of the document, participants, and reviewer respectively. Each contains the name of the owner of the threat model document. The participants who took part in the threat model process on the application, and the name of the reviewer of the threat model.

b) External dependencies

External dependencies are components outside the application code that can make a threat happen to the application [24]. These components may be not in the control of the development team but usually still within the control of the organization. The following are external dependencies that exist in the application shown in Table 2.

Table 2. External dependencies

ID	Description
1	The application is mobile based, but data exchange requires a REST API and is run in a shared hosting environment with the Apache webserver. Hosting management using the latest Cpanel system with support for security patches.
2	The database server uses PostgreSQL running in a shared hosting environment. Database management using the latest Cpanel system with support for the latest security patches.
3	The connection between the database server and the webserver is on the same hosting, in this case, on the private network hosting itself.

c) Entry points

The entry point defines the interface through which a potential attacker can exploit the application or supply it with data. The entry point in the application can consist of several layers. As an example, each state in a mobile application may consist of many entry points. The following entry points are shown in Table 3.

Table 3. Entry points

ID	Name	Description	Trust Level
1	HTTPS Port	Data exchange using REST API via Transport Layer Security (TLS). All pages in the application are layered at these entry points.	Login User Unlogin User
2	Login Page	The login form appears when the user has not logged in for the first time running the application.	Unlogin User
3	Login Functional	The connection between the login functional receives the login data from the user and compares it with the credential saved in the database server.	Unlogin User
4	Homepage	The main page appears only for logged-in users.	Login User

d) Trust levels

The trust level represents an application's permissions to an external entity. This makes it possible to define the required privileges or permissions at each entry point on the application. The following trust levels are shown in Table 4.

Table 4. Trust levels

ID	Name	Description
1	Login User	Users who are already connected to the application and have logged in with valid credential information.
2	Unlogin User	Users who are already connected to the application but have not logged in.

e) Data flow diagram

All the information that has been collected in the previous stages is then modeled accurately using data flow diagrams (DFD). DFD provides benefits for researchers to gain an understanding of the application by providing data visually about how the data is processed according to which flow from the user to the application. The following DFD is shown in Figure 1.

The focus of DFD is on how data moves through the application and what happens to the data as it progresses. DFD has a hierarchical structure, so DFD can use it to decompose applications. The high level of DFD helps researchers to better explain the scope of the application model. Low-level iteration will help in focusing on the explicit processes get involved when doing works on certain data.

From Figure 1, the DFD is described in the complaint service application. The data flow starts from the user requesting data from the application until the mobile app gives the response it gets from the database file. The data that the first-time user wants to obtain must be requested first using a mobile application. There are boundaries between users and mobile applications, separating their data flows. The mobile application will first display local assets to the user as an interface.

To call the data that the user wants, the application then requests the database server through the REST API. Between the application and the database server, there is also a limit. The database server then asks the intended database table to be returned to the user through the application first.

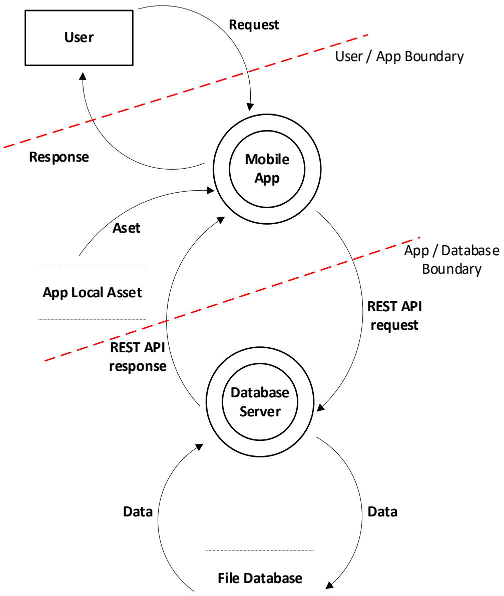

Figure 1. DFD of application

f) STRIDE

Threat methodology such as STRIDE in this study is useful in identifying threats by grouping the target [25]. Threat lists obtained based on the STRIDE model are useful in identifying threats based on the hacker's goals for example, the danger is using someone's identification number during registration. Will the attacker do social engineering on certain people to get that identification number? Another example, if the threat scenario is to make continuous requests to the REST API endpoint, will the attacker use bots and specific methods to carry out the attack? A list of threats list that known using STRIDE is provided in Table 5.

Table 5. STRIDE threat list

Type	Description	Data Flow
Spoofing	Can register using someone else's ID Number.	User data request
Tampering	User data can be stolen by attackers if using a public network.	User data request
Repudiation	Users use fake addresses to report complaints.	User data request
Information	Users snooping due to man-in-the-middle attacks.	User data request
Denial of service	Users make requests continuously on the provided form.	User data request
Elevation of privilege	The attacker steals using the user's internal data on the application and uses the token on that data to take over the user's account.	User data request
Spoofing	The attacker uses another user's API key to access that user's data.	REST API data Request
Denial of service	Attackers know REST API endpoints and make requests continuously.	REST API data Request

From the threat in Table 5, it can be seen that the types of attacks that appear more often than other types of attacks are spoofing and denial of service. With data flow through user data requests and REST API data requests. From this, researchers can gain new insights regarding which vulnerabilities should be secured.

g) Use and abuse case

Once common vulnerabilities, attacks, and threats have been assessed, have been evaluated, threat analysis that becomes more focused should consider abuse cases and use cases. By further learning of usage scenarios, vulnerabilities can be recognized and become aware of other threats. Cases of abuse should also be recognized. This abuse case can describe how existing security and prevention measures can be bypassed by attackers, whether there is security in the system. A use and misuse case of the application is shown in Figure 2.

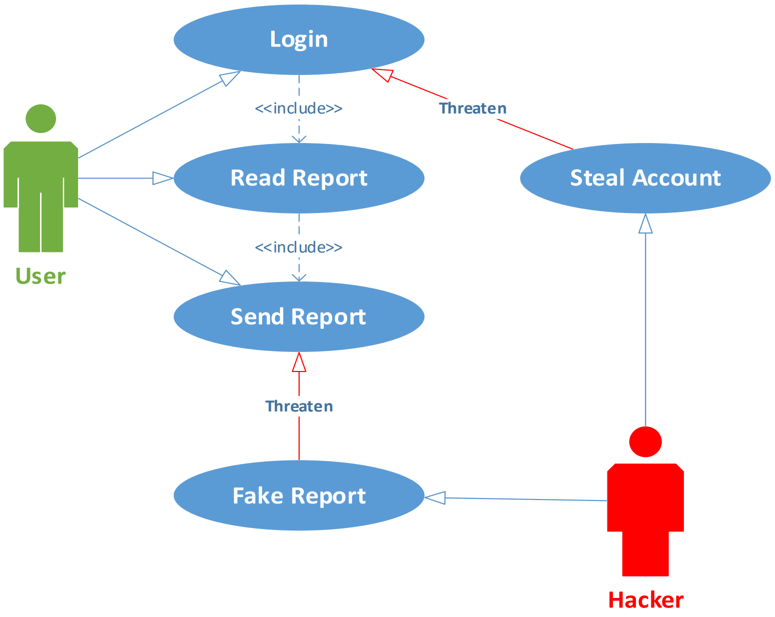

Figure 2. Use and misuse case

h) STRIDE threat and mitigation

The goal of identifying countermeasures is to decide whether there are some types of protective measures such as policies or security controls that might prevent. Vulnerabilities are threats that do not have protective measures. Once hazards have been appropriately classified using STRIDE, it is undoubtedly possible to hunt countermeasures that fit a given type. Table 6 is used to determine the method that researchers can use to reduce threats when utilizing the STRIDE methodology.

Table 6. Threat and mitigation technique

Threat Type	Mitigation Technique
Spoofing	1. Doing double validation when registering an account. 2. Enforce expiration time on REST API tokens.
Tampering	1. Using TLS in the form of HTTPS when transferring data. A system with support for security patches.
Repudiation	2. Shows a warning when the application is opened and when running is running Fake GPS or the like.
Denial of service	1. Provide a captcha if the user is indicated to have made a spam request. 2. Limiting the number of API requests that a user can make in a specific period.
Elevation of privilege	1. Limits the number of devices a user can use in each session.

Once threats and appropriate countermeasures are identified, a threat profile can be created according to the following standard:

1. Unmitigated threats: Threats that do not have countermeasures and show a vulnerability that allows to be fully exploited and have a severe effect on the application.
2. Partially mitigated threats: Threats that can be handled by one or more than one countermeasure also causes a limited impact and can only be partially exploited.
3. Fully mitigated threats: Threats that have an initial preventive measure and do not indicate a vulnerability.

4. CONCLUSION

Here are three reasons why data security/data security is essential. The first is to prevent potential material loss. The second is to reduce the risk of data/information. The last is to minimize the chances of criminal activity, The technology used in digital implementation must comply with world-recognized security standards. Don't just for the sake of saving costs; you must sacrifice company assets that are not worth the price.

Threat modeling creates a cloud of code in the process of review. Using threat modeling in the early phase of the software development life cycle process is very helpful in ensuring that software is developed with adequate security based on threat mitigation from the beginning. This study analyzes potential security threats for service complaint applications. It suggests several mitigation strategies to reduce the risk of identified threats through threat modeling analysis that has been widely used in information security. We propose several practical defense strategies to mitigate the identified threats based on the threat analysis results. In the following research, we will simulate the attacks that have been analyzed against the application system to show whether the threat modeling implementation has been successfully implemented.

ACKNOWLEDGEMENTS

We want to express our deepest gratitude to the editors, reviewers, and managers of the Indonesian Journal of Electrical Engineering and Computer Science. They have been willing to review and provide input and suggestions for developing and improving the quality of this research publication.

REFERENCES

[1] D. Wu, G. D. Moody, J. Zhang, and P. B. Lowry, "Effects of the design of mobile security notifications and mobile app usability on users' security perceptions and continued use intention," *Inf. Manag.*, vol. 57, no. 5, p. 103235, 2020, doi: 10.1016/j.im.2019.103235.

[2] V. Mishra, P. Laperdrix, A. Vastel, W. Rudametkin, R. Rouvoy, and M. Lopatka, "Don't count me out: on the relevance of IP address in the tracking ecosystem," *Web Conf. 2020 - Proc. World Wide Web Conf. WWW 2020*, pp. 808–815, 2020, doi: 10.1145/3366423.3380161.

[3] M. Bosamia and D. Patel, "Wallet payments recent potential threats and vulnerabilities with its possible security measures," *Int. J. Comput. Sci. Eng.*, vol. 7, no. 1, pp. 810–817, 2019, doi: 10.26438/ijcse/v7i1.810817.

[4] A. Tedyyana and O. Ghazali, "Teler real-time http intrusion detection at website with nginx web server," *Int. J. Informatics Vis.*, vol. 5, no. 3, pp. 327–332, 2021, doi: 10.30630/joiv.5.3.510.

[5] J. Hawdon, K. Parti, and T. E. Dearden, "Cybercrime in America amid COVID-19: the initial results from a natural experiment," *Am. J. Crim. Justice*, vol. 45, no. 4, pp. 546–562, 2020, doi: 10.1007/s12103-020-09534-4.

[6] S. K. Prabha, "Security threats in computer systems and internet web services," *Shanghai Ligong Daxue Xuebao/Journal Univ. Shanghai Sci. Technol.*, vol. 22, no. 10, pp. 2127–2147, 2020.

[7] S. Sharma and B. Kaushik, "A survey on internet of vehicles: applications, security issues & solutions," *Veh. Commun.*, vol. 20, p. 100182, 2019, doi: 10.1016/j.vehcom.2019.100182.

[8] A. Balapour, H. R. Nikkhah, and R. Sabherwal, "Mobile application security: Role of perceived privacy as the predictor of security perceptions," *Int. J. Inf. Manage.*, vol. 52, no. November 2019, p. 102063, 2020, doi: 10.1016/j.ijinfomgt.2019.102063.

[9] W. P. Wong, K. Tan, I. Inkgo, and B. Chiu-yiong, "The effect of technology trust on customer e-loyalty in online shopping and the mediating effect of trustworthiness," *J. Mark. Adv. Pract.*, vol. 1, no. 2, pp. 38–51, 2019.

[10] E. D. Madyatmadja, H. Nindito, and D. Pristinella, "Citizen behavior: The evaluation of complaint application that connected to smart city," *Adv. Sci. Technol. Eng. Syst.*, vol. 5, no. 4, pp. 24–29, 2020, doi: 10.25046/aj050403.

[11] A. M. Sari, A. N. Hidayanto, B. Purwandari, N. F. A. Budi, and M. Kosandi, "Challenges and issues of E-participation implementation: A case study of e-complaint Indonesia," *Proc. 3rd Int. Conf. Informatics Comput. ICIC 2018*, pp. 1–6, 2018, doi: 10.1109/IAC.2018.8780467.

[12] W. Xiong and R. Lagerström, "Threat modeling – A systematic literature review," *Comput. Secur.*, vol. 84, pp. 53–69, 2019, doi: 10.1016/j.cose.2019.03.010.

[13] A. Alkodri, B. Isnanto, and S. Sujono, "Public complaint application for reporting events and disasters at the national basarnas Bangka Belitung," *CSRID (Computer Sci. Res. Its Dev. Journal)*, vol. 11, no. 2, p. 96, 2021, doi: 10.22303/csrid.11.2.2019.96-104.

[14] S. Abidi, M. Essafi, C. G. Guegan, M. Fakhri, H. Witti, and H. H. B. Ghezala, "A web service security governance approach based on dedicated micro-services," *Procedia Comput. Sci.*, vol. 159, no. 2018, pp. 372–386, 2019, doi: 10.1016/j.procs.2019.09.192.

[15] M. Välja, F. Heiding, U. Franke, and R. Lagerström, "Automating threat modeling using an ontology framework: Validated with data from critical infrastructures," *Cybersecurity*, vol. 3, no. 1, 2020, doi: 10.1186/s42400-020-00060-8.

[16] N. Shevchenko, T. A. Chick, P. O. Riordan, T. P. Scanlon, and C. Woody, "Threat modeling: a summary of available methods," *Res. Rep.*, no. July, p. 26, 2018, [Online]. Available: https://apps.dtic.mil/sti/citations/AD1084024%0Ahttps://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf.

[17] N. R. Mead, F. Shull, K. Vemuru, and O. Villadsen, "A hybrid threat modeling method," *Carnegie Mellon Univ. Softw. Eng. Inst.*, no. March, p. 41, 2018, [Online]. Available: http://www.sei.cmu.edu.

[18] G. Cho, J. Choi, H. Kim, S. Hyun, and J. Ryoo, "Threat modeling and analysis of voice assistant applications," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11402 LNCS, pp. 197–209, 2019, doi: 10.1007/978-3-030-17982-3_16.

[19] L. Sion, K. Yskout, D. Van Landuyt, and W. Joosen, "Solution-aware data flow diagrams for security threat modeling," *Proc. ACM Symp. Appl. Comput.*, pp. 1425–1432, 2018, doi: 10.1145/3167132.3167285.

[20] L. Conklin, "Threat modeling process," 2020. https://owasp.org/www-community/Threat_Modeling_Process (accessed Jan. 04, 2022).

[21] L. P. Gonçalves and A. R. da Silva, "Towards a catalogue of reusable security requirements, risks and vulnerabilities," *Proc. 27th Int. Conf. Inf. Syst. Dev. Des. Digit. ISD 2018*, 2018.

[22] P. Nespoli, F. Gómez Mármol, and J. Maestre Vidal, "Battling against cyberattacks: towards pre-standardization of countermeasures," *Cluster Comput.*, vol. 24, no. 1, pp. 57–81, 2021, doi: 10.1007/s10586-020-03198-9.

[23] A. Frik, L. Nurgalieva, J. Bernd, J. Lee, F. Schaub, and S. Egelman, "Privacy and security threat models and mitigation strategies of older adults," *In Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, pp. 21-40, 2019.

[24] T. F. Düllmann, C. Paule, and A. Van Hoorn, “Exploiting devops practices for dependable and secure continuous delivery pipelines,” *Proc. - Int. Conf. Softw. Eng.*, pp. 27–30, 2018, doi: 10.1145/3194760.3194763.

[25] J. Straub, “Modeling attack, defense and threat trees and the cyber kill chain, ATTCK and STRIDE frameworks as blackboard architecture networks,” *Proc. - 2020 IEEE Int. Conf. Smart Cloud, SmartCloud 2020*, 2020, pp. 148–153, doi: 10.1109/SmartCloud49737.2020.00035.

BIOGRAPHIES OF AUTHORS

Agus Tedyyana is a senior lecturer at the Politeknik Negeri Bengkalis, Bengkalis, Riau, Indonesia. He has an educational background in computer science. He has worked in education as a lecturer since 2014. He is currently continuing his Doctoral (PhD) studies at the Universiti Utara Malaysia (UUM) Campus in Kedah Darul Aman, Malaysia, since early 2020. His research interests In computer security. He can be contacted at email: agustedyyana@polbeng.ac.id.

Fajar Ratnawati, worked at the Politeknik Negeri Bengkalis as a lecturer with a home base of D4-Software Engineering, Department of Informatics. Undergraduate education at STMIK AKAKOM is now known as the University of Digital Technology Indonesia Yogyakarta, majoring in Informatics Engineering. Master's education at Gadjah Mada University Faculty of Mathematics and Natural Sciences Computer Science Study Program interested in Software Engineering. The field of software research and data mining. She can be contacted at email that can be contacted is fajar@polbeng.ac.id.

Elgamar Syam is a senior lecturer at the Universitas Islam Kuantan Singingi, Telukkuantan, Riau, Indonesia. He has an educational background in computer science. He has worked in education as a lecturer since 2012 and has taught at various campuses in Riau. He is currently continuing his Doctoral (Ph.D.) studies at the Universiti Utara Malaysia (UUM) Campus in Kedah Darul Aman, Malaysia since early 2020. His research interests include the fields of information systems and government. He can be contacted at email:elgamar@uniks.ac.id.

Fajri Profesio Putra is a senior lecturer at the Politeknik Negeri Bengkalis, Bengkalis, Riau, Indonesia. He has an educational background in computer science. He has worked in education as a lecturer since 2015. Undergraduate education at Universitas Pendidikan Indonesia Bandung, majoring in Computer Science. Master's education at Gadjah Mada University Faculty of Mathematics and Natural Sciences Computer Science Study Program interested in Software Engineering. His research interests in software engineering, database, and web programming. He can be contacted at email: fajri@polbeng.ac.id.