INTERNATIONAL JOURNAL
ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Secure Agent-Oriented Modelling with Web-based Security Application Development

Macklin Ak Limpan [a,*] , Cheah Wai Shiang [a] , Eaqerzilla Phang [a], Muhammad Asyraf bin Khairuddin [a], Nurfauza bt Jali [a]

[a] *Faculty of Computer Science and Information Technology, University of Malaysia Sarawak, Kota Samarahan, 94300, Sarawak, Malaysia*
*Corresponding author: *macklinlimpan@gmail.com*

***Abstract*— Nowadays, privacy and security have become challenges in developing web-based applications. For example, e-commerce applications are threatened with security issues like scammers, SQL injection attacks, bots, DDOs, Server Security, and Phishing. Although various security requirement methodologies are introduced, it has been reported that security consideration is consistently ignored or treated as the lowest priority during the application development process. Hence, the application is being violated by various security attacks. This paper introduces an alternative methodology to secure a web-based application through an Agent-Oriented Modelling extension. The secure AOM starts with Context and Asset Identification. The models involved in this phase are the Goal Model and Secure Tropos model. The second phase is the Determination of Security Objective. The model that will be used is Secure Tropos. The third phase is Risk Analysis and Assessment. The model that will be used is Secure Tropos. The fourth phase is Risk Treatment. In this phase, there is no model, but we use the suggestion from Secure Tropos: to eliminate risk, transfer risk, retain risk, and reduce risk. The fifth phase is Security Requirements Definition. The models that will be used are the scenario model, interaction model, and knowledge model. The last phase is Control Selection and Implementation. The model that will be used is the Behavior Model. We conducted a reliability analysis to analyze the participants' understanding of Secure AOM. From the reliability test, we can conclude that Secure AOM can become the alternative methodology, as the percentage that agrees that Secure AOM can protect users against making errors and mistakes is 80.9%, and 71.9% agree that SAOM can help to prevent users from specifying incorrect model elements and the relation between the model. This result means that over 50% of the participants agree that Secure AOM can be an alternative methodology that supports security risk management.**

***Keywords*— AOM; secure modelling; secure methodology; Secure Tropos.**

Manuscript received 15 Sep. 2023; revised 19 Dec. 2023; accepted 23 Feb. 2024. Date of publication 31 Mar. 2024.

I. INTRODUCTION

Priority of security in the e-commerce system is a must because, with security, the number of users that trust the platform can be increased [2] and lead to fewer users using the platform. Based on [1], there are two disadvantages to the security of e-commerce if security is not their priority. There is a risk of losing financial information and a wrong perception from users who want to use the system. It will affect the users ' trust without privacy [2] and security. This is supported by [3] study, where they study how the security and privacy of the information can affect the user's level of trust in the e-commerce platform.

Several methods have been introduced for Security Requirement Engineering (SRE). There are KAOS [4] [5], STS [6], SEPP [7], and Secure Tropos [8]. SRE elicits, analyzes, and specifies the system's security requirements [9]. Security requirements can be classified as Non-functional requirements because they do not have clear criteria for specification and satisfiability [9], [10]. Although lots of SRE have been introduced, more methodologies are still needed to model a secure application. A study by [11] found that no secure methodologies have been proposed. The findings have validated the usefulness of Secure AOM, which methodology supports security risk management. Meanwhile, [12] found that few methodologies include security analysis in requirement engineering, although software development depends on the requirements engineering. P. Yeng et al. [10] also mention that none of the methodologies that have been entirely suitable outline all the security requirements activities.

For this study, we focused on Secure Tropos. As mentioned before, Secure Tropos is one of the SREs. Based on [13], the author said Secure Tropos uses graphical language to analyze the system's environment using the graphical language. Secure Tropos is used to help developers elicit the security requirements [14]. [12], [13].

This paper introduces an alternative methodology to secure a web-based application through an Agent-Oriented Modelling extension. AOM methodology uses an agent notion in each phase [15][16]. Based on [16], AOM adopts the agent notion in analyzing and gathering the requirements, designing, and implementing the complex system. Agents in this context can be represented as individuals, organizations, or software components[17]. Based on [18], AOM helps model a socio-technical system or application. However, the AOM methodology does not support risk management like SRE, where there are processes to elicit and analyze security requirements in SRE. With this gap, we developed the proposed methodology called Secure Agent-Oriented Modelling.

Secure AOM can be an alternative methodology to engineer a secure web application. The secure AOM highlights the importance of security and treats security as the first central entity in web application development. It is based on the Information System Security Risk Management (ISSRM) standard, which involves six phases. Based on [9], ISSRM integrates several security standards. One of the security standards that has been incorporated into ISSRM is ISO/IEC Guide 73. The definition of risk in this security standard is the combination of the probability of the risk and the consequences of the risk [19], where this security standard helps the developer to find the probability and the consequences of the risk. This security standard supports the developers in establishing, implementing, operating, monitoring, maintaining, and improving the Information Security Management System[19]. In [20], the author found three principles to ensure cyber security based on ISO 27001: the *principle of confidentiality of the information, the principle of information integrity,* and *the principle of availability of information.* Only authorized individuals can access their information for the principle of confidentiality of information. The principle of information integrity is to process the data to determine its accuracy. The last principle is the availability of information, which allows authorized individuals to access their data when requested. The following security standard is AS/NZS 4360. Based on [21] [22], the function of this standard is a guideline within the Management System framework. It will guide the company in proceeding with the risk management process. Security standards are necessary when gathering security requirements because we must first have a security standard to establish information security measures that can satisfy an organization's needs [23]. The information security standard must also be defined because it ensures that Information Technology follows local and international rules[24].

The structure of this paper is as follows: In Section 2, we explain more about what ISSRM is, what a risk management process is, what SRE is, and our proposed methodology. Section 3 shows the results of our experiment based on the questionnaire. The result demonstrates the participants' reliability and understanding of the modeling languages like UML, AOM, and AOM+Secure Tropos.

II. Material and Method

Several criteria are introduced when proposing an SRE [9]. According to [9], a good SRE can model the threat and make the risk analysis. This is because threat modeling is an important activity in the security requirement domain, while risk analysis provides the details of the threat. Meanwhile, a good SRE must also integrate security standards to support security and risk analysis, which several security standards have explained in Section I. The usability and performance in requirement phases are essential to choose the best fit SRE approach.

Based on the elaborated criteria, a secure AOM is designed. The Secure AOM is the Requirement Engineering method created with threat modeling and risk analysis based on ISSRM. SAOM can help the developer support requirement elicitation, resolve conflicts in security requirements, produce complete security requirements, and support requirement validation.

The introductory section elaborates that Secure AOM is designed based on the ISSRM reference model [25]. The ISSRM reference model helps people or organizations manage security risks. Hence, it is sensible to transform the Secure AOM steps based on the ISSRM security requirements lifecycle. Because the step is based on ISSRM, Secure AOM integrates the security standard discussed in Section I. This will make Secure AOM support risk management like SRE.

ISSRM has some core definitions: Asset-related *concepts, Risk-related concepts, risk treatment-related concepts, and the Security risk management process. The purpose of the core definition is to ensure that* the developer understands the idea that has been used in ISSRM.

1) Asset-related concept: In the *Asset-related concept,* the developer/modeler needs to identify the critical *asset* that needs to be protected since an *asset* has value and is necessary to achieve an objective. The developer/modeler must also determine the criteria to protect the asset.

2) Risk-related concepts: This concept explains security threats, attacks, and consequences.

3) Risk treatment-*related concepts*: This concept consists of the terminology for security solutions. It covers the security requirements and implementation.

4) Security Risk Management Process: There are 6 phases in managing security risk as follows:

- Contexts and Assets Identification of the organization.
- The determination of the security objective of the organization, which needs to be related to the assets that need to be protected.
- Risk analysis and assessment: The developer/modeler needs to identify possible security risks that can harm the organization and estimate how the attacker performs the security attack.
- Risk treatment will be performed after the risk assessment. In this phase, the developer/modeler must devise a possible solution that includes avoiding, reducing, transferring, and retaining.
- Security Requirements Definitions can be identified and classified as security solutions that can solve the security attack problem.

- Control selection and implementation, in which the countermeasure is implemented within the organization.

Secure AOM is a hybrid methodology that combines Secure Tropos and AOM. The reason is that Secure Tropos is beneficial for security analysis, while AOM will go with designing the system collected from Secure Tropos. As Secure Tropos [26] focuses on analysis, transforming the analysis model into the design is essential and has been overlooked in all SRE. The Software Development Life Cycle (SDLC) cannot be completed if there is no phase to design the system. This is because, during the design phase, we need to create the fundamental structure of the entire system, which we discuss in the requirement analysis phase[27]. The activities that will be involved in this phase include the choosing of programming language, verifying, specifying, and documenting the design activity.

The AOM consists of the goal model, role model, organization model, domain model, agent and acquaintance models, knowledge model, scenario model, interaction model, and behavior model[17][28] said that AOM is a methodology for complex socio-technical system development, which can support the modeling of the complex system. The AOM goal is high-level and suitable for non-technical people to conceptualize the purpose of the system. It is also good in design and supports model transformation. As a result, the combination of secure tropos and AOM can produce a comprehensive SRE method for safe application development.

Fig. 1 presents the SAOM for secure web application engineering. The SAOM consists of 6 phases based on the security risk management process from ISSRM. The first three phases focus on identifying the goal of the system, the security constraints of the system, and the attack method. With these three pieces of information, the developer/modeler can use them as a reference for developing a secure design. The subsequent three phases focus on designing the system. The developer/modeler will design the system based on three pieces of information that they gathered before.

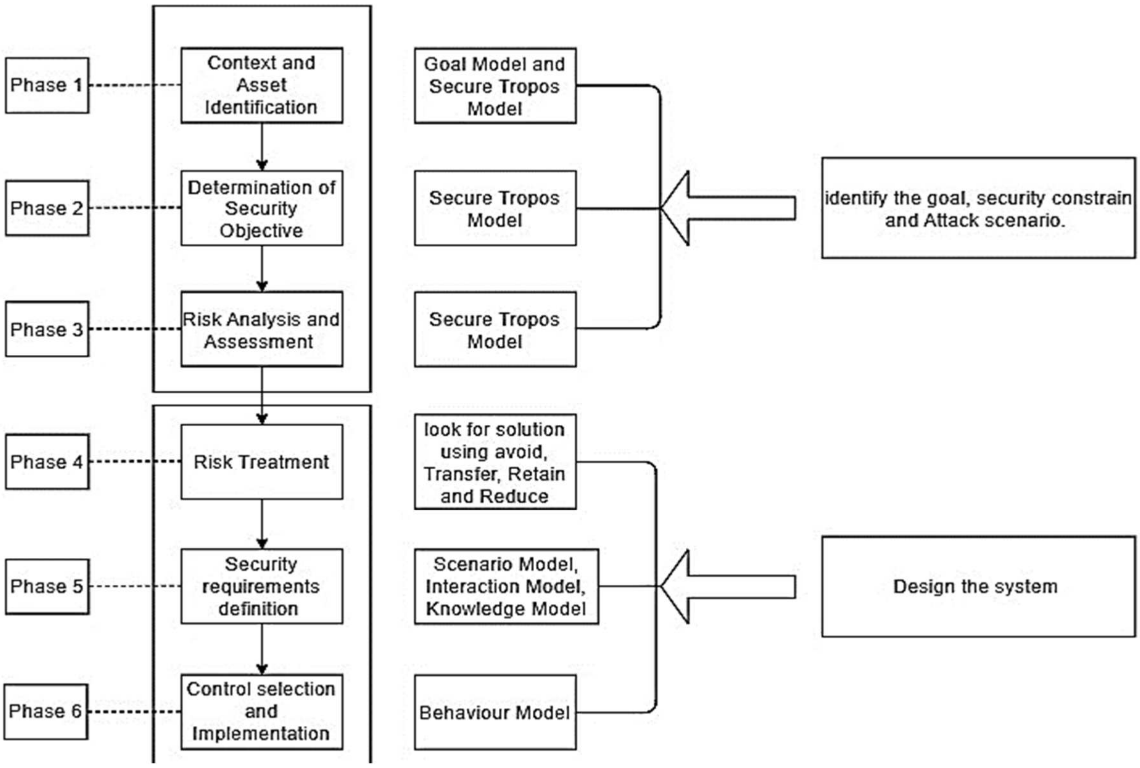

Phase 1
Context and Asset Identification
Phase 2
Determination of Security Objective
Phase 3
Risk Analysis and Assessment
Phase 4
Risk Treatment
Phase 5
Security requirements definition
Phase 6
Control selection and Implementation
Goal Model and Secure Tropos Model
Secure Tropos Model
Secure Tropos Model
identify the goal, security constrain and Attack scenario.
look for solution using avoid, Transfer, Retain and Reduce
Scenario Model, Interaction Model, Knowledge Model
Behaviour Model
Design the system

Fig. 1 Secure AOM Methodology

SAOM has two representations: a graphical diagram and a textual template. For the textual template, the Scenario Model is the model that will use a table to show the scenario between actors/agents. In the following description, the details of SAOM are presented through a walkthrough example of engineering a secure e-commerce system.

Phase 1: ***Context and Asset Identification***. Fig. 2 shows the goal model of the system. The main objective of the Goal model is to model the problem and purpose of the system. Based on Fig. 2, one primary goal model is to handle the sales and purchases. There are two subgoals: to manage buying and to manage selling. The agent that will manage buying is the Customer Agent, and the Seller Agent manages selling. Customer Agent will manage their buying by Managing the payment, Browser, and Purchase Product, the sub-goal under managing buying. Seller manages their selling by Managing the Product, inventory, and orders, which subverts the goal of managing selling.

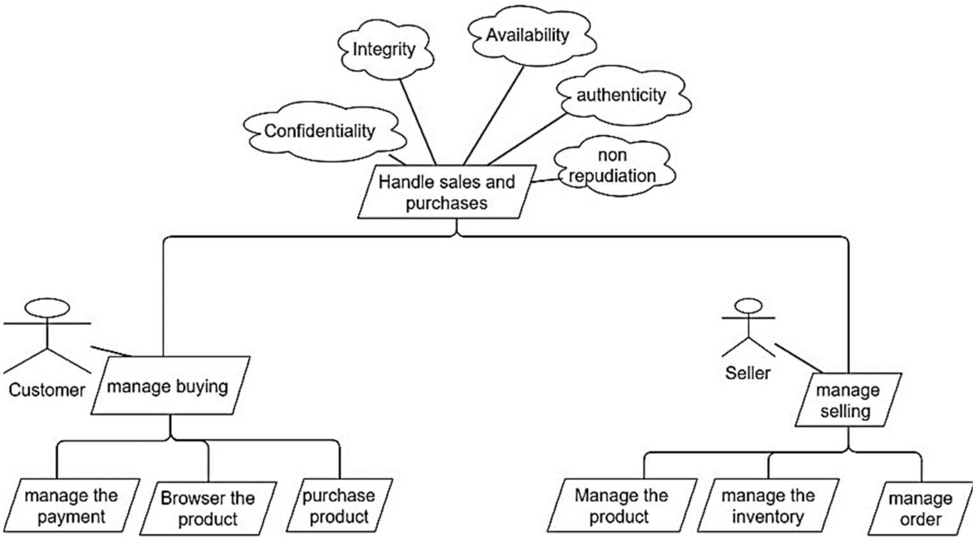


Fig. 2 Goal Model

At the primary goal, five quality goals function as non-functional goals. There are Confidentiality, Integrity, Availability, Authenticity, and Non-repudiation. Fig. 3 presents the Secure Tropos model for context and asset identification. From Fig. 3, the developer studies what the e-commerce web-based application is about. Customer and Seller will carry out their task, which is buying and selling, respectively. The developer decide that, to use this e-commerce web-based application, the user needs to create an account based on what role they want, for example, account for the Customer and Seller. The developer plans to make the account for the Customer so that the Customer is able to log in to the account. The goal of having the login is for the Customer to manage the account. This account creation leads to the storage of customer info, which will become the resource. All of this can occur only if the Customer has an account. So, the security constraint will be Only if the Customer has an account.

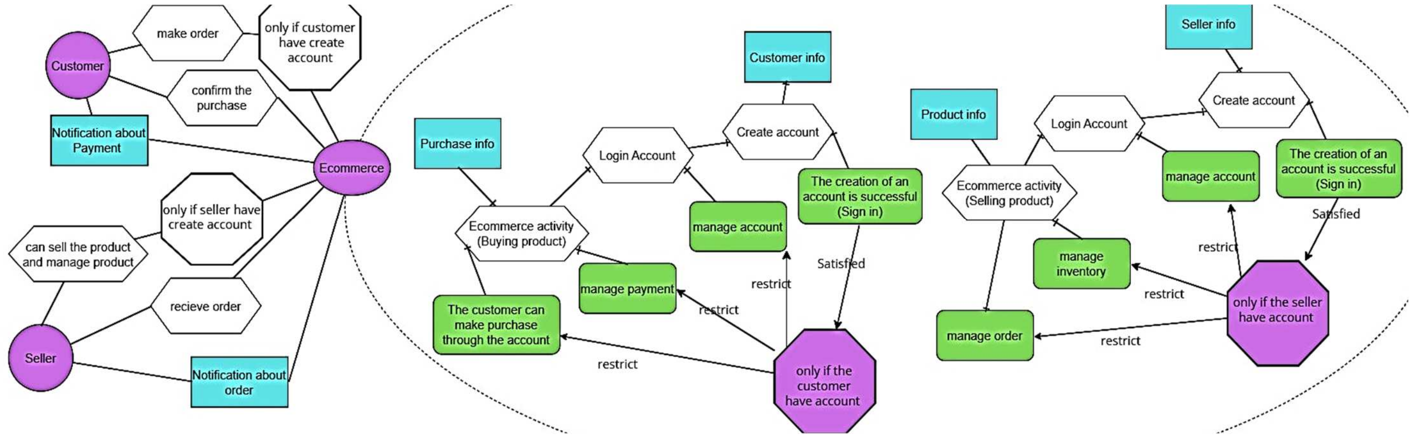
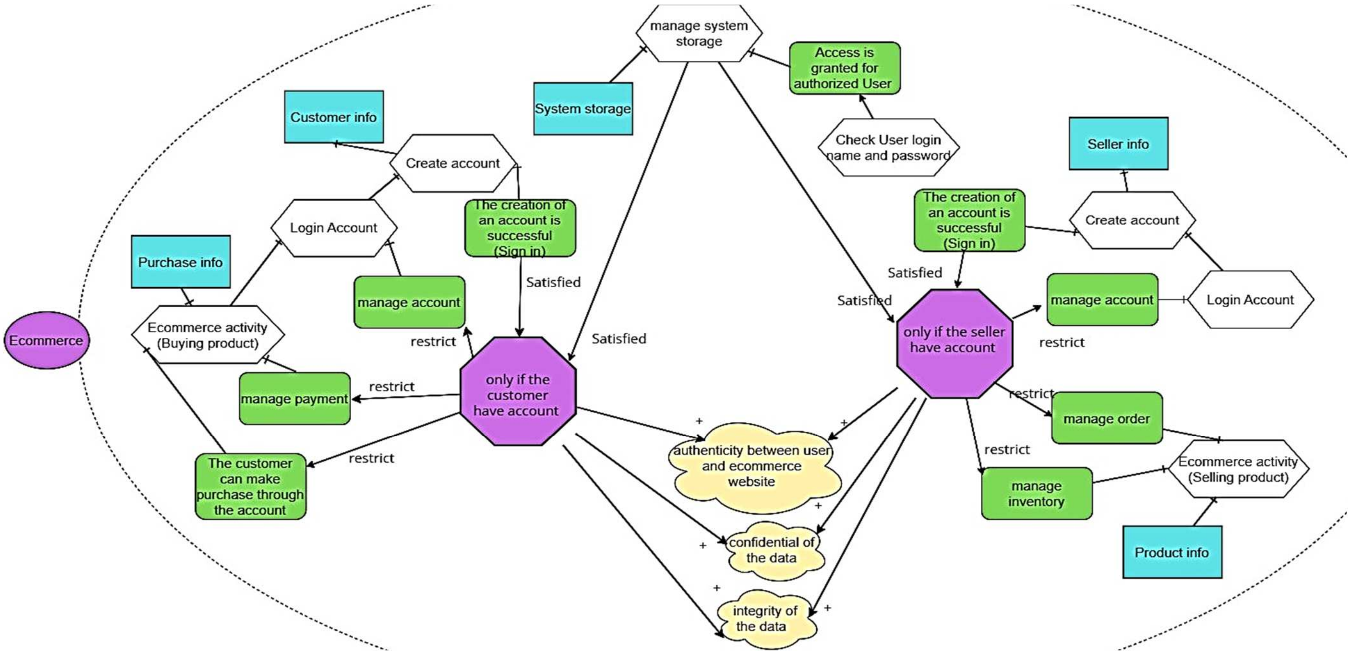

Customer
make order
only if customer have create account
confirm the purchase
Notification about Payment
Ecommerce
only if seller have create account
can sell the product and manage product
recieve order
Seller
Notification about order
Purchase info
Customer info
Create account
Login Account
Ecommerce activity (Buying product)
The creation of an account is successful (Sign in)
manage account
manage payment
The customer can make purchase through the account
restrict
restrict
restrict
Satisfied
only if the customer have account
Seller info
Product info
Login Account
Create account
Ecommerce activity (Selling product)
manage account
The creation of an account is successful (Sign in)
Satisfied
manage inventory
restrict
restrict
only if the seller have account
manage order
restrict

Fig. 3 Context and Asset Identification

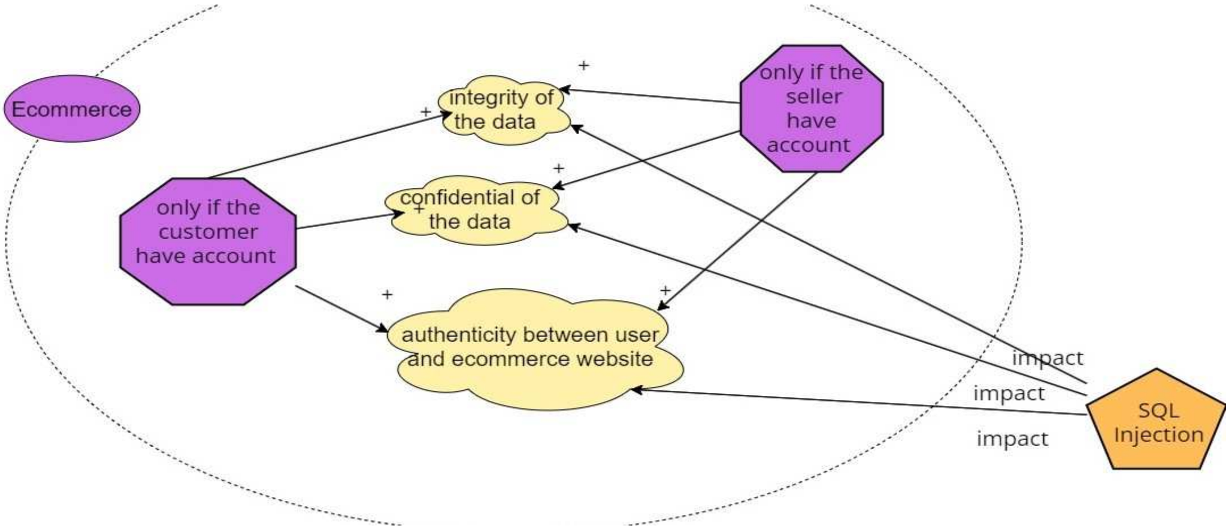
Phase 2: ***Determination of Security Objective***. In this phase, the developer needs to identify the security objective. In this case, the developer modeled the security constraint of 'only if the customer has an account' to ensure the authenticity between users and the e-commerce website, the data's confidentiality, and the integrity of the data. Here, privacy and integrity are modeled as security objectives, as shown in Fig. 4.



manage system storage
Access is granted for authorized User
System storage
Customer info
Create account
Check User login name and password
Seller info
The creation of an account is successful (Sign in)
Login Account
Purchase info
manage account
Satisfied
Ecommerce
Ecommerce activity (Buying product)
restrict
Satisfied
only if the customer have account
only if the seller have account
manage payment
restrict
manage order
The customer can make purchase through the account
restrict
authenticity between user and ecommerce website
Ecommerce activity (Selling product)
manage inventory
confidential of the data
integrity of the data
Product info

Fig. 4 Security Objective Determination

Phase 3: ***Risk Analysis and Assessment.*** In this phase (refer to Fig. 5), the developer must determine the possible attack that will attack or harm the system. The developer needs to consider the security objective identified in step (c). Assume that the developer identifies that the SQL injection can impact the security objective. In this phase, the developer must also identify how the attacker will conduct the SQL Inject attack. This will be called the attack method. This is important because it can help the developer develop the security requirement. As shown in Fig. 6, we model how the attacker conducts the SQL injection by collecting the information from the e-commerce website page. The attacker uses the information to obtain the username. The string type of the data will give the attacker an advantage in conducting the SQL injection on the Login Page. The attacker uses the username and key in any char with a payload such as "1 + 1" at the end of the username, and for the password, the attacker can put any char. After the SQL injection is successful, the attacker will collect the details about the Customer info and change the info.



Ecommerce
integrity of the data
only if the seller have account
only if the customer have account
confidential of the data
authenticity between user and ecommerce website
impact
impact
impact
SQL Injection

Fig. 5 Attack Modelling-Part 1

Phase 4: ***Risk Treatment.*** The Secure AOM does not suggest any risk treatment, but we use the ISSRM suggestion, which is based on the four possible risk treatments: Avoiding Risk, Transferring Risk, Retaining Risk, and Reducing Risk. So, in the next phase, we choose to Reduce risk by modelling the threat and solving the security problem through the Scenario Model.

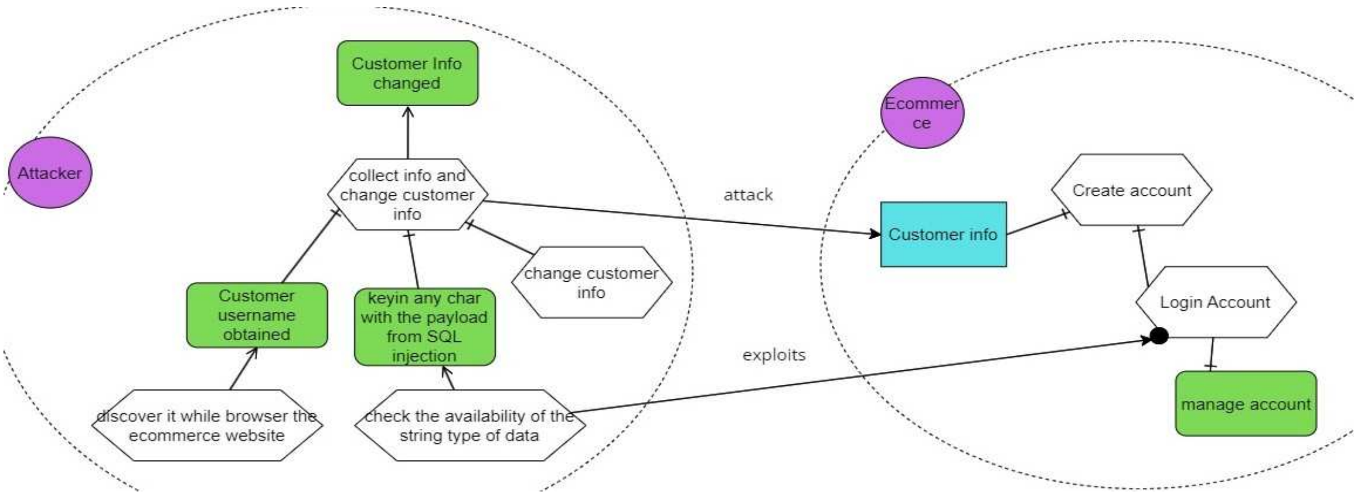

Customer Info changed
Attacker
collect info and change customer info
attack
Ecommerce
Create account
Customer info
change customer info
Customer username obtained
keyin any char with the payload from SQL injection
Login Account
exploits
manage account
discover it while browser the ecommerce website
check the availability of the string type of data

Fig. 6 Attack Modelling- Part 2

Phase 5: ***Security Requirements Definition.*** The model shows the analysis stage from phase (a) until phase (c). The developer already knows what the security constraint and attack method are. Based on the [8] study, Secure Tropos is a framework that models security using security constraints and attack methods. The developer can come up with the security requirements and design the system. In this phase, the developer needs to come out with three Agent Model: Scenario Model, Interaction Model, and Knowledge Model. The model will be explained as follows:

First, the developer needs to design the scenario model. The developer needs to identify the quality goal, initiation, trigger, condition, step, set of activities, agent type or roles, and immutable or mutable resources. For example, Table 1 shows how to ensure the Customer signs in with the correct format.

TABLE I
SCENARIO MODEL FOR SCENARIO 1

Scenario 1: Sign in the Customer with the correct format				
Goal	Create account			
Initiator	Customer			
Trigger	Don't have an account.			
Description: Customer wants to create an account.				
Condition	**STEP**	**The customer carries out activity to ensure the authenticity of users and the e-commerce website, the confidentiality of the data, and the integrity of the data.**	**Agent/ Role**	**Resource**
	1	The Customer clicks on the website link.	Customer	
	2.	The Customer will see the main page of the e-commerce	Customer	
The Customer must have no account.	3.	The Customer clicks on the sign-in button	Customer	
	4	The Customer can see the sign-in page.	Customer	
	5	The Customer needs to create the account by keying in the email/ username, password, and confirmation password.	Customer	
If the username correct format == true	6	The Customer needs to click the confirmation	Customer	
	7	The Customer will receive the pop-up message of the successful sign-in and ask the Customer to log in.	Customer	
If the username is correct format! = true	8	The Customer needs to create a username that follows the format and follow steps five until 6		
	9	The Customer will receive the pop-up message of the successful sign-in and ask the Customer to log in.		
	10	**Scenario 2**: Overcome SQL inject	Customer	

For this problem, we recycle the security problem solution called security pattern since the solution approach is signed in with the correct format. Without the correct format, the system will reject the sign-in, and the Customer needs to repeat the required format until they reach the proper format. Table 2 shows that the Customer wants to log into the system after creating the account. So, the customer is required to fill in the username that was created before. The format must be correct before the Customer can fill in the password.

TABLE II
SCENARIO MODEL FOR SCENARIO 2

Scenario 2: Overcome SQL inject				
Goal	Login into the system			
Initiator	Customer			
Trigger	The Customer already has an account.			
Description: Customer wants to create an account.				
Condition	**STEP**	**The customer carries out activity to ensure the authenticity of users and the e-commerce website, the confidentiality of the data, and the integrity of the data.**	**Agent? Role**	**Resource**
	1	The Customer clicks on the website.	Customer	
	2	The Customer clicks on the login button	Customer	
	3	The Customer sees the login page	Customer	
	4	The Customer fills in the username.	Customer	
If the username correct format == true	5.1	The Customer can proceed to fill in the password	Customer	
	5.2	The Customer clicks the confirm button.	Customer	
If the password == true	5.3	The Customer can see the e-commerce website for the customer site, browse the products, and buy the products.	Customer	
If the password! = true	5.4	The Customer can retry to key in the password for two more tries before the username block.	Customer	
If the username is correct format! = true	6.1	The Customer will be notified that the format of the username is incorrect.	Customer	
	6.2	If the Customer proceeds to the password and clicks the confirm button, the Customer will be notified that the login is unsuccessful.	Customer	

Next, the developer needs to make the interaction model. The interaction model is an interaction pattern between the agents that is designed based on the responsibilities. Figure 7 shows that there are four main agents: customer agent, website agent, SQL database agent, and attacker agent. There will be a performing agent and a perceiving agent. First, the Customer must register before proceeding to the login. The Website Agent acts as a performing agent and requests the Customer Agent as the Perceiving Agent to register. Then, the Customer Agent becomes the Performing agent when the Customer agrees to the request. As in the register step, the Website Agent becomes the Performing Agent, which informs the Customer Agent to register using the correct format. The website agent will accept the registration proposal made by the customer agent and send it to the SQL database agent. Then, the Website Agent will inform the Customer Agent that the registration has been successful as the SQL database has accepted the correct format data. So, the Website Agent will request the Customer Agent to log in. The customer agent agrees with the request and fills in the required login information in the correct format. When the required information is in incorrect format, the website will inform us the format is incorrect. If the Customer Agent successfully fills in the necessary information with the correct format, the Website Agent will notify the Customer Agent that the login is successful.

Lastly, the developer needs to figure out what is knowledge or information of each agent necessary to carry out their behavior. It can act as an ontology that can provide the developer with a framework of knowledge on the agent's knowledge. The developer will become more understanding of the issue that occurs between the agents. In Fig. 8, there are Seller agents that know the Seller account; in Seller Account, it contains name, email, phone no, address, IdOder, and IdProduct. The customer agent knows the customer account, which includes the name, email, phone no, address, payment, and idOrder. When the Customer makes the Order, the Customer must complete the payment through the app, which contains payment, price, date, and time. After the payment, an order will be created that contains the IdOrder, IdPayment, date, and time, which the Seller agent will know through the Seller Account.

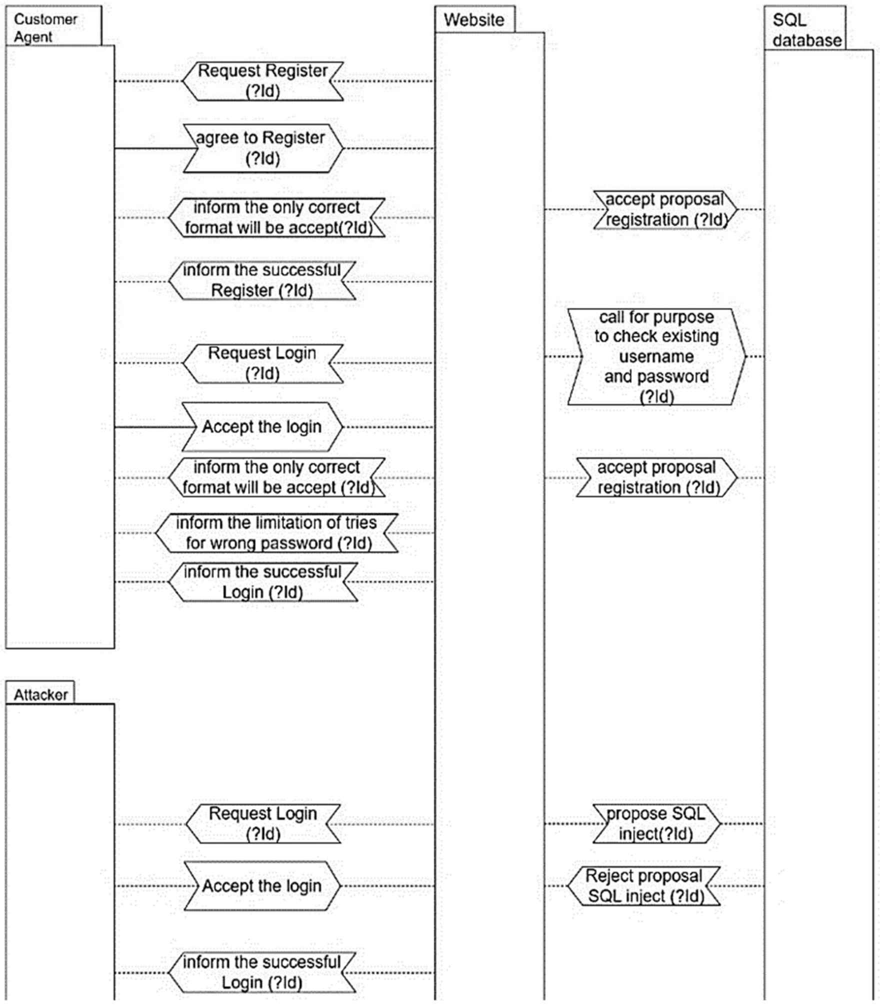

Customer Agent
Website
SQL database
Request Register (?Id)
agree to Register (?Id)
inform the only correct format will be accept(?Id)
accept proposal registration (?Id)
inform the successful Register (?Id)
call for purpose to check existing username and password (?Id)
Request Login (?Id)
Accept the login
inform the only correct format will be accept (?Id)
accept proposal registration (?Id)
inform the limitation of tries for wrong password (?Id)
inform the successful Login (?Id)
Attacker
Request Login (?Id)
propose SQL inject(?Id)
Accept the login
Reject proposal SQL inject (?Id)
inform the successful Login (?Id)

Fig. 7 Interaction Model

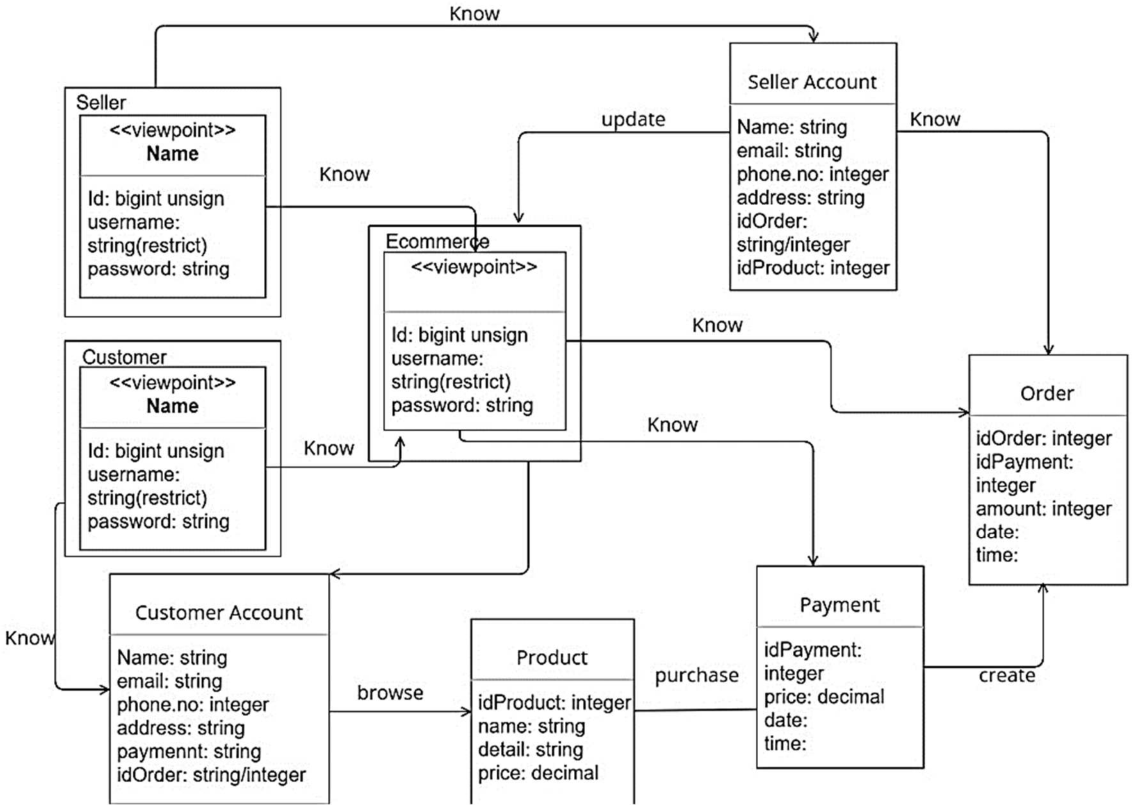


Fig. 8 Knowledge Model

Phase 6: ***Control Selection and Implementation***. We introduce a behavior model in this phase. The behavior model is an activity and behavioral interface that expresses the trigger, precondition, and postcondition to perform the activity. There are three types of actions: communication action, physical action, and epistemic action. The activity always starts with the start event, and any precondition needed before the activity can be carried out will be known as the rule, and sure of it will have the condition. If any activities have multiple choices that lead to various postconditions, then it will be used to indicate it. Based on Fig. 9, the Customer has already created the customer account and wants to log in. So, the Customer agent opens the website, and the website will show the customer interface. At the same time, the website will trigger the login and send the request to the Customer for the login. The Customer will handle the login by keying in the username. If the username is in the correct format, the Customer agent will proceed to the password, but if the customer key is in the wrong format, the Customer must reenter their username. If the customer successfully keys in the correct format of the username and correct password, the SQL database will check the username and password and give the response. If the username and password exist in the database, the website will notify the login as a success. However, if the username and password do not exist, the website will inform the Customer that the login is unsuccessful.

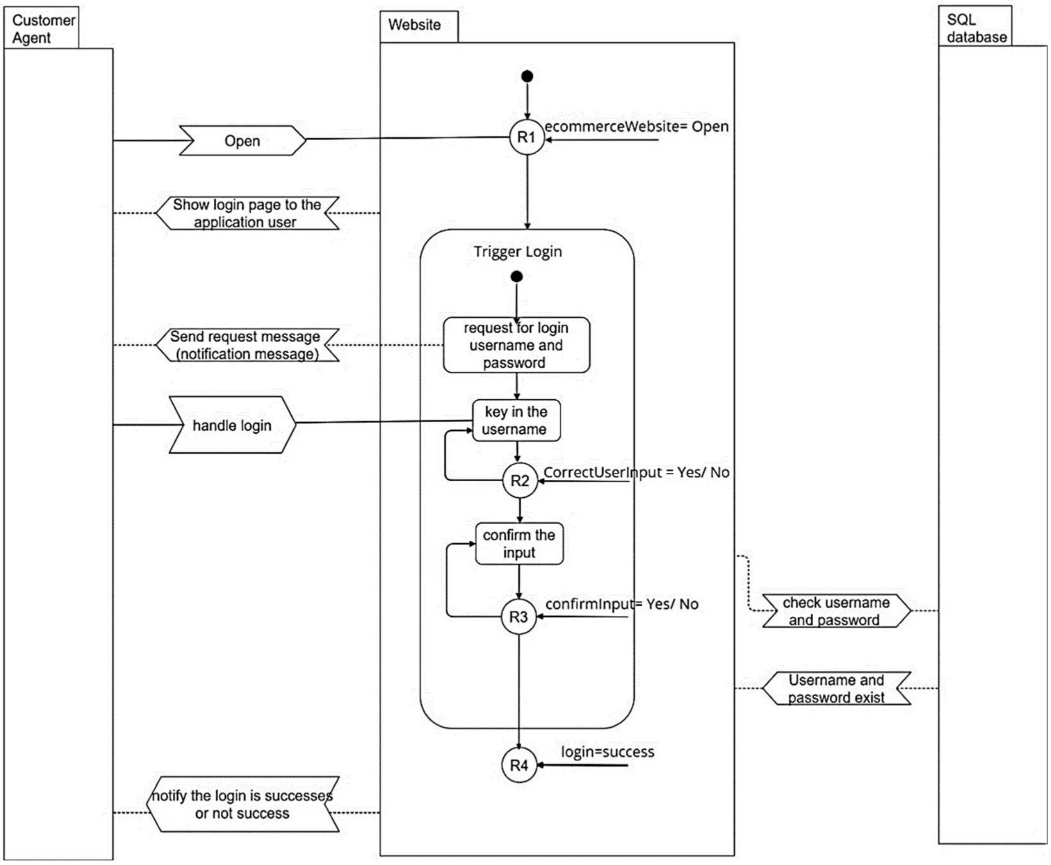

Customer Agent
Website
SQL database
Open
R1
ecommerceWebsite= Open
Show login page to the application user
Trigger Login
Send request message (notification message)
request for login username and password
handle login
key in the username
R2
CorrectUserInput = Yes/ No
confirm the input
R3
confirmInput= Yes/ No
check username and password
Username and password exist
R4
login=success
notify the login is successes or not success

Fig. 9 Behavior Model

III. Results and Discussion

In this section, we present the usability analysis of SAOM among 110 students taking TME4093 Advanced Topic in Software Engineering. The experiment aims to conduct the reliability test of SAOM. The students were asked to answer a questionnaire. Based on [29] study, reliability is "*the property of a language that aids in producing reliable programs"*. The reason for the reliability test in this study is to understand whether the participants agree or not SAOM can reduce the probability of users making errors and ensure users include the right element and correct relation between the elements. So, based on the [30] study, we reuse the question for the reliability test. This questionnaire needs to be answered through Google Docs.

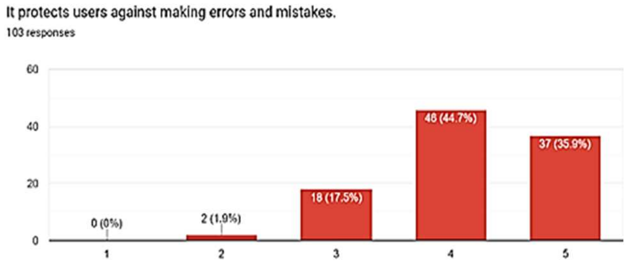


Fig. 10 Results of a questionnaire for the first question

Two questions will be settled after the participants model the Secure E-commerce web-based application. The first question is whether it protects users against making errors and mistakes. Based on the survey, 44.7% and 35.9% of the participants chose 4 and 5, respectively. This concludes that

AOM + Secure Tropos can help participants avoid making errors and making mistakes while they come up with the requirements and design of the system.

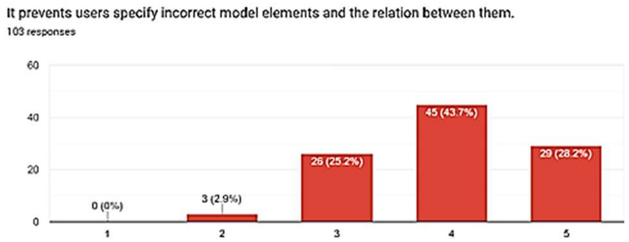


Fig. 11 Results of a questionnaire for second questions

The next question is whether it can prevent users from specific incorrect model elements and the relation between them. Based on the survey, 43.7% and 28.2% of the participants chose 4 and 5, respectively. This can conclude that AOM + Secure Tropos can help prevent users from specifying incorrect model elements and the relation between them. Secure Tropos and AOM have specific steps and models to be used while documenting the project from collecting to designing the system.

Based on the reliability question, it shows that the highest-rated agree (4-5) that AOM + Secure Tropos can prevent the user from making errors and mistakes and help participants select a correct agent model. This concludes that the participants agree that AOM + Secure Tropos can help users avoid errors and mistakes. Plus, to make sure the participants specify the correct model element.

We conclude that after we undergo the experiment and collect the data through a questionnaire, the AOM can serve as a methodology that supports security risk management. Based on the data, we know that participants believe that SAOM can help the user develop the security requirements and design a secure system.

IV. Conclusion

SAOM is a methodology that uses the ISSRM reference model to engineer a secure web application systematically. There are 6 phases involved in SAOM: *Context and Asset Identification, Determination of Security Objective, Risk Analysis and Assessment, Risk Treatment, Security Requirement Definition,* and *Control Selection and Implementation.* Each phase has its model to represent the analysis of a secure system's security requirement and design. SAOM helps the developer/modeler to collect information regarding the system, analyze the system and security requirements, and design the system. In the future, we need to make this SAOM more mature since there is much room to improve. The first improvement we want is to align SAOM with the ISSRM reference model. This will make the SAOM more understandable for users who wish to use SAOM for their secure application development. Next, we want to try this SAOM in other domains, such as the Internet of Things (IoT). IoT is a complex system since it involves software and hardware parts. We want to see if this SAOM can protect the software and hardware part of IoT against security and physical attacks.

Acknowledgements

This project is funded by the Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak (UNIMAS).

Reference

[1] N. Kuruwitaarachchi, P. K. W. Abeygunawardena, L. Rupasingha, and S. W. I. Udara, "A Systematic Review of Security in Electronic Commerce- Threats and Frameworks," *Global Journal of Computer Science and Technology*, pp. 33–39, Feb. 2019, doi:10.34257/gjcstevol19is1pg33.

[2] Z. Wu, S. Shen, H. Zhou, H. Li, C. Lu, and D. Zou, "An effective approach for the protection of user commodity viewing privacy in e-commerce website," *Knowl Based Syst*, vol. 220, p. 106952, May 2021, doi: 10.1016/j.knosys.2021.106952.

[3] M. J. Girsang, Candiwan, R. Hendayani, and Y. Ganesan, "Can Information Security, Privacy and Satisfaction Influence The E-Commerce Consumer Trust?," in *2020 8th International Conference on Information and Communication Technology (ICoICT)*, IEEE, Jun. 2020, pp. 1–7. doi: 10.1109/ICoICT49345.2020.9166247.

[4] R. Darimont, E. Delor, P. Massonet, and A. van Lamsweerde, "GRAIL/KAOS," in *Proceedings of the 19th international conference on Software engineering - ICSE '97*, New York, New York, USA: ACM Press, 1997, pp. 612–613. doi: 10.1145/253228.253499.

[5] N. Ulfat-Bunyadi, N. Gol Mohammadi, R. Wirtz, and M. Heisel, "Systematic Refinement of Softgoals Using a Combination of KAOS Goal Models and Problem Diagrams," 2019, pp. 150–172. doi:10.1007/978-3-030-29157-0_7.

[6] E. Paja, F. Dalpiaz, M. Poggianella, P. Roberti, and P. Giorgini, "Modelling Security Requirements in Socio-Technical Systems with STS-Tool," vol. 855, Aug. 2012.

[7] D. Hatebur, M. Heisel, and H. Schmidt, "A Security Engineering Process based on Patterns," in *18th International Conference on Database and Expert Systems Applications (DEXA 2007)*, IEEE, Sep. 2007, pp. 734–738. doi: 10.1109/DEXA.2007.36.

[8] R. Matulevicius, N. Mayer, and P. Heymans, "Alignment of Misuse Cases with Security Risk Management," in *2008 Third International Conference on Availability, Reliability, and Security*, IEEE, Mar. 2008, pp. 1397–1404. doi: 10.1109/ares.2008.88.

[9] M. N. Anwar Mohammad, M. Nazir, and K. Mustafa, "A Systematic Review and Analytical Evaluation of Security Requirements Engineering Approaches," *Arab J Sci Eng*, vol. 44, no. 11, pp. 8963–8987, Nov. 2019, doi: 10.1007/s13369-019-04067-3.

[10] P. Yeng, S. Wolthusen, and B. Yang, "Comparative Analysis of Software Development Methodologies For Security Requirement Analysis: Towards Healthcare Security Practice," Aug. 2020. doi:10.33965/is2020_202006L009.

[11] E. B. Fernandez, H. Washizaki, N. Yoshioka, and T. Okubo, "The design of secure IoT applications using patterns: State of the art and directions for research," *Internet of Things*, vol. 15, p. 100408, Sep. 2021, doi: 10.1016/j.iot.2021.100408.

[12] R. A. Khan, S. U. Khan, M. Ilyas, and M. Y. Idris, "The State of the Art on Secure Software Engineering," in *Proceedings of the Evaluation and Assessment in Software Engineering*, New York, NY, USA: ACM, Apr. 2020, pp. 487–492. doi: 10.1145/3383219.3383290.

[13] G. Kavallieratos, S. Katsikas, and V. Gkioulos, "SafeSec Tropos: Joint security and safety requirements elicitation," *Comput Stand Interfaces*, vol. 70, p. 103429, Jun. 2020, doi: 10.1016/j.csi.2020.103429.

[14] G. Kavallieratos, V. Diamantopoulou, and S. K. Katsikas, "Shipping 4.0: Security Requirements for the Cyber-Enabled Ship," *IEEE Trans Industr Inform*, vol. 16, no. 10, pp. 6617–6625, Oct. 2020, doi:10.1109/TII.2020.2976840.

[15] C. W. Shiang, A. A. Halin, M. Lu, and G. CheeWhye, "Long Lamai Community ICT4D E-Commerce System Modelling: An Agent-Oriented Role-Based Approach," *The Electronic Journal of Information Systems in Developing Countries*, vol. 75, no. 1, pp. 1–22, Jul. 2016, doi: 10.1002/j.1681-4835.2016.tb00547.x.

16] S. Filzah, Z. A., W. Shiang, M. A. Khairuddin, and N. Jali, "Modeling Emotion Oriented Approach through Agent-Oriented Approach," Aug. 2020.

[17] L. Sterling and K. Taveter, *The Art of Agent-Oriented Modeling*. 2009. doi: 10.7551/mitpress/7682.001.0001.

[18] S. F. binti Zulkifli, C. Waishiang, M. A. bin Khairuddin, N. binti Jali, and Y. R. binti Bujang, "How to Model an Engaging Online Quiz? The

Emotion Modeling Approach," *Journal of Telecommunications and Information Technology*, vol. 1, no. 2022, pp. 54–63, Mar. 2022, doi:10.26636/jtit.2022.156221.

[19] L. A. Stoica and R. A. Candoi-Savu, "Math approach of implementing ISO 27001," *Proceedings of the International Conference on Business Excellence*, vol. 14, no. 1, pp. 521–530, Jul. 2020, doi: 10.2478/picbe-2020-0049.

[20] A. Alexei, "Ensuring Information Security in Public Organizations in The Republic of Moldova Through The ISO 27001 Standard," *Journal of Social Sciences*, vol. IV(1), Mar. 2021, doi:10.52326/jss.utm.2021.4(1).11.

[21] M. D. Arifin and F. Octaviani, "Occupational Health and Safety Analysis Using HIRA and AS/NZS 4360:2004 Standard at XYZ Shipyard," *International Journal of Marine Engineering Innovation and Research*, vol. 7, no. 3, Sep. 2022, doi:10.12962/j25481479.v7i3.14151.

[22] G. R. H. Aji, D. DA Putranto, and I. Juliantina, "Health and Safety Analysis of Light Rail Transit Projects in Palembang," *J Phys Conf Ser*, vol. 1198, no. 8, p. 082017, Apr. 2019, doi: 10.1088/1742-6596/1198/8/082017.

[23] D. Tofan, "Information Security Standards," *Journal of Mobile, Embedded and Distributed Systems*, vol. 3, Aug. 2011.

[24] G. Farid, N. F. Warraich, and S. Iftikhar, "Digital information security management policy in academic libraries: A systematic review (2010–2022)," *J Inf Sci*, p. 016555152311600, Apr. 2023, doi:10.1177/01655515231160026.

[25] N. Mayer, P. Heymans, and R. Matulevičius, "Design of a Modelling Language for Information System Security Risk Management.," in *Proceedings of the 1st International Conference on Research Challenges in Information Science*, Jun. 2007, pp. 121–132.

[26] R. Matulevičius, "Security Risk-Aware Secure Tropos," in *Fundamentals of Secure System Modelling*, Cham: Springer International Publishing, 2017, pp. 77–91. doi: 10.1007/978-3-319-61717-6_6.

[27] S. Ergasheva and A. Kruglov, "Software Development Life Cycle early phases and quality metrics: A Systematic Literature Review," *J Phys Conf Ser*, vol. 1694, no. 1, p. 012007, Dec. 2020, doi:10.1088/1742-6596/1694/1/012007.

[28] M. Ten LiBin, C. WaiShiang, M. A. B. Khairuddin, E. Mit, and A. Erianda, "Agent-Oriented Modelling for Blockchain Application Development: Feasibility Study," *JOIV : International Journal on Informatics Visualization*, vol. 5, no. 3, p. 248, Sep. 2021, doi:10.30630/joiv.5.3.670.

[29] G. Kahraman and S. Bilgen, "A framework for qualitative assessment of domain-specific languages," *Softw Syst Model*, vol. 14, no. 4, pp. 1505–1526, Oct. 2015, doi: 10.1007/s10270-013-0387-8.

[30] F. Santos, I. Nunes, and A. L. C. Bazzan, "Quantitatively Assessing the Benefits of Model-driven Development in Agent-based Modeling and Simulation," Jun. 2020, doi: 10.1016/j.simpat.2020.102126.