International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 1, January-February 2022 |

DOI:10.15662/IJEETR.2022.0401008

Security-by-Design: Threat Modeling and Requirements for Enterprise Applications

Sreenivasa Rao Basavala

Independent Researcher, Brambleton, Virginia, USA

Email: basavala@gmail.com

ABSTRACT: Enterprise applications are implemented in distributed backdrop, combined services, cloud apps, API, databases, and diverse user interfaces, which present complex and dynamic security threats. To minimize vulnerability and enhance resilience in an organization, incorporation of security consideration in the development of the applications is thus indispensable. A systematic threat discovery and security requirement description throughout the enterprise application development life-cycle forms a structured framework of Threat Modeling and Security Requirements Engineering (TSM-SRE) which is introduced in this research paper. The proposed model includes the best ways of determining asset and the attack-surface analysis techniques, threat classification techniques, risk assessment techniques, derivation of security requirement techniques and prioritization of requirement techniques to develop a trace of identified threats and the security control taken to manage these threats. The method enables the continuous improvement of security requirements as the application architecture, data flows, dependencies and threat landscape evolve. In order to test the framework, a formal evaluation is done which assesses the framework on the aspects of threat coverage, requirement completeness, traceability, the effectiveness of prioritization, and security assurance. The findings indicate that a hybrid of threat modeling and security requirements engineering may lead to greater security risk pre-detection, reduce the discontinuities among the architectural threats and development requirements and provide the security decision-making process with more consistency. The framework offers a practical foundation about the integration of security in the development of enterprise applications in addition to aiding risk-aware design, automated validation, and ongoing improvement of security.

KEYWORDS: Threat Modeling, Security Requirements Engineering, Enterprise Application Security, Risk Assessment, Attack Surface Analysis, Secure Software Development

I. INTRODUCTION

Enterprise applications nowadays are highly integrative software ecosystems that formulate web services, mobile interfaces, application programming Interfaces (APIs), databases, cloud services, third-party services, identity providers, distributed infrastructure. This complexification of architecture adds to the attack surface and introduces security dependency among the application components, the communication and data store and external services. Accordingly, security ceases to be an issue at testing stage or even an issue at deployment stage. This ought to be incorporated in the software development life cycle beginning with architectural planning and requirements engineering. The threat modeling is a structured process that is employed to identify the potential threats and understand that the application assets, trust degrees, interfaces and components could be exposed to security hazards [1].

Threat modeling has gained a lot of research focus as an approach to determining and assessing security risks prior to the implementation of applications. Available literature shows that threat modeling consists of various techniques, such as asset-based analysis, attack-based modeling, and analysis of architecture, vulnerability analysis, and categorization of threats in a manorial manner [1]. The techniques enable development teams to reason about the occurrence of attacks at the earlier stages of system design. Enterprise applications have been observed as particularly prone to early attacks, as architecture security vulnerabilities can be propagated with a large number of different components, and may be even more expensive to fix after these vulnerabilities have been exploited. Inclusion of threat modeling in building up an application can hence offer developers and architects with security intelligence that can be used to shape decisions in design prior to undertaking a lot of effort in implementation.

Threat modeling does not always imply the threats that are identified will be transformed into a security interventions set regardless of its importance. One of the main difficulties is setting up a systematic connection between threats and

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 1, January-February 2022 |

DOI:10.15662/IJEETR.2022.0401008

security requirements. Security requirements engineering focuses on the task of identification, definition, establishment of security requirements, maintenance and prioritization of security requirements which streamline system design and system implementation. In the event that threat analysis and requirements engineering are carried out separately, the results of security can be inconsistently transferred to a development requirement. This can result in unmet requirements, redundancy of controls, poor prioritization and unpleasing traceability between the risks and the safeguards that have been identified. This requires a very coordinated process in which we have threat information emerging as an obvious input into the security requirements engineering.

This threat modeling based on STRIDE provides a defense-in-depth foundation of determining universal category of security threats that include spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege. Research on STRIDE has shown that it can extend to software components and architectural applications and its techniques can be used to improve threat characterization, detecting and treating them [2]. The systemic labeling will be possible to enhance consistency in the naming of threats and provide development teams with a common language in which they can discuss security threats. Nevertheless, enterprise application environments are often complex, which can have a complicated interaction between internal components, external services, users, privileged identities, and data flows. As such, threat identification should be accompanied with mechanisms of determining the relevancy and severity of threats in a certain application setup.

The other risk is related to the assumptions that are used in the context of threat modeling. Security analysis is based on assumptions made with regard to boundaries of the systems, relation of trust, component behavior, assets, user privileges, potential conditions of attacks. A bad or undocumented assumption can meddle with the reliability and completeness of threat detection. A study investigating the assumption in the threat modeling of STRIDE indicates that it is essential to gain an idea on the logic behind the threats identified [3]. Within enterprise application development, the traceability can be improved by clearly specifying how the assumptions have been used in the system architecture and security requirements, hence making the security decisions sustainable as the application is continued to grow.

Another reason why a systematic integration cannot be avoided is the proliferation of threat-modeling techniques and the tools that can be used to facilitate threat-modeling. The available tools differ in terms of modeling capabilities, the level of automation, threat data, capabilities of the analysis and the type of development tasks they support [4]. Although automated tools can accelerate the extent of threat identification, and reduce the reliance on manual resources, the deliverables of threat-modeling tools still need to be read in the penumbra of application consumptions, business functions, business risks and constraints of the architecture. It follows as a result that a successful enterprise security process entails a lot more than merely creating a list of so-called threats. It should create a systematic platform between the application resources and architecture components to threats, risks, safety necessities, controls and authentication exercises.

There are also different priorities and impact of security requirements. Enterprise applications typically deal with sensitive data and are often utilized in business-critical processes and, consequently, not all security requirements are equally important and costly to buy. A risk-conscious requirements process should look at factors such as criticality of assets, risk of threat, potential impact, the exposure to attack, regulatory requirements, and dependencies of implementation. Prioritization also enables the development teams to allocate the security resources according to the perceived risks as well as offer traceability between the security objectives and technical controls.

This paper therefore proposes formalization of Threat Modeling/ Security Requirement Engineering Enterprise Application Development. In designing the framework, application asset discovery, architectural diagnosis, threat identification, risk assessment, security requirement derivation, prioritization, and traceability have been included into a single development-oriented process. The aim is to enhance the relationship between security analysis and the software requirement such that the identified threats can be effectively transformed into a structured set of security requirements. The proposed solution will be able to facilitate the above security decision making process, requirements completeness, traceability and provide systematized base of security assurance throughout the life cycle of the enterprise application development process.

II. RELATED WORK

Threat modeling is now an important part of secure software engineering, because it enables the detection of vulnerabilities and delves into them, before they spread to operating systems. The useful systematic literature review of the threat modeling approaches provided by the authors Xiong and Lagerström [1] categorizes the presented methods

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 1, January-February 2022 |

DOI:10.15662/IJEETR.2022.0401008

depending on their objectives, aspects of modeling, and the situations of their usage. They demonstrate in their work that threat-modeling may take a number of different methods and that there is a need to have systematizations that could carry out bridging architectural knowledge as well as security analysis. The review also however indicates that most of the methods concentrate on the recognition and analysis of threats as opposed to developing a continuous contact between the identified threats and defined formal security requirements.

The idea of detecting threats in software subsystems at organized methods, grounded on STRIDE, has become very popular. Hamid, Jaskolka, and Rouland [2] proposed a methodology to specify, identify and remediate software components, which have a connection to the threat of STRIDE. Their work is a marriage between formal methods and modeling as well as tool support, which shows how it is possible to build a software architecture analysis reflecting threat identification. This provides a critical foundation on which threat analysis can be formulated along with secure software development. Van Landuyt and Joosen [3] also investigated the assumptions in STRIDE threat modeling as well as demonstrated that threat-modeling activities involve numerous implicit assumptions towards the system behavior, assets, trust equations and attacks. The findings of their research reveal that it is necessary to specify the assumptions used in threat-modelling in order to increase consistency and reliability in security-analysis.

The increased complexity of the threat-modeling operations has also stimulated the study of automated tools. Shi, Graffi, Starobinski and Matyunin [4] have designed taxonomy of threat-modeling tools and analyzed their properties and capabilities. Their work gives a systematic framework approach to comprehending variations among automated and semi- automated threat-modeling solutions. However, tool-based solutions can be largely concerned with modeling and analysis facilities, but the combination of the threat output with the requirements of the development-oriented security should be taken into account. The topic of probability distributions had been explored by Xiong, Hacks and Lagerstrmo [5] on attack simulation languages which contained mechanisms of explaining uncertainty of the behavior of the attacks. This kind of probabilistic modeling can facilitate a more knowledgeable risk evaluation, notwithstanding, more integration is needed to apply the model to the security requirements prioritization in the process of developing enterprise applications.

Some details are also given on the correlation between threat modeling and software development practices. Dhillon [6] studied threat modeling that can be done by developers and gave some practical lessons on how a threat analysis can be integrated into the development process. The research emphasizes the necessity of integrating developers in security analysis as opposed to making threat modeling a security solitary activity. This view holds special weight on the development of an enterprise application where security requirements have to be learnt and observed by several development and architecture teams.

Organizational architecture is another resourceful technique for security threat analysis. A model created by Sommestad, Ekstedt, and Holm (as presented in [7]) was labeled Cyber Security Modeling Language (CySeMoL) to be used to measure how vulnerable the architecture of enterprise systems is. Their work suggests that reason on security weakness and reliance in an enterprise environment can be achieved using an architectural representation. A further extension in this direction was by Holm, Shahzad, Buschle and Ekstedt [8] using P2 CySeMoL, a predictive and probabilistic programming language with cyber security models. Probabilistic view favors the security evaluation in the case of uncertainty and gives a platform of evaluation of risk of attack case. Nevertheless, all these methods mainly focus on the assessment of architectural security, instead of the end-to-end method of transformation of threat discoveries to prioritized security requirements.

Another well-known algorithm in representing vulnerability relations and attack paths is attack graph. The combination of vulnerability measures in an enterprise network was tested using attack graphs by Homer et al. [9]. Their approach supports quantitative analysis of enterprise security through the realization of interactions between vulnerabilities and any possible attack vectors also mutually. These methods can enhance the risk-based prioritization but the conversion of attack-graphs outcomes to real-world application-level security requirements is still a research area of interest.

Threats to knowledge representations have also been addressed as reusable taxonomies and pattern-based methods. An extensible distributed system security threat pattern and taxonomy was suggested by Uzunov and Fernandez [10]. Their mission eases the methodical categorisation and reuse of the menace comprehension in the distributed arrangements. The Frydman et al. [11] studied how this risk analysis of the software design model can be automated, and the design information can be systematically studied to identify security risks. Those methods develop useful building blocks of automated security analysis but allow enhancing integration of traceability between the architectural threats, risk assessment and development requirements.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 1, January-February 2022 |

DOI:10.15662/IJEETR.2022.0401008

Methodologies related to risks also extend the scope of the threat analysis beyond a single vulnerability. Bedi et al. [12] came up with a multi-agent security system that manages threats based on a threat-oriented framework. Their approach proves the use of threat information and its combination with risk-management processes. Meszaros and Buchalcevova [13], have founded the OSSF online service cybersecurity risks management scheme, that is dedicated to systematizing the management of cybersecurity risks in the service environment. Such studies support the significance of correlating the technical threat data with organizational risk-management decisions.

Conceptual approaches, previously, are useful to generate threat-analysis methodologies putting them in an orderly manner. Attack-based modeling has been studied by Al-Fedaghi and Moein [14] as a modeling tool of security threats and attack interrelationships. The PowerLang design [15], a probabilistic attack simulation language, was suggested by Hacks et al., and demonstrates how the behavior of an attack could be simulated and modeled, found on probabilistic simulation methods. Introduced within a specific area, its modeling principles show us how attack simulation can be utilized to help with more systematic security analysis.

Overall, there exists a base of solid foundations in current literature to identify a threat, examine the STRIDE, assess enterprise architecture, examine an attack-graph, perform simulates probabilistically and conduct automated risk analysis. However, the literature demonstrates that we still need to come up with a comprehensive process that meets the gap between enterprise application architecture, systematic threat modeling, risk prioritization, security requirement derivation and requirement traceability as one development lifecycle. The research proposed fills this gap by placing threat modeling and security requirement engineering as integrations, instead of autonomous security processes.

III. ARCHITECTURE-AWARE THREAT MODELING AND SECURITY REQUIREMENTS ENGINEERING FRAMEWORK

The proposed **Architecture-Aware Threat Modeling and Security Requirements Engineering Framework (ATSRF)** provides a structured approach for integrating threat identification, risk analysis, and security requirements engineering into enterprise application development. The framework is designed to address a key limitation of conventional security practices in which threat modeling and security requirements are often treated as separate activities. ATSRF establishes a continuous relationship between application architecture, assets, threats, risks, security requirements, and mitigation controls. This ASTRF framework is shown in figure 1. The framework uses application architecture as the central source of security context and progressively transforms identified security risks into prioritized and traceable security requirements.

3.1 Framework Objectives and Design Principles

It is driven by five key objectives that involve systematic discovery of assets that have security relevance, Threat identification oriented on architecture, risk prioritization of a context specific way, systematic derivation of security requirements and the bidirectional traceability of threat and requirement. These goals allow consideration of security to be included in the early development of applications instead of being added at the late phases of development to testing or deployment.

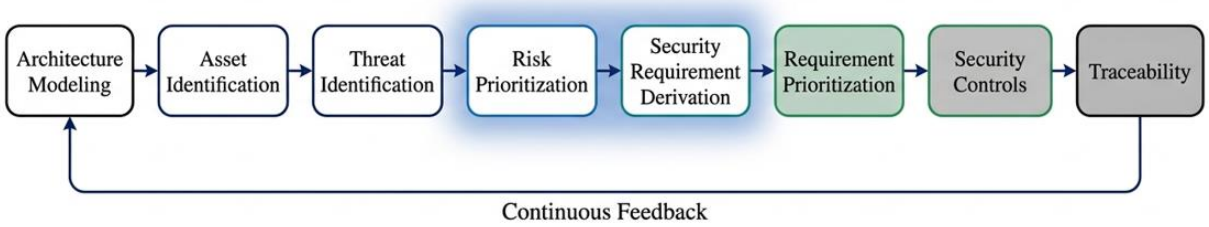


Figure 1- Proposed ATSRF Architecture

The framework has four principles of design. Architecture awareness, first, makes sure that threats are studied based on the components of the application, interfaces, and trust boundaries, data streams, authentication, and external dependencies. Second, the threat-to-requirement transformation establishes a direct relationship between the threats and the security requirements that are identified. Third, risk based prioritization will ensure that there is limited allocation

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 1, January-February 2022 |

DOI:10.15662/IJEETR.2022.0401008

of development and security funds to the risks which are most likely to be effective. Fourth, traceability aids asset-to-architecture element, threat, risk, and requirement and mitigation action in the development lifecycle.

3.2 Enterprise Application Context Modeling

The first step characterizes a security-oriented business application image. This step is elaborated in figure 2. The proposed framework signifies both the architectural functional aspects and security relevant aspects than the logical method of the traditional requirement analysis which concentrates on the functionality services primarily.

The application is further categorized into major components in order to integrate user interfaces, application services, APIs, databases, authentication services, cloud resources, third-party services, communication channels and administrative interfaces. Flow of data between these components is also indicated together with trust boundaries between components which have different security assumptions.

Special focus is put on externally accessible interfaces and dependency as they can play a vital role in controlling the attack surface in the application. The API endpoints, the remote services, the identity providers, the database connections, the cloud interfaces, and the integrations with the third parties are therefore explicitly modeled. The initial point of threat identification and risk analysis is the above architectural environment.

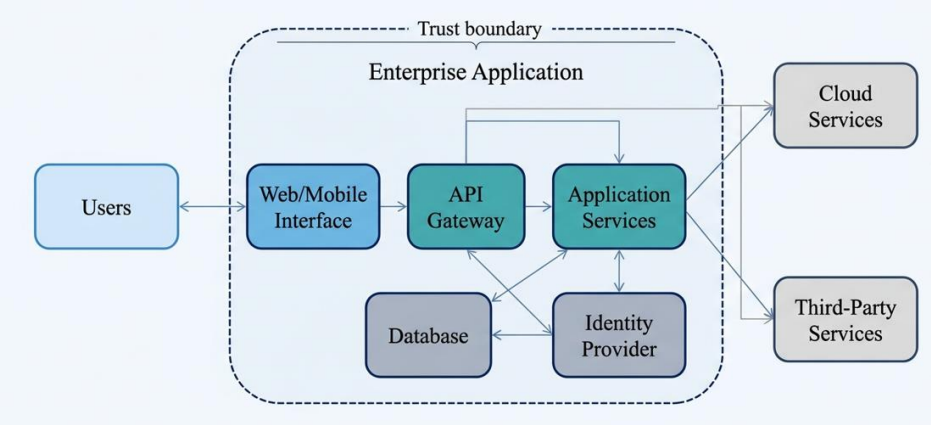


Figure 2- Enterprise Application Security Context Model

3.3 Asset and Security-Criticality Identification

The second step identifies resources that require protection, and makes them pertinent to security. Personal information, financial data, authentication, business data, application configuration, intellectual property, transaction data, source code, infrastructure resource are examples of assets.

The analysis is done on all assets in terms of confidentiality and integrity and availability requirements. Other considerations like authenticity, accountability, privacy and non repudiation can be added as per establishment of the enterprise. The criticality of the asset is determined using the business value, sensitivity, regulatory risk, operational risk and compromise ramifications.

This step enables the framework to set a difference between application resources that are ordinary, and those that are security-critical. Such differentiation helps to better analyze threats as these can then be connected to the assets and architectural features they can compromise.

3.4 Architecture-Based Threat Identification

Third phase is a procedural and systematic approach of locating all threats and threats through inspecting on the components of the applications, data flows, trust boundaries, interfaces and resources. Threat categories based on STRIDE have a systematic foundation in identifying meaningful categories of threats, such as spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege [2]. This threat modeling is shown in figure 3 below. The assumptions on which threat-modeling approaches are based are also a part of the framework since they are wrong or incomplete of determining threat identification quality [3].

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 1, January-February 2022 |

DOI:10.15662/IJEETR.2022.0401008

Detection of threats is done on the various architectural layers. At the component level, the weaknesses in regards to the individual services or resources are analyzed. On interaction level, channels of communication and API communications are examined. System Level Trust Relations: Taking into consideration are the trust relations, authentication dependencies, external services and cross component paths of attack.

Another aspect incorporated in the framework is the notion that threat-modeling workflow should be realistic and usable in the development processes. The threat-modeling literature has emphasized the importance of having a systematic approach and the sufficient tool support to frame the threat identification and analysis procedures [1], [4].

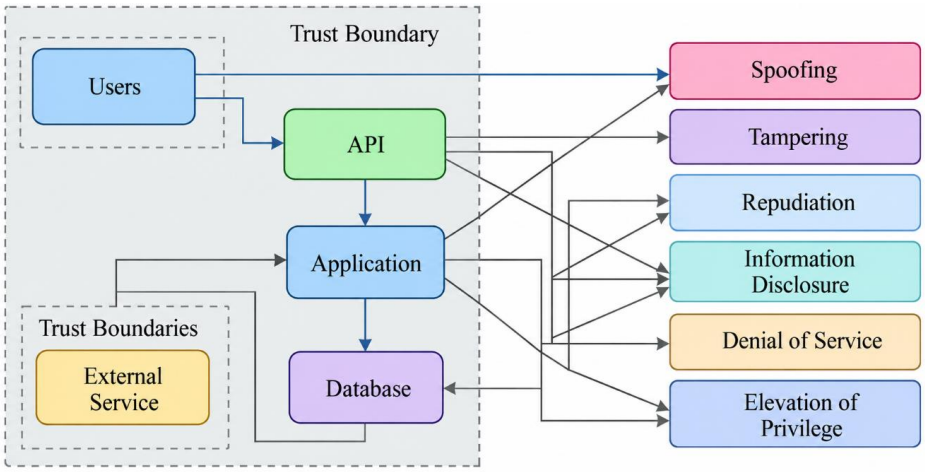


Figure 3- Architecture-Aware STRIDE Threat Mapping

3.5 Threat-to-Risk Prioritization

After threats have been stated then all threats will be rated depending on the likelihood of occurrence and impact. Likelihood considers exposure, availability, difficulty and attack and authentication conditions, dependency and availability of prior security provisions. Impact considerations include potential impacts on confidentiality, integrity, availability, business activities, regulatory requirements, and organizational resources.

The framework categorizes the risks in ranking of the risk to assist in development decisions. High-threat risks have the highest priority in the derivation process of the security requirements, and the rest medium and low-priority risks are managed according to the policies and capabilities available in the organization.

Such a risk-based technique helps to avoid the occurrence of security requirements that are an unstructured set of generic controls. Rather, it is requirements that are created due to documented security risks of a particular architectural component.

3.6 Security Requirement Derivation

The core transformation phase transforms prioritized threats into clear security requirements. Each of the major threats is mapped to requirement(s) which define the required security behavior of the system.

Using a standard example, a threat identified to be related to authentication may bring the demand of strong authentication, secure session management, the credential protection and enforcement of access-control. The risk of data-tampering may lead to the necessity to be capable of offering integrity validation, authorization, secure communication, logging, and safeguarding of transactions. Such access-disclosure threats might in turn result in encryption, data-minimization, and access-control and secure-error requirements.

Requirements are divided into requirements that are connected to authentication, authorization, data protection, secure communication, input verification, logging and monitoring, availability, privacy and secure configuration. Every requirement is expressed in the form of verifiable security condition, instead of being in the form of general security recommendation.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 1, January-February 2022 |

DOI:10.15662/IJEETR.2022.0401008

3.7 Requirement Prioritization and Traceability

The framework gives a traceability of the threat, risks, and security requirements between the architectural components and assets and the mitigation control. The security chain is presented in figure 4. This provides a security traceability bill of exchange with a view to maintaining it throughout the lifecycle of development.

The two security requirements are both associated with the threat or risk that the requirement had been created due to and the architectural element the requirement had to be imposed on. The impact analysis is facilitated by this relationship in case there are architectural changes. Whenever one of the elements such as API, data flow or even external dependency is altered, obtaining associated threats, and requirements do not need to be identified in a systematic way anymore.

There are some specific areas that are considered in a requirement prioritization, this includes the severity of risks, criticality of an asset, dependency of implementation, regulatory importance and importance to potential business. This assists development teams to find out the security requirements that are absolutely required and what can be negotiated in the later development cycles.

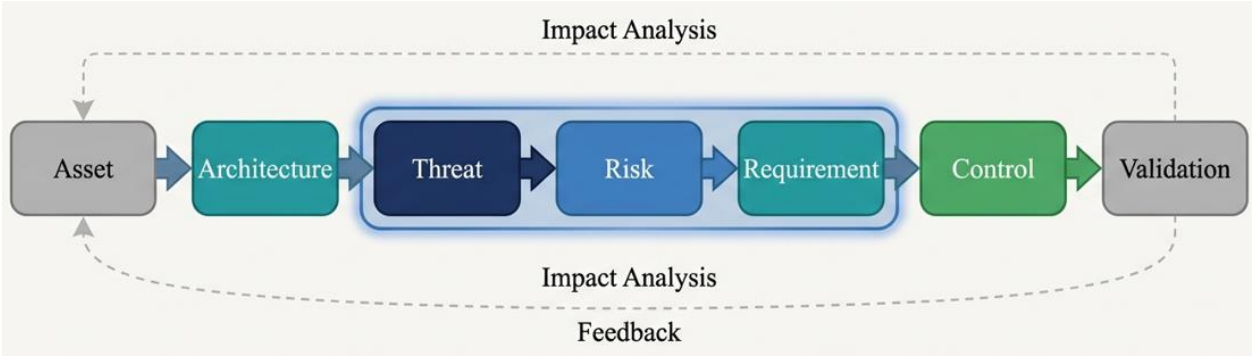


Figure 4- Security Traceability Chain

3.8 Continuous Security Feedback Loop

The last phase provides a feedback system that links the development of the security analysis with that of the applications. Continuous feedback and monitoring process is shown in figure 5. Enterprise applications frequently are modified by new functionality through business processes, APIs, user roles, and new services. Consequently, a previously developed threat model might be incomplete because of the architecture development.

The specified framework, therefore, views threat modeling and security requirements engineering as a process consisting of iterations. The architectural change initiates a reassessment of assets, trust, threats, risks as well as requirements that are affected. New threats will generate new or modified architectural security requirements and requirement changes can cause reassessment of architectural controls.

This architecture-conscious lifecycle offers an orderly approach to making security an integral part of enterprise application design whilst keeping up with the changing application architecture and security issues.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 1, January-February 2022 |

DOI:10.15662/IJEETR.2022.0401008

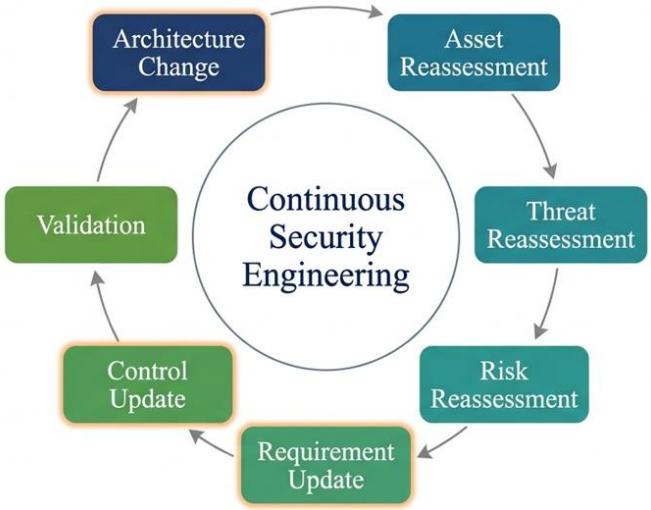


Figure 5- Continuous Security Feedback Loop

IV. FRAMEWORK VALIDATION AND SECURITY TRACEABILITY ASSESSMENT

The proposed Architecture-Aware Threat Modeling and security Requirement Engineering Framework (ATSRF) is tested to determine its effectiveness in connecting the Enterprise application architecture, identification of threats, prioritization of risks and the derivation of security requirements. The assessment is conducted based on four dimensions which consist of threat coverage, requirement completeness, traceability and risk prioritization. Threat modeling and requirements engineering are not considered as individual processes during the assessment but whether the framework has any meaningful relationships among architectural components, threat as identified, and security requirements generated.

4.1 Assessment Dimensions

The former is the threat coverage where we examine how effectively the framework is able to systematically identify the security threats on application components, interfaces, data flows, trust boundaries, and external dependencies. Cat-based categorization based on STRIDE offers a solid framework on which this evaluation can be built, whereas architecture-conscientious analysis guarantees that threats can be attributed to particular elements of an application [2]. The framework can be argued to provide a more comprehensive coverage as threats can be detected to both structural elements and resources and not just be reported as being generic security issues alone.

The second dimension is completeness of security requirement. This dimension evaluates the integration of some threats identified into expressive and operational security requirements. These are analyzed requirements that are clear, security relevant, architecturally consistent and verifiable. A requirement is regarded to be adequately defined when it defines an observable state of security which can later be met through a proper control/development endeavor.

4.2 Threat-to-Requirement Traceability

One of these critical as-assessment areas of ATSRF is traceability. Each security requirement must be allied with the menace or danger that it originated in, and each danger has to be allied with assets influenced as well as with architectural elements. This forms a traceability chain between asset → architecture→ threat → risk→ requirement→ control.

Such traceability facilitates impact analysis in case there is a change in the application architecture. Using an example of an API or authentication service, it is possible to adjust it to identify similar threats and requirements that have to be re-considered. The methodology also helps to avoid introduction of security requirements without a written security justification. Studies on threat-modeling practices and tools have shown that it is essential to have a systematic way of organizing threat information and modeling actions [1], [4].

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 1, January-February 2022 |

DOI:10.15662/IJEETR.2022.0401008

4.3 Risk Prioritization Assessment

This framework is also considered on the basis of its ability to isolate high and low security risks and concerns. In risk prioritization, aspects of asset criticality, threat likelihood, potential, architectural exposure and security controls are considered. This provides it with a background to devote resources to development of security of the high-risk aspects. Prioritization consistency is also taken into account in the assessment. Similar threats to similar assets should be treated similarly (unless there are contextual characteristics which should be treated differently). This assists in arriving at more systematic security decisions dealing with large enterprise applications with a number of components and dependencies.

4.4 Architectural Change Resilience

Enterprise applications are constantly developed and changed in terms of APIs, cloud-based services, databases, user identities, integrations and application modules. Therefore, framework assessment has its capacity to embrace security examination subsequent to architectural changes.

ATSRF also provides a feedback mechanism such that changes in architecture instigate the analysis of asset, threat, risk and needs. This feature decreases the overreliance on one-time threat models and allows ongoing security alignment over the business application lifecycle. The body of work that is concerned with interrelated threat modeling and general systematic security study is compatible with an architecture-conscious practice [1] [3].

4.5 Overall Validation Perspective

According to the framework analysis, ATSRF provides well-structured interface between body of engineering in architectural examination and security needs. The main evaluation criteria used by it, threat coverage, completeness of requirements, their traceability, risk priorities, and the ability to resist the change of architecture all present a platform against which the question of ensuring that security analysis is still in line with the evolution of business applications can be evaluated.

It does not replace the renowned threat-modeling techniques it places in a management-by-a-unified-security-needs workflow. By creating the explicit links between assets, architecture, threats, risk, requirements and controls, ATSRF will provide the methodical setup in the creation of context-motivated security requirements, which are risk-motivated and may be tracked across to the enterprise application progression lifecycle.

V. FUTURE EVOLUTION TOWARD ADAPTIVE AND CONTINUOUS SECURITY ENGINEERING

The model proposed in the following establishes an integrated foundation of interconnecting the enterprise application architecture, risk prioritization, threat modeling and security requirements engineering. It can also be improved with different extensions because of its dynamism, automation, and universality to suit the rapidly changing enterprise settings.

5.1 Continuous Architecture-Aware Analysis

The framework can be improved in the future to move the periodic assessment to continual security analysis. The updates of the APIs, cloud materials, authentication features, data set, and third-party integrations could automatically trigger a reevaluation of the affected assets, threats, risks, and requirement. This would enable security models to be kept up to date with rapidly evolving enterprise architecture.

5.2 Automated Traceability and Requirements Management

A special traceability system could be employed to maintain connections automatically between the architectural elements, security threats, prioritized risks, security requirements, and implemented security requirements. Lifecycle visibility and manual maintenance could be enhanced by adding requirements-management, issue-tracking, DevSecOps, and software-development platforms.

5.3 Advanced Risk Intelligence

The contextual threat intelligence, vulnerability information, and attack-pattern repositories, organization-specific historical data, and risk prioritization will be incorporated in future studies to integrate contextual threat intelligence, vulnerability information, and attack-pattern repositories. This might enable risk assessments to capture dynamic conditions of threat as opposed to only using the architectural data which is not dynamic. Probabilistic modeling may also be helpful in more detailed analysis of complex attack situations in an enterprise.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 1, January-February 2022 |

DOI:10.15662/IJEETR.2022.0401008

5.4 Empirical Validation Across Domains

Further confirmation on other areas such as healthcare, banking, education, e-commerce system, and government application, may help to further test the applicability of frameworks in other contexts of regulation, architecture and functioning. Scalability to large microservice and multi-cloud and API-intensive environments can also be considered in future research because scalability increases the usefulness of the framework to current security engineering of enterprises.

VI. CONCLUSION

Enterprise applications are becoming better deployed on distributed architecture which have APIs, cloud service, databases, identity providers, third party platforms and connected-up applications components. This intricacy instills a need to methodological security engineering solutions that can illuminate the gap between the architectural examination and methodological threat recognition as well as pragmatic security needs. To satisfy this need, the proposed Architecture-Aware Threat Modeling and Security Requirements Engineering Framework (ATSRF) combine these activities as the part of the structured and traceable security engineering process.

The framework will start by modeling and defining security-critical assets of enterprise application context. It then explores architectural, data, interfaces, and trust boundaries to identify potential threats. The risks are then better prioritized based on contextual risks attributes in order that security requirements may be generated given on the recorded security concerns relative to the generic security practices. Traceability among assets, architectural components, threats, risks, security requirements and security controls is also available in the structure. This communication provides systematic underpinning to security alignment based on the modifications of application architectures.

The evaluation framework takes into account threat coverage, completeness of the security requirement, threat-to-requires traceability, risk priority and stability to architecture changes. These dimensions offer a methodical foundation in order to look at whether security analysis continues to be intertwined with the changing environment of enterprise applications. The approach also helps in repetition of post processing which will enable the newly introduced elements, interfaces, and dependencies to produce comparable analysis of security.

The advantage of the framework is that threat modeling is not an independent security process that has to be handled by security requirements engineering to improve the quality of the requirements development. Its architecture-aware and risk-informed structure can assist development teams to find security concerns earlier, prioritize their requirements with respect to contextual risk, and have a clearer comprehension of security decisions/application architecture relationships. The extensions that will be introduced in the future will also have the capacity to assist in AI-based threat detection, a continuous architecture-iterative analysis, automatic tracing, threat intelligence, and validation in various enterprise domains. The adaptability and scalability of the framework can also be improved through such developments. On balance, ATSRF offers a novel framework of incorporating threat modeling and security requirement mapping in modern application development of enterprises.

REFERENCES

[1] Xiong, W., & Lagerström, R. (2019). Threat modeling—a systematic literature review. Computers & Security, 84, 53–69. https://doi.org/10.1016/j.cose.2019.03.010

[2] Rouland, Q., Hamid, B., & Jaskolka, J. (2021). Specification, detection, and treatment of STRIDE threats for software components: Modeling, formal methods, and tool support. Journal of Systems Architecture, 117, Article 102073. https://doi.org/10.1016/j.sysarc.2021.102073

[3] Van Landuyt, D., & Joosen, W. (2021). A descriptive study of assumptions in STRIDE security threat modeling. Software & Systems Modeling, 21(6). https://doi.org/10.1007/s10270-021-00941-7

[4] Shi, Z., Graffi, K., Starobinski, D., & Matyunin, N. (2021). Threat modeling tools: A taxonomy. IEEE Security & Privacy, 20(4). https://doi.org/10.1109/MSEC.2021.3125229

[5] Xiong, W., Hacks, S., & Lagerström, R. (2021). A method for assigning probability distributions in attack simulation languages. Complex Systems Informatics and Modeling Quarterly, (26), 55–77. https://doi.org/10.7250/csimq.2021-26.04

[6] Dhillon, D. (2011). Developer-driven threat modeling: Lessons learned in the trenches. IEEE Security & Privacy, 9(4), 41–47. https://doi.org/10.1109/MSP.2011.47

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 1, January-February 2022 |

DOI:10.15662/IJEETR.2022.0401008

[7] Sommestad, T., Ekstedt, M., & Holm, H. (2013). The cyber security modeling language: A tool for assessing the vulnerability of enterprise system architectures. IEEE Systems Journal, 7(3), 363–373. https://doi.org/10.1109/JSYST.2012.2221853

[8] Holm, H., Shahzad, K., Buschle, M., & Ekstedt, M. (2015). P² CySeMoL: Predictive, probabilistic cyber security modeling language. IEEE Transactions on Dependable and Secure Computing, 12(6), 626–639. https://doi.org/10.1109/TDSC.2014.2382574

[9] Homer, J., Zhang, S., Ou, X., Schmidt, D., Du, Y., Rajagopalan, S. R., & Singhal, A. (2013). Aggregating vulnerability metrics in enterprise networks using attack graphs. Journal of Computer Security, 21(4), 561–597. https://doi.org/10.3233/JCS-130475

[10] Uzunov, A. V., & Fernandez, E. B. (2014). An extensible pattern-based library and taxonomy of security threats for distributed systems. Computer Standards & Interfaces, 36(4), 734–747. https://doi.org/10.1016/j.csi.2013.12.008

[11] Frydman, M., Ruiz, G., Heymann, E., César, E., & Miller, B. P. (2014). Automating risk analysis of software design models. The Scientific World Journal, 2014, 1–12. https://doi.org/10.1155/2014/805856

[12] Bedi, P., Gandotra, V., Singhal, A., Narang, H., & Sharma, S. (2013). Threat-oriented security framework in risk management using multiagent system. Software: Practice and Experience, 43(9), 1013–1038. https://doi.org/10.1002/spe.2133

[13] Mészáros, J., & Buchalcevova, A. (2017). Introducing OSSF: A framework for online service cybersecurity risk management. Computers & Security, 65, 300–313. https://doi.org/10.1016/j.cose.2016.12.008

[14] Al-Fedaghi, S., & Moein, S. (2014). Modeling attacks. International Journal of Safety and Security Engineering, 4(2), 97–115. https://doi.org/10.2495/SAFE-V4-N2-97-115

[15] Hacks, S., Katsikeas, S., Ling, E., Lagerström, R., & Ekstedt, M. (2020). PowerLang: A probabilistic attack simulation language for the power domain. Energy Informatics, 3(1), 1–17.