Security-by-Design in API-Driven Systems: Threat Modelling and Mitigation Approaches

Author Name: Deepak Singh
Affiliation: Gainwell Technologies, USA
Role: Advisory Solution Architect
Email: deepaksingh1981@gmail.com

Abstract: ***The purpose of the study is to examine the security-by-design threat modelling and mitigation approaches in API-driven systems. The findings show how APIs are experiencing high degrees of vulnerability during data exchanges and transmissions. The findings reveal how threat modelling has a vital role to play in fighting threats to API. The threat modelling is anticipating the threats that can occur within a system. Successful mitigating measures can be crafted based on the knowledge derived from threats. The APIs can be made secure with use of Security-by-Design as per the examination. The recommendations include the inception of security measures during development stages to avoid threats.***

Keywords: Keywords: Security-by-Design (SBD), API Security, API Vulnerabilities, Threat Modelling, API Threats, Mitigation Approaches, API Security Framework, Application Programming Interface (API), Security Architecture, Secure API Development, API Attack Types, Cybersecurity in APIs

I. INTRODUCTION

A. Background of the research

The Security-by-Design (SBD) in an API-driven system integrates the security measures directly during the design stage of APIs. This is in contrast to adding them later. The integration of security measures beforehand is building a more secure framework. The reduction of breaches and vulnerabilities is possible through it. The API connections are facing numerous security threats. The service-to-service and server-to-server connections are posing critical difficulties for API [1]. The threat modelling and mitigation approaches are essential for APIs.

B. Overview

The application of SDB in API-driven systems will strengthen the threat models. The using of agile development in API often does not have the integration guidance vital for tackling the threats to the system. The designing of extensions with improper restricting of the resources can lead to the denial-of-service attacks [2]. The API faces a set of vulnerabilities that need to be managed appropriately [3].

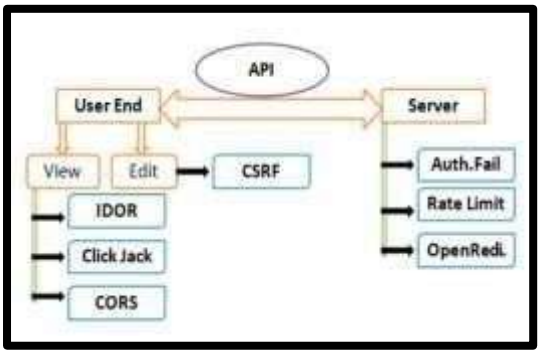

Figure 1: Categories of vulnerability in API

(Source: [3])

There is a set of vulnerability categories in API encompassing the user-end and server [Refer to Figure 1] [3]. The various websites are making use of vulnerable APIs. There are 58% high-risk web applications [3]. The evidence indicates the need for SBD-empowered threat modelling and mitigation approaches for API. The threat models can

Doi:10.48047/ijiee.2023.13.4.11

ward off the various attacks faced by the API systems. **C. Problem Statement**

The API is increasingly facing threats in the applications. The API abbreviated for Application Programming Interface is integrating various systems, services and applications [4]. There is an urgent need for dynamic models of threat handling with the continuous changes in the threat landscape. The use of SDB in threat modelling and mitigation approaches can help API to be secure. The possibilities of hacking are significantly reduced with SDB. The viability of SDB and its impacts will help various organisations to ensure secure interaction.

D. Objectives

The objectives for the present study are as follows: 1) To examine the various threats faced by API in its applications 2) To analyse the impacts of Security-by-Design on improving the security of API 3) To identify the SDB inclusion in threat modelling and mitigation approaches benefitting APIs.

E. Scope and Significance

The scope of the research is to assess the applications of Security-by-Design that can develop secure API. The threat models and mitigation approaches by SDB that can ward off attacks will be analysed. The purpose is to develop a more secure design by examining the features of SDB. The study is significant as it will pinpoint the important changes needed in API development. The various software companies can benefit from the threat models that will ensure secure interactions.

II. LITERATURE REVIEW

A. Vulnerabilities faced by API

The API initiates interactions within the multiple components of a system. However, it is vital to note how API is vulnerable to a set of external threats and attacks. The API can receive a large number of requests from malicious parties [5]. The API will exhaust all of the resources and trigger a denial-of-service attacks. There will be outages with the various components unable to interact with each other.

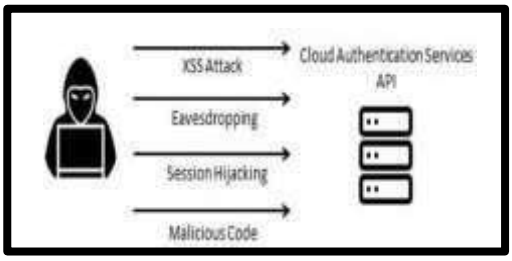

Figure 2: Types of attacks faced by the authentication services of API

(Source: [2])

There can be multiple attacks regarding the authentication services of API [Refer to Figure 2] [5]. There will be disruption in the consistent availability of services. The integration of proactive security in API can yield positive results for software companies. There are login attacks taking place in API jeopardising the overarching security of the systems [6]. The attacks can be using stolen credentials or feeding large volumes of random data within the system. The increase in vulnerable points in API is one of the major threats faced by the system. The APIs are being extensively used in the present software applications [7]. The extensive usage is increasing the scope of attacks amplifying the risks. The attackers are constantly looking for flaws that can be used to their advantage. The systems infused with targeted reinforcement can ward off the important threats faced by API. The injection attacks, anomaly, intrusion and denial of service can be tackled with reinforcement learning [7]. The API is subject to various levels of threats and vulnerabilities that need to be tackled effectively. 80% of the web traffic are AI

Doi:10.48047/ijiee.2023.13.4.11

based establishing the need for effective threat modelling [7].

B. The use of Security by Design

The security by design includes the security concerns right from the beginning of designing API. The SDB is vital in terms of early detection of any categories of vulnerability. Thus, there is timely mitigation for APIs possible in the future. The SDB is done at the design time with the application of crucial threat models [8]. The expert-design inspecting of design at the time of development helps in making the system more secure. The security analysis done at the early stages of development identifies the threats a system can face. The guideline is necessary for the proactive inspection of threats within a system. The guidelines are pinpointing to the designers about the locations in the model that can have flaws [8]. The use of SDB is aiding in the development of more secure systems eliminating possibilities of flaws. The filtering of data flows in the threat models helps in identifying sensitive data that has not been encrypted.

The secure application can only be achieved by SDB. The systems being designed with the security aspects at the core will be able to mitigate threats [9]. The development of attacks and threats modelling will pave the way for the creation of a more secure environment. There is an urgent need to standardise the threat modelling in order to gain impact-driven results.

C. The applications within threat modelling and mitigation approaches

The threat models and mitigation approaches are playing a vital role in the proper development of secure systems. The SDB for APIs can be fortified as well with the integration of comprehensive threat models. The successful anticipating of attacks and preparing for them are possible with the applications of threat models. The listing of potential attacks, profiles of the attackers, strategies and goals can aid in developing impactful threat models across the system [10]. Security, risks and privacy are the top areas of focus when developing an effective threat model. The best threat models have built-in priorities for mitigating the threats encountered. Machine learning can fail to develop improved results in the threat models [11]. The threat models need to be integrated with robustness to deliver strong performance.

The APIs are facing a number of threats across their operations. Thus, the use of SDB with comprehensive threat modelling can reduce the attacks. However, there is a lack of threat models posing a critical challenge [10]. A classification of incidents and threats taxonomy can aid in development of enhanced threat models [12]. The APIs can benefit with the SDB listing and taxonomizing the threats. The classifying of incidents can lead to enhanced results for the APIs.

III. RESEARCH METHODOLOGY

A. Research Design

The study is using an explanatory design to understand the SDB in API-driven systems aided by threat modelling. The research is analysing the impacts of SDB and how the integration of threat models can lead to better outcomes. The explanatory research design links one determinant to another considering their specific characteristics [13]. The study is applying the explanatory design to understand how the SDB, threat modelling and mitigation approaches can benefit the APIs. The explanatory design is effective in linking the features of SDB and threat modelling with the ability to reduce API vulnerabilities. The impacts in terms of increasing the robustness of API is derived

Doi:10.48047/ijiee.2023.13.4.11

with the use of explanatory design in the study. **B. Data Collection**

The study is using the quantitative and qualitative data to analyse the threat modelling in SDB. The study is collecting qualitative data from relevant secondary sources including industry reports, journal articles and the literature. The qualitative data is benefitting the research by elaborating the approaches to SDB and threat models that can minimise the API attacks. The quantitative data is being gathered from the various graphs, charts and statistics from secondary data sources.

The precise results regarding the impacts of SDB, threat models and impacts are increasing clarity on the steps needed. The applying of both types of data are enriching the research. The quantitative data is aiding to gather exact facts assessing the validity of SDB. The qualitative data is gathering the knowledge on concepts of API and SDB.

C. Case Studies Assessment

Case Study I: Microsoft

Microsoft is using an effective Threat Modelling Tool to capture any threats during the Development Lifecycle. There is better visualisation and understanding of threats [15]. The testing and security activities in the verification phase help to develop more robust systems. The Security-by-Design for any application is benefitted through the threat modelling.

Case Study II: Apple

Apple makes use of a set of procedures for robust threat modelling across its various applications needing SDB. Across the APIs are ascertained with the system security steps. The threat modelling comprises of assessing the network security threats and preparing for them accordingly [16]. Apple makes use of secure authentication and encryption of data during the transmission of in-built devices. The company provides app security protecting its APIs. **D. Evaluation Metrics**

The study is making use of certain evaluation metrics to assess the data and reach findings. The best evaluation metrics are able to discriminate the data provided [14]. The research is using the accuracy of the results to determine the trends within the data. The accuracy of SDB and threat models in the context of reducing threats is being examined. The precise outcomes of API threats and attacks requiring SDB are being used in the study. The accuracy of the data is identifying the mitigation approaches and threat modelling needed within an API framework. There are accurate insights being extracted regarding the use of threat model practises and SDB benefitting the API applications.

IV. RESULTS

A. Data Presentation

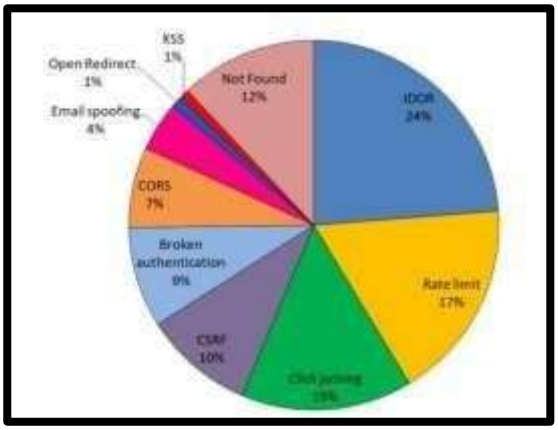

Figure 3: The detection of API vulnerabilities identified

(Source: [3])

The above chart reveals the different API vulnerabilities that can be faced during interaction. The threats of IDOR are the most at 24% [3]. The Rate Limit (17%) and Click Jacking (15%) are the other potent threats that an API can face. The listing of risks during the threat modelling will need

Doi:10.48047/ijiee.2023.13.4.11

to take into account the various vulnerabilities the API can face. The exposing of file paths or database keys needs immediate attention to ward off the threats.

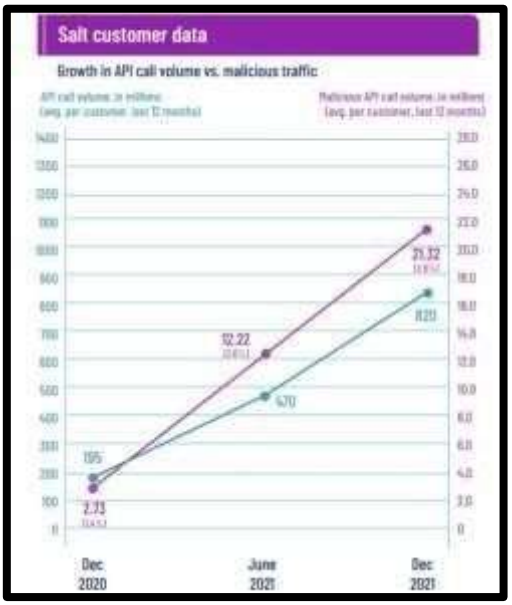

Figure 4: Increase in API attacks

(Source: [17])

There has been an increasing growth in the API volume and malicious attacks. There has been a 681% increase in API attack traffic in 2021, denoting the need for effective measures [17]. The attacks have increased to 21.32% denoting the need for impactful practices. The data is a testament to the increase in API attacks across the system requiring robust security measures to ward them off. The risks of the API attacks are large scale establishing the need for responsive designs.

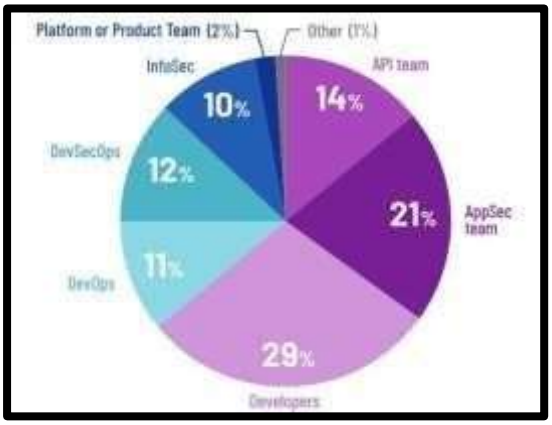

Figure 5: Steps for developing secure APIs

(Source: [17])

The above pie chart reveals the primarily responsible person for securing APIs successfully during the interactions. The developers have been identified as most responsible at 29% followed by App Specialisation Team at 21% [17]. The data reveals how the developers designing the software can ensure better security for API software. The data derives the importance of dynamic API updates. The inclusion of SBD can ensure robust designs for the software ensuring positive outcomes. The companies can develop robust models that are capable of reducing the threats and risks within the system.

B. Findings

The analysis reveals how there are a set of API attacks. The APIs are increasingly facing threats with the data exchange. The companies need to secure the systems to reach positive outcomes. There are a set of threats faced by APIs including Rate Limit, Click Jacking and IDOR. The IDOR is especially the most prevalent threat faced by the APIs [3]. The revealing of the database key or file path is included within the IDOR. Hence, there is a need for impact-driven threat modelling that takes into account the attacks posed by IDOR and ClickJacking. The SBD using threat modelling can gain

Doi:10.48047/ijiee.2023.13.4.11

positive results in terms of a robust system. The analysis further reveals how the API malicious attacks have significantly increased requiring effective measures [17]. The increase in attacks is establishing the need for more secure designs and systems for APIs.

The developers have been acknowledged as being most responsible for the APIs facing attacks [17]. Thus, developers designing the process should look for security measures across the applications. The SDB by developers can ensure that the APIs remain robust and secure compared to the outside attacks. The developers making use of effective threat modelling can develop APIs capable of warding off critical attacks during interactions. The system can attain improved results with the threat modelling and mitigation approaches integrated during the design itself. The APIs will be able to mitigate critical threats with the inception of proper measures within them.

C. Case Study Results

Case Study	***Strategy and Applications***	***Impacts***	***Results***
Microsoft	Using threat modelling during the development of applications to ensure robust security [15]	There is improved visualisation of threats leading to informed development that can minimise threats.	Increased security of applications ensuring satisfaction [15]
Apple	Using threat modelling to develop secure products with built-in encryption [16]	Devices and applications with strong and secure API reducing threats	Enhanced products and services due to the security in the network during exchanges [16]

Table 1: Case Study Outcomes

(Source: self-created)

The above table demonstrates the case studies of Microsoft and Apple making use of threat modelling in their design applications. The analysis of the case studies reveals how both companies are benefitting with secure applications and devices. The data exchanges are supported by the network protocols secured through the anticipation of threats. The threat management becomes easier with the inception of proper measures across the companies.

D. Comparative Analysis

Journal	***Aim***	***Findings***	***Gaps***

Doi:10.48047/ijiee.2023.13.4.11

[3]	The vulnerabilities of public APIs and the identification of their root causes are the aim.	The identification of a set of threats such as IDOR and ClickJacking affecting the systems has been comprehended.	The lack of analysis on the counteractive measures needed
[4]	To understand the robust approach needed for securing critical interfaces	The analysis of case studies establishing the need for securing APIs with impact-driven steps	The lack of primary data posits a critical gap within the study
[5]	To analyse the vulnerabilities in the API within the cloud computing platform	The authentication measures and algorithms for attacks have been identified within the study.	The lack of a robust framework reduces the overall validity of the system.
[7]	To establish the use of algorithms in threat detection ascertaining API security	The validity of algorithm-based threat detection established	The lack of exploring real-life case studies reduces the importance
[10]	The advantages of threat modelling in being able to protect sensitive information	The salience of threat modelling in protecting information conveyed	The use of threat modelling in Security by Design has not explored
[12]	To define the insider threat taxonomies and countermeasures	The use of a structural taxonomy that can guide threat management	The lack of a proper model for secure designs

Table 2: Comparative Analysis

(Source: self-created)

The comparative analysis of the API's security and threat modelling has been executed. The studies reveal the increasing attacks on APIs and the responsive threat modelling needed to overcome them during the design phase itself.

V. DISCUSSION

A. Interpreting Results

The results reveal the increase in API attacks requiring increased security. The use of proper threat modelling during the design

Doi:10.48047/ijiee.2023.13.4.11

can greatly benefit the system. The potential adversaries and the threats to be protected should be included in the threat model [18]. There is a need for developers to secure the system during the design itself. The anticipating of threats can lead to increased security for APIs during the exchange. The critical analysis reveals the need for Machine Learning algorithms to identify threats. The listing and classification of threats can help during the development of API [12]. There can be critical results accomplished with the use of threat modelling that can reduce IDOR or ClickJacking for the applications.

B. Practical Implications

The companies are making extensive use of API. The use of threat modelling can create more secure data exchanges [10]. The companies can derive secure results with the integration of threat modelling with security by design. The API applications will be more secure with the knowledge. **C. Challenges and Limitations**

There are certain limitations when using security by design in APIs. There can be biases in the algorithms failing to capture the dynamic updates needed. The API needs dynamic updates [17]. The threat modelling working on biased algorithms can fail to attain the desired results.

D. Recommendations

The companies need developers to integrate the counteractive security measures during the design phase. The designers have been identified as most responsible during the development of secure APIs [17]. The developers being deeply involved in identifying threats will ensure robust models. The companies should adhere to training employees for integrating threat modelling into the applications.

VI. CONCLUSION AND FUTURE SCOPE

The study reveals how APIs are facing massive possibilities of threats during data exchange. There is an urgent need for security-by-design measures. The implementation of security steps beforehand can ensure robust results. The SBD making use of threat models can ensure positive results for API. The future scope lies in assessing how threat modelling should be implemented for APIs. The listing, categorisation and measures for APIs should be identified. The future scope will aid in developing robust models.

VII. REFERENCE LIST

[1] Munsch, A. and Munsch, P., 2020. The Future of API Security: The Adoption of APIs for Digital Communications and the Implications for Cyber Security Vulnerabilities. *Journal of International Technology & Information Management*, *29*(3).

[2] Sun, R., Wang, Q. and Guo, L., 2021, July. Research towards key issues of api security. In *China Cyber Security Annual Conference* (pp. 179-192). Singapore: Springer Nature Singapore.

[3] Bhuiyan, T., Begum, A., Rahman, S. and Hadid, I., 2018. API vulnerabilities: Current status and dependencies. *International Journal of Engineering & Technology*, *7*(2.3), pp.9-13.

[4] Ranjan, P., Khunger, A., Satya, C.B.V.V. and Dahiya, S., 2022. Threat Modelling and Risk Assessment of APIs in Fintech Applications.

[5] Ariffin, M.A.M., Ibrahim, M.F. and Kasiran, Z., 2020. API vulnerabilities

Doi:10.48047/ijiee.2023.13.4.11

in cloud computing platform: attack and detection. *International Journal of Engineering Trends and Technology*, *1*, pp.8-14.

[6] Hussain, F., Hussain, R., Noye, B. and Sharieh, S., 2020. Enterprise API security and GDPR compliance: Design and implementation perspective. *IT professional*, *22*(5), pp.81-89.

[7] Ranjan, P. and Dahiya, S., 2021. Advanced threat detection in api security: Leveraging machine learning algorithms.

International Journal of Communication Networks and Information Security, *13*(1).

[8] Tuma, K., Sion, L., Scandariato, R. and Yskout, K., 2020, October. Automating the early detection of security design flaws. In *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems* (pp. 332-342).

[9] Sequeiros, J.B., Chimuco, F.T., Samaila, M.G., Freire, M.M. and Inácio, P.R., 2020. Attack and system modeling applied to IoT, cloud, and mobile ecosystems: Embedding security by design. *ACM Computing Surveys (CSUR)*, *53*(2), pp.1-32.

[10] Shevchenko, N., Chick, T.A., O'Riordan, P., Scanlon, T.P. and Woody, C., 2018. Threat modeling: a summary of available methods. *Software Engineering Institute| Carnegie Mellon University*, pp.124.

[11] Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A. and Mukhopadhyay, D., 2018. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*.

[12] Homoliak, I., Toffalini, F., Guarnizo, J., Elovici, Y. and Ochoa, M., 2019. Insight into insiders and it: A survey of insider threat taxonomies, analysis, modeling, and countermeasures. *ACM Computing Surveys (CSUR)*, *52*(2), pp.1-40.

[13] Bentouhami, H., Casas, L. and Weyler, J., 2021. Reporting of "Theoretical Design" in explanatory research: a critical appraisal of research on early life exposure to antibiotics and the occurrence of asthma. *Clinical Epidemiology*, pp.755-767.

[14] Bylinskii, Z., Judd, T., Oliva, A., Torralba, A. and Durand, F., 2018. What do different evaluation metrics tell us about saliency models?. IEEE transactions on pattern analysis and machine intelligence, 41(3), pp.740-757.

[15] Microsoft.com, 2022, *Microsoft Threat Modelling Tool*, Available at: https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling- tool [Accessed on: 14th December, 2022]

[16] Apple.com, 2022, *Apple Platform Security*, Available at: https://support.apple.com/en-in/guide/security/seccd5016d31/web [Accessed on: 29th December, 2022] [17] BleepingComputer.com, 2022, *Attacks abusing programming APIs grew over 600% in 2021*, Available at: https://www.bleepingcomputer.com/news/security/attacks-abusing-programming-apis-grew-over-600-percent-in-2021/ [Accessed on: 11th November, 2022]

[18] Li, J., Khodak, M., Caldas, S. and Talwalkar, A., 2019. Differentially private meta-learning. *arXiv preprint arXiv:1909.05830*.

Doi:10.48047/ijiee.2023.13.4.11