ESP-JETA
ESP Journal of Engineering & Technology Advancements
ISSN: 2583-2646 / Volume 2 Issue 2, June 2022 / Page No: 44-61
Paper Id: JETA-V2I2P108 / Doi: 10.56472/25832646/JETA-V2I2P108

Original Article

Threat Modeling and Risk Assessment of APIs in Fintech Applications

Piyush Ranjan[1], Akhil Khunger[2], Chalamayya Batchu Veera Venkata Satya[3], Sumit Dahiya[4]
[1]*Technology Architect Cognizant Technology USA.*
[2]*SVP CITI Bank USA.*
[3]*Sr. Enterprise Architect, Georgia-Pacific.*
[4]*Solution Architect , Barclays.*

Received Date: 22 April 2022 Revised Date: 25 May 2022 Accepted Date: 21 June 2022

Abstract: *The proliferation of Fintech applications has dramatically transformed the financial services industry, offering unprecedented levels of convenience, efficiency, and accessibility to consumers and businesses alike. This transformation is largely driven by the extensive use of Application Programming Interfaces (APIs), which enable seamless integration and communication between diverse financial systems and services. However, the increased reliance on APIs has concurrently introduced significant security challenges, necessitating a deeper examination and robust approach to securing these critical interfaces. This paper presents a comprehensive framework for threat modeling and risk assessment specifically tailored to APIs in Fintech applications. We delve into the unique security risks inherent to Fintech APIs, such as data breaches, unauthorized access, injection attacks, and denial of service (DoS) attacks, and examine the evolving threat landscape that poses continual challenges to financial data security. Through a methodical analysis of current threats and vulnerabilities, we propose effective mitigation strategies, including technical controls like encryption, multi-factor authentication, and rate limiting, alongside organizational measures such as security policies, employee training, and incident response planning. The paper incorporates detailed case studies, such as the Capital One data breach and the Plaid API vulnerability, to illustrate the real-world implications of API security failures and the necessity of rigorous security practices. Empirical analysis further underscores the critical importance of implementing robust API security measures to protect sensitive financial information and maintain consumer trust. By systematically identifying and addressing potential threats, this framework aims to enhance the security posture of Fintech applications, ensuring their resilience against cyber-attacks and fostering a secure environment for financial transactions. In conclusion, as Fintech continues to innovate and expand, the imperative for strong API security becomes ever more crucial to safeguard financial data and uphold the integrity and reliability of financial services in the digital age.*

Keywords: *Fintech, Apis, Threat Modeling, Risk Assessment, Cybersecurity, Financial Services, API Security, Mitigation Strategies, Consumer Trust.*

I. INTRODUCTION

Fintech can be defined as a growing industry that has developed as a result of technologies over the past decade and the need for innovation. This expansion has been significantly dictated by the use of Application Programming Interfaces (APIs) which are basically crucial in guaranteeing that various systems and services are integrated. APIs allow for a range of activities in Fintech applications that include payments, loan transactions, acquisition of data, and real-time monitoring. These act like a mediator between various software components so that the exchange of information between the various components of software can be done effectively and, in the process enhance the finance services. However, as helpful as APIs are in the different facets of development, they are also home to a huge number of security vulnerabilities. [1] This is because since APIs are actually virtually exposed to the external environment, an easyGREM target is provided by APIs to hackers who intend to use the existing vulnerabilities for unlawful operations. These threats of APIs are data compromise, unauthorized access, injection and denial of service (DoS) attacks. All of these threats are dangerous mainly to the content of the financial data in terms of its: authenticity, confidentially, and accessibility. Such losses may include the breach of customers' financial details and disclosure of the data and, on the other hand, Examination of unauthorized users of the financial services and fraudulent transactions. For instance, injection attacks such as SQL injection attacks enable the alteration of queries and, therefore, affect the databases' reliability; conversely, DoS attacks affect availability, causing potential strains to the financial systems.

The financial sector is more prone to these threats since the information it deals with is very sensitive, and the profits are very high if the trickery is made successfully. Thus, cybercriminals are most incentivized to attack financial institutions, processors and other entities within the Fintech theme because financial loss, legal sanctions, and reputational loss are likely to occur. Besides monetary loss, security breaches in financial services entail a reduction in the level of consumer trust and potentially compromise the stability of the financial sector. Since APIs control the process of connecting Fintech applications, it is crucial to emphasize the significance of their security. A security breach in an API basically results in massive losses in terms of money, brand image and customers' trust. For instance, leakage of customer's personal finance information exposes the financial company's users to fraud, unlawful use of identity, and major losses. In addition, when an organization experiences a security threat, it is likely to witness a negative blow to their image, which translates to customers' loss of confidence in the organization. To avoid such risks, there is a need to assess the various kinds of threats that can happen and put proper security measures in place. This encompasses the best practices as a way of securing APIs; standard security measures like encryption, proper authentication, and security audits are to be employed. Besides, organizations must develop a culture of security awareness, the practice of security audits, and management of new threats. Through proper security measures of the chosen API and concrete approaches towards security attainment, Fintech applications can manage the risks, secure the financial data and preserve the clients' confidence in the safe and effective provision of financial services in the context of the digital environment.

A. Importance of API Security in Fintech:

Therefore, when evaluating the importance of API security in Fintech, it can be considered vital because API plays the crucial role of a connector in modern finance. [2,3] APIs are the short form of Application Programming Interface, which is essential for the integration of various financial systems, applications as well as services. Some of them are payment and transfers through the use of electronics, account maintenance and servicing, managing the dealing processes, consolidation of financial data and other functions. At the same time, thanks to their importance, APIs are one of the most preferred targets of hackers, which emphasize how much protection is needed.

Figure 1: Importance of API Security in Fintech

a) Protecting Sensitive Financial Data:

APIs handle really sensitive financial information in most cases, including the PII, credit card numbers, transactions and balances. This implies that its absence or even a loss has terrible consequences, such as identity thefts, fraudulent operations, and many others that are unlawful. API security has thus become very important in order to prevent such information from being entrusted to the wrong hands or being used by other parties. Eliminating password guessing, data, and record sniffing, data and record cracking are some of the aspects that simple encryption, safe authentication, and strict access controls.

b) Maintaining Regulatory Compliance:

In the past, the government has engaged itself in placing much effort intensely into enhancing the financial sector for the benefit of the customers and the financial systems. The data protection regulation acts then, particularly the GDPR, the Payment Card Industry Data Security Standard, together with other numerous financial regulation acts existing in different countries, pose strict requirements towards data security and privacy. Thus, API security becomes one of the main factors to meet these regulations. This was due to failure to protect APIs also sparked non-compliance and thus legal ramifications like penalties and fines and an organization's reputation. To curb such repercussions, organizations must implement suitable security measures that will enable the company to meet the set standard of regulatory authorities to prove the sensitivity of handling clients' information.

c) Ensuring Service Availability and Reliability:

To the utility and availability of Fintech services, APIs are seen as rather related. The effects of DoS are on the availability of services; that is, these attacks, specifically APIs, highly influence the availability of services. This greatly affects the relationship with the customers and makes organizations incur a loss because they cannot transact or provide key services. Therefore, proper application of rate limits, controlling the traffic, and possibly the deployment of IDS systems are the matters that secure the steady availability of the mentioned services.

d) Enhancing Consumer Trust:

Confidentiality is one of the key constituents of the financial and credit business, as the consumer expects that their finance data will be processed safely. Especially the violation of security may reduce the trust of customers and seriously threaten the appeal of the organization. API protection also has a strategic role to play in regaining and sustaining customers' confidence through visible efforts to protect customers from the refer illicit access. The action of publicizing measures and practices of security, as well as, more specifically, the sessions in periods of security incidences, can also help in the process of instantiating the aspect of financial information of the consumer as secure in the eyes of the organization.

e) Preventing Financial Loss:

The monetary repercussions of vulnerability in a Fintech application are not to be underestimated. The company's costs may consist of the costs to recover from a data breach, lawyer fees, possible regulatory penalties, and expense of compensation for the customer. Moreover, there are usual secondary financial consequences concerning the company's reputation and customer loss in the long term. Therefore, in the present-day setting, by working on the API security facet and developing the safety methods, it is possible to minimize financial threats and shield the organization's financial outcomes.

f) Supporting Innovation and Growth:

In the future, when the Fintech sector emerges into new services due to the creation of new technologies, APIs are going to stay as the key in Fintech's value chain to provide and perform innovations. Thus, API security is not only the safeguarding of existing systems but also the possible advancement of APIs. Sumably, the strongly protected API is the basis for creating new financial products and services, consolidating partnerships, and expanding the audience. Hence, through API security, more organizations shall be in a position to support innovation within their firms besides ensuring they have instituted measures to foster increased development hence cooperation across different firms.

Summing up thereby, it is possible to state that API security plays a crucial role in the efficiency and sustainability of Fintech applications. Preliminary goals when it comes to API security include guarding confidential information about financial operations, addressing the obligatory requirements of supervisory bodies, ensuring the availability of services, building confidence with the clients, avoiding the risks of financial losses, and developing new products and services. Fintech systems are ever developing and expanding, and precisely for this reason, more emphasis will need to be placed on the security of APIs to ensure the continued stability and dependability of the services that are provided through Fintech solutions.

B. Evolution of Threat Modeling and Risk Assessment of APIs in Fintech Applications:

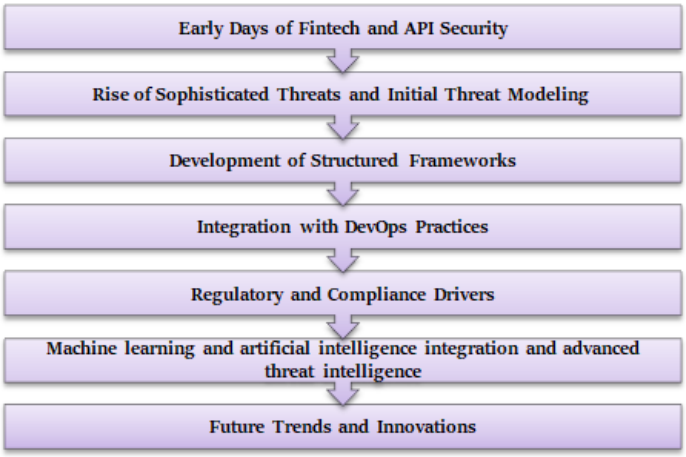

Figure 2: Evolution of Threat Modeling and Risk Assessment of APIs in Fintech Applications

a) Early Days of Fintech and API Security:

Harvey observes that in the early stages of Fintech, API was used mainly for very basic tasks inclusive of data transfer in addition to creating straight-forward financial services. [4] Security in this period is quite basic and essentially addressed at the transport level in the case of message security, in which the only security parameter is the encryption of messages, and at the application level, in which the only security concern is user authentication to gain access to the application. The industry, however, had not clearly outlined perceptions of many threats and risks that are associated with APIs. Hence, the majority of security initiatives, which were aimed at IT resources, regarded servers' and databases' security instead of APIs'. Their failure to pay attention to such risks entailed that several fintech applications were vulnerable to novel threats.

b) Rise of Sophisticated Threats and Initial Threat Modeling:

And accordingly as the structure of the fintech applications increased and these structures became more compact with a deeper detail, the increase and the advancement of the cyberattack pointed to the APIs follows a corresponding relation. API was a newly introduced weak link as menace actors had identified a way to exploit it as a means to infiltrate and steal the firm's financial details. This is the level at which was performed the first practices of threat modeling because it was considered it was necessary to avoid the escalating of the threats, which now included the features already at that level. The nature of threat models in this stage was based on the first objective, which was to identify areas of risk and vulnerability in API design and architecture. Studying the structure of the application and its relationships with other systems, some organizations started to understand where their weak points were and how they could be used, the start of more methodical protection solutions.

C) Development of Structured Frameworks:

The trends and threats related to APIs, including their complexity and malevolence, resulted in evolving better ways of structuring and systematizing approaches to threat modeling. Coherent structures and models were developed in order to get an extensive view of probable threats. These models, which include the likes of STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege), were used in API security. Concerning risk management, risk assessment methodologies were evolved to quantify the risks in terms of the possible consequences of the threats and the probability of their occurrences. This made it possible for organizations to direct their security systems to the most risky threats in a more efficient manner than before, hence improving their overall security.

d) Integration with DevOps Practices:

Then, the transition to applying security into the DevOps model, also known as DevSecOps, led to a more proactive approach to securing APIs. This integration meant that threat modeling and risk assessment were integrated into the processes and turned into one of the successive steps during the development of software. That way, with the implementation of DevSecOps practices, threat modeling and risk assessment mechanisms were built and used automatically. These tools, which include the OWASP Threat Dragon and Microsoft's Threat Modeling Tool, made the processes more efficient and easy thus making it possible for organizations to constantly assess and enhance the API security at various stages of development and deployment.

e) Regulatory and Compliance Drivers:

To a lesser extent but more pressing factors, the increasing complexity of regulations in recent years like GDPR, PSD2, and other financial regulations required stronger security for APIs. Due to these regulations, needed threat modeling and risk assessment procedures were prettified. Furthermore, common and recommended procedures in API protection included OWASP's guidelines, belonging to the Open Web Application Security Project. These standards ensured that organizations faced few issues when it came to the implementation of security standards paramount within organizations and also availed organizations with ways and means of implementing cutting-edge security standards, thus ensuring that besides the set standard regulatory requirements, vulnerable APIs and the financial data they contained were also well protected.

f) Machine learning and artificial intelligence integration and advanced threat intelligence:

The assimilation of threat intelligence into threat modeling and risk evaluation processes helped the organizations continue to remain relevant with developing threats. Fintech firms can also dramatically improve their security stance if they get real-time feeds of the perpetrators or threats in their environment, and the patterns of attacks. The application of AI and machine learning in these processes brought novelties in the management of risks. These technologies provided better techniques to predict threats and detect vulnerability, which helped the organization fight against the threats at a higher level of competence.

g) Future Trends and Innovations:

Looking forward, future advancements in API security are likely to focus on context-aware security measures that adapt to the specific needs and risks of individual fintech applications. This includes dynamic threat modeling that evolves with changing threat landscapes, providing a more responsive and adaptive security approach. Increased collaboration between fintech companies, regulators, and security experts will drive the development of more comprehensive and effective security frameworks. Sharing threat intelligence and best practices will be crucial in this collaborative effort. Additionally, emerging technologies such as blockchain may play a significant role in enhancing API security. Decentralized security mechanisms could provide additional layers of protection for API transactions and data exchange, offering new ways to secure the ever-evolving fintech ecosystem.

II. LITERATURE SURVEY

A. Overview of Fintech API Security:

Fintech is a rising and still-evolving subsector of financial technologies, being relatively recent, having emerged in the last decade majorly due to the incorporation of those technologies as well as customers' increasing expectations for better and efficient solutions. At the very heart of such evolution, one could identify Application Programming Interfaces (APIs) that give different applications a means of interacting and, in turn, help Fintech companies offer customers a comprehensive set of useful and convenient services. However, the use of APIs also involves tremendously big security risks as well.

These challenges have been described in detail in many papers in the literature, especially in the challenges that relate to the requirement for more secure solutions for API. According to the Open Web Application Security Project, commonly referred to as OWASP, API security was considered to be among the ten threats that threatened web apps. OWASP provides a detailed list of threats applicable to API, which are as follows: One of these is the broken object-level authorization in which APIs often do not verify the identity of the requesting users for certain objects. The last type is the over-exposure problem, in which APIs offer extra data than needed and become the favorites of hackers.

Also, many APIs contain less resource-like rate limiting and, as a result, can, without a doubt, be targeted through a Denial of Service (DoS) attack where the attacker overwhelms the API resources in a manner that is irritating to genuine users. These weaknesses explain why measures are being taken to ensure that Fintech companies have secure defenselessness to API hackers. API protection does not only mean rejecting malicious connections, but it also prepares the APIs to respond to a number of attacks, including injections, session purchases, and information leakage. Security has to be a high priority in the provision of APIs in the Fintech sector because data and the handling of the data are sensitive, and where there is a breach, there can be a lot of consequential damage.

B. Threat Modeling in Cybersecurity:

They are a cornerstone concept in cybersecurity this technique helps to systematize the analysis of information security threats. This process is critical since it helps in defining the security factors and threat domains of a specific system, hence allowing organizations to be in a safer position to prevent or deter the expected threats. Threat modeling is a process of developing an architectural plan for the system, analyzing threats in the system, and assessing the seriousness and probability of those threats. [11].

There are several methods for careful and fruitful structuring of the proposed process; one of the most used approaches for threat modeling is the STRIDE that Microsoft created. These are Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Services, and Elevation of Privileges. These six categories of threats show that it is possible to organize potential threats systematically and make the work of security professionals easier to focus on security issues. For instance, Spoofing threats entail the perpetrator faking to be someone else or tampering threats entailing changes to the data, which the rightful possessors did not make. Repudiation threats aim at activities which cannot be linked back to the initiator, and Information Disclosure comprises access that is unauthorized on the sensitive data.

DoS attempt the availability of service, while Elevation of Privilege is to gain privileges at a higher level than the authorized one. Another extensively used approach is DREAD, which consists of factors such as Damage, Reproducibility, Exploitability, Affected Users, and Discoverability. This framework appraises threats with these five parameters to assist organizations in handling threats depending on their potential danger and the degree of vulnerability. STRIDE is another effective approach in threat modeling and is complemented by DREAD, which with the help of two models, organizations are given the tools to predict and counteract threats that may be fatal to an organization at the hands of an attacker. In this way,

through the regular evaluation of possible threats related to organizations' systems, companies can achieve the base of security controls, which will help to decrease the probability of security threats and improve the position of security systems.

C. Risk Assessment Frameworks:

Protecting data is an essential element of any business that operates in a world of increased digital threats. It gives guidance in assessing risks to an organization's information resources and ranking the risks carried out. Risk assessment helps organizations decide on the optimum plan to address the issues arising by practicing priority on the critical risks. Many frameworks have been established to enable an organization to conduct a comprehensive risk analysis. The main two recognized frameworks are the National Institute of Standards and Technology (NIST) Special Publication 800-30 (NIST SP 800-30) and the International Organization for Standardization (ISO) 27005.

NIST SP 800-30 gives more information and structures about how to perform risk analysis. The document focuses on understanding the system environment, recognizing the assets and threats, and employing controls in order to reduce risks. Within the introduced framework, there are subdivisions of steps which are necessary for risk assessment, such as asset identification, threat sources, and risks. It also helps to make recommendations on how to measure the probability of certain threats and the effects that they might have on the organization; this assists in ranking the risks depending on the probable effects that they may have on the organization it is meant to protect. Similarly, controlling the risks of information security is described in ISO/IEC 27005 as a systematic procedure.

The framework gives more or less equal significance to all levels of information security risk which is a practical approach towards the risk management methodology. It gives tangible recommendations on risk evaluation and management, including risk control activities and control tracking for the detected risks. NIST SP 800-30 mostly and ISO/IEC 27005 also lean more into the risk management approach to cybersecurity, and this helps in prioritization. These risk management frameworks enable an organization to priorities risk and apply the required security controls to decrease the risk of security breaches within an organization's environment. These frameworks offer a clear basis for building an adequate risk management solution that will help protect organizations' information and guarantee the system's confidentiality, integrity, and accessibility.

D. Case Studies:

The following is a list of some regarding actual threats to API security in the Fintech industry, which support the necessity of a powerful protection strategy. Another example is the Capital One Company, for which an attack on the company was conducted in the year 2019. And it affected the customers through the exposure of their details to the public and was inclusive of over one hundred million customers. Narrating the provocative, owing to the breach, the main reason was identified and reported to be the misconfiguration of the web application firewall, permitting the attack that targeted the Capital One API for the purpose of data theft. According to this case, configuration for the right settings and security scans for possible vulnerability that such hostile actors could leverage are to be conducted as the basics of security.

Two more similar instances that are worth mentioning were the one when Plaid API in 2020 led to customers' millions of financial data leaks. This was especially important for the proper input validation, which, however, was lacking in the software; still, due to the fact that the API of the application was compromised by the attackers and was filled with their favourite values in the memory, they were able to get the needed information. Using this case, one can understand that in order to write a program, a programmer has to encode it in such a way that even he/she would not be able to identify the other instances of similar problems and, therefore, s/he has to perform a security analysis of the code.

This will then proceed to describe how API security threats can, in this way, impact organization and their customers and the rationale why hence API security should be a concern. The loss of such a sum and the damage done to the company's reputation may be a further cost of some of the breaches that make it compulsory that the damages be prevented, ex-ante with measures to enhance security. Therefore, by incorporating these real-life cases in organizations, it becomes easier for them to understand some risks that are associated with API security, and therefore, there will be better measures to enhance the protection of systems and data. Such cases also support knowing that security is occurring; several technical and organizational measures are required to be used to reduce the risks.

III. METHODOLOGY

A. Threat Modeling Process:

Regrettably, the systematic threat modeling which is described above to defend the kind of application in the domain of Fintech from the above-mentioned threats, current and future is absent. [14-20]This process involves the following steps:

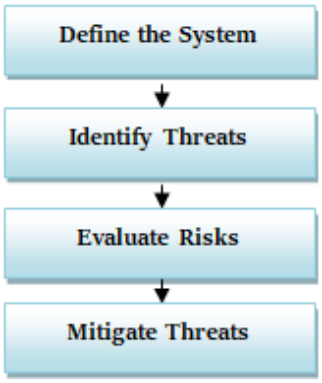

Figure 3: Threat Modeling Process

a) Define the System:

To begin with, the threat modeling process entails an understanding of the Fintech application's system. This is achieved by developing a proper DFD (Data Flow Diagram) that will display the structure of the system being worked on. The DFD is critical because it depicts the flow of data throughout the system and how the components of the system within and outside interact, for instance, the authentication services, the transaction processing units, user interfaces, and other elements such as banks and third-party financial services providers. These interactions depict possibilities of penetration points or points of entry into the system. It gives a bird's eye view of the system and reveals where the critical data is located, how it is shifting from one component to another, and where control has to be applied. This mapping is very important in establishing the basis for the threat modeling procedure as it entails an analysis of the whole system without leaving out any aspect.

b) Identify Threats:

Once the system has been described and depicted in the DFD, some threats are as follows, utilizing typical approaches such as STRIDE and DREAD. The STRIDE framework helps categorize threats into six main types: There are Spoofing (posing to be someone else), Tampering (modifying data), Repudiation (actions that can bear no traces to the attacker), Information Disclosure (unauthorized disclosure of information), Denial of Service (interference the availability of services), and Elevation of Privileges (gaining privileges that are not permitted to be accessed). Relatively, depending on the nature of the STRIDE analysis of each component in the DFD, one realizes different forms of security threats. For instance, authentication APIs may be prone to spoofing if the right measures of protection are not put in place, transaction data may be modified when integrity checks are not put in place, and information disclosure may occur if the insecure endpoints are not well-protected. Thus, this systematic identification of threats makes sure that all possibilities which can be exploited to attack the system are taken into consideration.

Table 1: STRIDE Threat Analysis

Component	Spoofing	Tampering	Repudiation	Information Disclosure	DoS	Elevation of Privilege
Authentication API	High	Medium	Low	High	Low	Medium
Transaction API	Medium	High	Medium	Medium	High	Low

c) Evaluate Risks:

Having identified potential threats, the next step that has to be taken is to assess the risks that are inclined to the identified threats using the DREAD procedure. DREAD helps us assess each threat based on five criteria: measures such as Potential Damage, Reproducibility, Exploitability, Affected users, and Discoverability. This structured method helps in totaling the degree of risk and ranking them in terms of the risk factors. For example, a threat that can bring larger harm, within which reproduction is easy and the number of users threatened is large, would be ranked high. On the other hand, a threat that is difficult to imitate and impacts a few customers can be categorized as low priority. This evaluation process assists in directing risk mitigation efforts on the areas that have been deemed most significant hence the effective use of resources in guarding the system's weakest links.

Table 2: DREAD Risk Evaluation

Threat	Damage	Reproducibility	Exploitability	Affected Users	Discoverability	Risk Score
Data Breach	High	High	Medium	High	High	90
Service Disruption	Medium	Medium	High	High	Medium	75
Data Integrity Compromise	High	Low	Medium	Medium	High	65

d) Mitigate Threats:

Following the risk assessment and prioritization process, the best method of handling the aforementioned risks is determined, and techniques for dealing with the threats are put into practice. These mitigation strategies can be technical and non-technical. Some of the technical controls may be the use of secure modes of communication, the use of secure protocols for processing and storage of information, the use of multi-factor authentication protocols and real-time monitoring and logging of activities. The other preventative measures include coming up with strict organizational security polices and ensuring that all employees adhere to them, periodic providing of anti-virus training to the employees in order for them to appreciate security issues within an organization and ensuring that the organization is security conscious. Thus, the implementation of these technical and organizational measures provides a multi-level approach to the Fintech application's security that increases its overall safeguards against threats.

B. System Definition:

To classify the system in the framework of threat modeling and risk analysis for Fintech applications, we develop a detailed DFD. This diagram helps identify all the intertwining connections within the realms of the application and all the data requirements it encompasses, the interactions with banks and third parties as well as users. Hdfs give the overall view of the system because they show how data is processed and moves from one component of the system to the next. This way, we can determine points of the flow where data is processed or could be stored which are insecure and vulnerable. This level of mapping is very important in revealing some of the weak areas because it reveals areas that may need security controls. Considering the data flow is critical to the threat modeling and risk evaluation processes as it contributes to a comprehensive assessment of potential threats throughout the system and its components' connections. Analogous to that, the systematic methodology helps cover potential security issues and improve the security conditions of the Fintech application overall.

C. Threat Identification:

Thus, using the STRIDE scheme, it is possible to describe all the threats that influence the Fintech system based on the analysis of each item of the DFD. This method provides a general understanding of the system and reveals many threat directions to security.

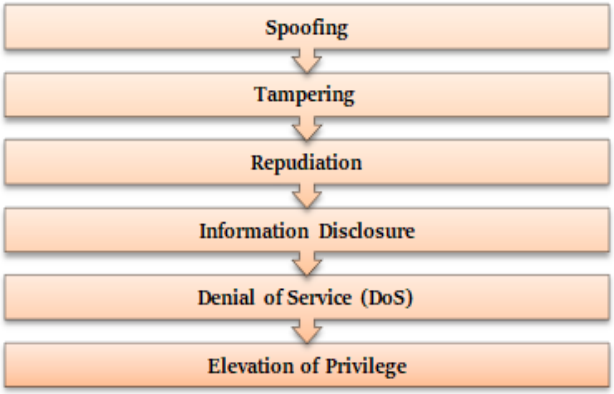

Figure 4: Threat Identification

a) Spoofing:

Spoofing threats are the type of threat in which the attacker finds himself within the system and pretends to be a legal user. This type of threat is most appropriate to authentication APIs; thus, it is possible to carry out a credentials theft as well as use any other trick to become a legitimate user. For example, an attacker may decide to steal and reinitialize all the session tokens with the aim of gaining access to the authoritative users' accounts. Such attacks quite often enter into unauthorized purchases, data theft, and many other unlawful activities. Concerning the spoofing threat, it is useful to adopt one or another

kind of user authentication means, such as multi-factor authentication, token management and others that would permit only authorized work with the given system.

b) Tampering:

Tampering threat, on the other hand, is a form of threat where the data is changed while it is in the system undergoing transmission or even when it is in storage. Therefore, if it was a Fintech application, the attacker would wish to alter the transaction data with the aim of committing fraud such as altering the number or the transfer amount or the recipient's accounts. These threats pose a great danger to the authenticity of a financial transaction, leading to great losses and the organization's image. To prevent these cases, it is required to keep the data and ensure that the attempts were made to modify it by others; this is where the usage of cryptographic algorithms such as digital signatures and checksums is required.

c) Repudiation:

The threats of repudiation involve actions that cannot be traced to the perpetrator of the action. Thus, users will be able to deny having performed the action in question. In the Fintech application, this could have been observed through a user declining a transaction that he or she had initiated, leading to conflict and fake compensation claims, among others. Trust is impacted because of repudiation, and the dependability of the system is impacted. In that regard, proper logging/audit trails that record all the activities performed by the users were to be implemented with adequate depth. Other services, which include digital signatures and non-repudiation services, may also guarantee that an act performed can be proved to have been completed as well as authenticated, which means that the distinct cannot be allowed any freedoms to disclaim that the action was carried out by him/her.

d) Information Disclosure:

Information disclosure threats pertain to the exposure of information to personnel who do not exercise discretion. This could be in the form of insecure API through which the attackers get contrary permitted access to information such as user login credentials, clients' information, as well as financial account information, among others. The responsibility to provide information can result in the loss of data, theft, and high fines or other punitive measures. Against these threats, it is important to apply sockets layer/transportation layer security during data transfer and storage, administering the application through secure hashing and carrying out the security audit in order to detect and fix problems affecting API interfaces.

e) Denial of Service (DoS):

Denial of Service (DoS) threats usually involve a flood of requests to a system that will make the availability of a service stall. In a Fintech application, such attacks can deny essential services and put the actual users on the other side out of their accounts or away from their transactions. DoS attacks, generally, can cause operation disruption and financial losses. Staples that have to be taken in order to provide adequate protection against DoS threats involve rate-limiting traffic filtering, among other matters that line up strong infrastructures to effectively handle attacks that are likely to be large in nature. Further, prevention measures or actions specifying ways and means of tracking from the time it is initiated to the time when it is executed should be there to monitor and handle the DoS attacks.

f) Elevation of Privilege:

The threats under this category entail the violators trying to gain access to another level of the system that they are not supposed to. An attacker may take advantage of the above vulnerabilities by attaining higher-level access to the restricted data and functions that he/she should not access. For instance, while an attacker may have restricted control, such as the user level, they can use the vulnerability to assume control of the administrator's level and change all sorts of data and system characteristics. As it pertains to the risk of privilege escalation, the guideline of letting subordinates have only the minimum access level much less than the level of their superiors is vital. Updating the system regularly and applying patches that have been developed to deal with the defined threat can also be useful in managing them since they are known quantities.

D. Risk Evaluation:

Analyzing the Fintech application with the help of the DREAD methodology, the risks linked to each of the above-mentioned threats are defined in detail. Each threat is assessed based on five key criteria: That is the harm that may be incurred by a project, reproduction capability, exploitable commodity, users that may be impacted by a project, as well as the discoverability of a project. The MBA style is useful for finalizing the kind of analysis structure ranking threats in regard to their risk and probability and participating in the process of effective strategy creation.

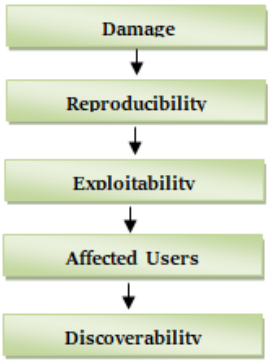

Figure 5: Risk Evaluation

a) Damage:

The Damage criterion assesses the readiness of an organization to face a threat in case the threat is to be implemented. This includes the outcomes that are financial and, operational and legal effects of the security breach. For example, suppose the attacker can spoof authentication APIs. In that case, this may lead to ability for an unauthorized attacker to make transactions, and this may lead to numerous losses and customer trust being lost. Similarly, the intervention of transaction data disrupts the following aspects of the system: fraud and regulatory fines among them. It helps in avoiding threats that entail more danger to inflict wide loss on an organization; this is because after assessing harm, one is aware of the areas that will be most affected by a particular threat.

b) Reproducibility:

Reproducibility is a subscale that depicts the capability of reproducing the threat by an attacker. In the same way, threats that can be easily reproduced are of higher risk since the attackers can try pushing the threats until they are successful. For instance, if there is vulnerability in an API it is easy for the attacker to launch an attack on it without hardly any complex methods of hacking. By evaluating the degree of reliability, one is able to determine threats that require attention at the soonest possible time, in light of the fact that these threats are easily implemented, subsequently lessening the chances of such attacks being carried out.

c) Exploitability:

The concept that measures the likelihood of manufacturing the exploitation of a threat as vulnerability is called Exploitability. This entails the determination of the magnitude of the challenge that the released threat demands in terms of the attacker's resources. For instance, if a vulnerability is exploitable with the help of tools available and little understanding of the former, then the factor is high exploitable. On the other hand, threats that need specialized skills and a lot of capital are not very feasible. Knowing exploitability is useful in handling threats that diverse attackers can exploit since it assists the ranking of easily reachable threats on attackers to apply gained information on the mode of handling threats based on their exploitability aspect.

d) Affected Users:

The Affected Users criterion focuses on the number of people who would be affected in case of the threat materialization. In addition, threat types that may impact many users simultaneously, such as DoS attacks, are especially malignant owing to their ability to affect the services of numerous clients. Conversely, threats that are likely to affect only some of the users are likely to be considered of lower priority. Through assessing the coverage of users, it is possible to identify major threats to user-base as such that shall affect most of them in case of occurrence and guarantee that the risk mitigation measures being taken are aimed at providing the majority of the users.

e) Discoverability:

Discoverability provides a measure of the chance that attackers will detect a specific threat. Those threats, which can be ascertained either by using organizational scanners or engaging in low-level espionage, are posited to be highly discernable. For instance, APIs that are exposed and have little security can easily be attacked because their paths can be easily revealed. As for the threats which involve detailed knowledge about the system and complex methods, one cannot easily deal with them. Whereas discoverability of the threats assists in determining which threats are most probable to be exploited, and this, in turn, can help in protecting the most conspicuous and frailty elements of the system.

E. Threat Mitigation:

In order to counter the threats outlined above, it is necessary to use technical and organizational safeguards along with the use of technical controls. The above-stated strategies aim at improving the security of the Fintech application in order to instill a strong form of defense against any possible attackers.

a) Technical Controls:

Technical controls are important in managing security threats as they enhance the application of security measures in the Fintech application. One of them is the use of encryption- to secure data communicated both internally and externally as well as the data stored. To this end, the system guarantees the protection of highly sensitive information, including user credentials and transactional funds, among others, by using strong encryption algorithms in case information is intercepted by the wrong hands. Moreover, the use of Multi-Factor Authentication (MFA) proves to be much more effective as it limits access to the materials with stricter settings, as after entering the password, the user is asked to provide several proofs of identity. This means that it becomes difficult for the attackers to penetrate the accounts even if they have the relevant login details. Moreover, proper configuration of monitoring and logging services is crucial for the timely identification of suspicious activities. The former of these systems can recognize novel behaviors, which might signal an ongoing attack, and then quickly act to reduce the consequences thereof. Technical controls are put in place to make the Fintech application more secure against different attacks so as to minimize cases of hacking.

b) Organizational Measures:

Although technical controls are crucial, they are reinforced by the organizational measures which are of equal significance in combating security threats. The most basic requirement is to set up security policies, which form the framework of security in an organization. It must include all forms of security, ranging from data and access control to security incidents and compliance with legal measures. In addition, security training and awareness to employees and or executive team shall also be carried out periodically. These programs tend to inform the staff on various new emerging threats, on methods of protecting information and on different practices to observe when it comes to security. In this way, security awareness and the further education of the company's employees minimize the likelihood of a human factor, which, as a rule, plays a crucial role in such events. Further, the incorporation of security measures into the organizational culture allows for the consideration of security factors in all the activities and programs of the organization. This cultural change assists in developing a security-awareness culture through which everyone in the organization is knowledgeable about the responsibility of securing the system and the information. Organizational measures assist in the continuous management of security problems by incorporating a broader approach than technical measures alone while considering the people aspect of the employees involved.

F. Risk Assessment Process:

Figure 6: Risk Assessment Process

a) Asset Identification:

The first level of the risk assessment model that has been conducted on the Fintech application is the assessment of the assets, whereby we determine the different assets at the application's disposal and group them according to the level of risk involved. [21] These assets can be broadly classified into three categories: information resources, application resources, and infrastructure resources. This may refer to information such as customer financial details, transaction records, identity information, as well as any other information that, if disclosed can cause a lot of financial or reputational losses. Technical application assets include all the APIs, microservices, and other software components that enable the Fintech application. Infrastructure assets refer to tangible and intangible resources in the form of hardware and software that support the running of

infrastructure for the application. Every asset is checked case by case for its worth, its importance to a system, and its failure. For instance, customer financial data is a type of data that is considered more sensitive and would, therefore need better security measures compared to internal systems' logs that may be equally significant. It is useful to divide them in order to determine which of them require the strongest protection measures.

b) Threat Identification:

Once the structure of the assets has been defined, the primary step is to define suspicious or potential threats towards these assets with the support of threat intelligence information and vulnerability databases. Regarding this, it entails the identification of potential threats and categorization of the relevant assets based on the nature of threats that may confront them. For example, in the case of a telecommunications company, it is possible that customers' confidential information might be of interest to hackers with the aim of perpetrating acts of identity theft or fraud. APIs may be attacked by the usage of such flaws as a lack of input validation to conduct injection attacks. Infrastructure assets may often be at risk of a Cyber threat such as Distributed Denial of Service (DDoS) or even an attempt to tamper with physical resources. Thus, using threat and vulnerability databases containing the latest data in the field helps to monitor the appearance of new threats and study how possible vulnerabilities may be used. Considering the assets' susceptibility and isolation to external threats, the level of security, and the data it processes, each of them is assessed.

c) Vulnerability Analysis:

After threats have been put down, we move straight to vulnerability analysis, in which we analyze the susceptibility of each asset to each of the threats. This means assessing vulnerability, which can be described as the flaws or holes within the system that the attackers can take advantage of. For instance, if an API is without good input validation mechanisms, it might be exposed to SQL injection occasions where hackers can use input vectors to run nasty SQL queries that endanger the database of the API. Likewise, if private data is carried, then the message could be accessed when being transferred and then read by undesirable individuals or corporations. In this step, various tests are performed for security and aspects such as code reviews to expose flaws. This includes penetration testing, where an attack is simulated with a view of exposing the loopholes that actual attackers are likely to use to gain entry into a network. For that, you have to get a basic understanding of the security state of each asset and determine where there are weaknesses.

d) Risk Evaluation:

After vulnerabilities are determined, risk assessment is the next step, which entails the identification of the risks involved in a particular threat, capitalizing on the said vulnerability. This is done through a structure known as a risk matrix, which, of course, aims at the graphical representation of the risks collected in a manner that can be easily understandable especially with regard to their probability of occurrence and their impact. The risk matrix considers two main factors: the consequences if the threats occur (e.g., expected monetary loss, negative impact on the organization's reputation and liability) and the probability of risk emergence (e. g., attack frequency, vulnerabilities). All the threats are mapped on the matrix; this then enables us to classify them depending on their risk level, which can be high, medium, or low. Essential threats are more hazardous risks, capable of inflicting high losses, are probable to occur, and are essentially viewed as the highest priority. Medium and low-risk threats though important, are managed taking into consideration the resources available or their effectiveness on the system. The structure of this approach also makes it possible to prevent the most serious risks, thereby improving the security of the Fintech application.

e) Control Implementation:

The last process in the risk assessment process is control implementation which involves putting measures for managing risks that have been identified. These controls can be grouped under technical control and procedural control. Examples of technical controls are specifying user access in order to prevent intruders from gaining access to confidential information, utilizing encryption to safeguard data in transit or storage, and using IDS to facilitate the identification of malicious attempts at penetration. Security updates and patches are also necessary on a recurring basis as well to resolve some of the open security risks and to strengthen the system's security even further. Organizational measures entail putting in place and continuing support of measures for the handling of security incidents. There are security awareness programs that are understood to increase the awareness of the workers or employees, as they are informed on how to go about matters that relate to security, being informed on the latest security threats and measures that should be taken to avoid security threats. A periodical security assessment is carried out, hence; checking compliance to the industrial standards and regulatory measures, as well as checking up on the effectiveness of the measures put in place. Thus, the advanced set of technical and organizational measures can be

stated as the structure of an integrated security system which would cover technological and personnel security risks and create a solid defense against possible threats.

IV. RESULTS AND DISCUSSION

A. Threat Modeling Results:

Our threat modeling process yielded the identification of several critical threats to the Fintech application, each necessitating specific mitigation strategies:

a)SpoofingAttacks:

The API threat that occurred at some point in time was the employment of the facilitated account with the aim of gaining access to the API. Some of the possible threatening explanations can be misrepresentation of users, through services like phishing as well as identification methods that are very substandard. Thus, we introduced MFA; in the situation where the password was compromised the intruder will be locked out by the next layer of security.

b) Tampering:

The issue regarding freely accessing and altering the transaction data was still a threat to the non-ethical personnel to the accuracy of the entity's data and the system. To curb this, we had to encrypt the data when transferring it from one end to the other and also when storing it and had to include the use of digital signatures to verify that the data being received in the transaction was indeed from the right source and had not been tampered with in any way.

c) Information Disclosure:

The fact regarding the observation of insecurity breach concerning one's sensitive information due to insecure endpoints was mentioned. To counter this, we enhanced the coding standards, secured all the endpoints and transmitted data over HTTPS like TLS.

d) Denial of Service (DoS):

Hence, the service, which is rather often exposed to excessive API requests, could be interrupted, and this crowned the need for the correct implementation of rate limiting and traffic monitoring features. There is also rate limiting to limit the number of requests for a particular time, besides IDS, with the aim of analyzing the frequent API call traffic.

B. Risk Assessment Results:

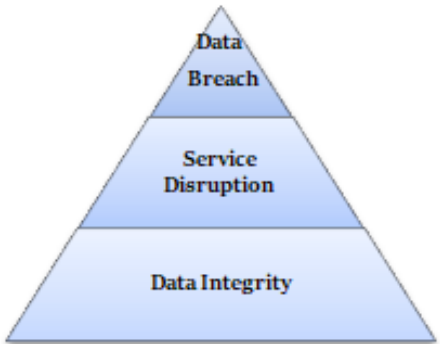

Figure 7: Risk Assessment Results

a) Data Breach:

During the course of identifying threats in this case, the threat of unauthorized access to customer's financial data came out as the most serious threat. Such a data breach could lead to the loss of a signification amount of money, discredit the organization in the public domain, and attract legal consequences. In order to manage this critical risk, the following measures were put in place. As a result of continuous/consistent monitoring of this critical risk the following measures were achieved. Data compartmentalization was also used in order to separate the core parts of the system from crucial data, which, in case of attack, would minimize threat effects. Thus, by segmenting the network, we are able to establish that even if the attacker gets past some level in the network, they cannot easily get to the other levels that are more sensitive. Also, to ensure the data is secured while stored and when in transit, we applied encryption. Encryption makes it hard for the data to be accessed or intercepted by an unauthorized individual because, in case it is accessed, the data is in an encrypted form hence challenging to decode. Form another point of view, there were implemented restrictions on using certain data, which would allow only users with particular

roles and responsibilities to read or change it. These measures cascaded ionically strengthen our defense against data breaches; in effect, customer financial data becomes save-guarded against exploitation by unauthorized persons.

b) Service Disruption:

Another high risk identified was the disruption of service by the hackers through a denial of service (DoS) attack. Such attacks can potentially hinder the operations of a Fintech application that deals with the constant processing of transactions and service availability, all of which cause customer dissatisfaction and quantifiable losses. To this effect, we use intrusion detection and prevention systems which continuously scan through the network traffic and prevent any event that could be associated with a DoS attack. Such systems involve complex programs that analyze for any form of deviation from normalcy and take corrective measures to ensure that the threat does not impact the service delivery. Furthermore, we also configured backup and the presence of backup in case of an attack so that the system is always up and running. Thus, with systems and other ways of processing the information on standby and available as redundancy, service availability and easy processor failure recovery become possible. These steps are important for providing the high availability and service reliability of the transactions processed without interruption due to attempts at the corresponding attacks.

c) Data Integrity:

The impact was noted in the areas of lost transaction data integrity through tampering, which was considered a compromise. This can again result in wrong records of financial transactions, bogus transactions taking place, and, above all, eroding the credibility of the system. In order to minimize this risk, the following cryptographic techniques were applied to the integrity of data: The hashes are the digital fingerprints of data that enable the identification of their original state and any changes that have occurred without the owner's consent. Digital signatures are useful as another form of security measure where it is possible for us to validate both the source of the data and also confirm that the data has not been tampered with since it was signed. There are also security reviews and scans carried out frequently to monitor the security status of data and to check for any signs of data tampering. When these cryptographic measures are coupled with frequent audits, it is possible to keep the accuracy and security of the transaction data steady in the long term, thus preserving trust in financial statements.

Table 3: Risk Assessment Results

Risk Type	Description	Mitigation Controls
Data Breach	Unauthorized access to customer financial data	Network segmentation, data encryption, regular audits
Service Disruption	Inability to process transactions due to DoS attacks	Intrusion detection systems, rate limiting, redundancy
Data Integrity	Compromise of transaction data integrity through tampering	Data integrity checks, secure coding practices

C. Case Studies:

a) Capital One Data Breach:

Another essential and well-known example of the adverse result of open API insecurity is the Capital One data breach that took place in 2019. The principal weakness was a misconfigured web application firewall, and through it, the attacker got unauthorized access to data belonging to over 100 million clients. The attacker leveraged this misconfiguration to get the credentials that he or she used to get data stored in an AWS S3 bucket. Besides, this breach involved a huge number of people's personal information, which in most cases includes social security numbers, credit scores, and bank account information. Recent attacks also emphasized the necessity of conducting a rigorous security assessment and frequent configuration audits. What is most critical is maintaining proper settings configurations across the board – anything that could be a weak link needs to be severed. This is why the aggressive Capital One breach underlines the need for effective security measures that encompass testing and verification of the configurations, constant monitoring of the system for potential threats, and timely response to resolve the problems endangering the safety of the clients' data.

b) Plaid API Vulnerability:

Therefore, the lack of security of the Welsh political group's Plaid API in 2020 is another critical real-world example of the consequences of API flaws. This vulnerability is an example of a type of issue quite frequently encountered in programming and development; the issue pertains to input validation, and it can be as consequential or malignant, depending on the social relevance of the corresponding APIs. On the same vulnerability, attackers were able to post malicious code into the input field

and acquire restricted access to at least sixty million users' financial information. Thus, it was evident that sensitive data like bank account details and transaction history leakage leads to the aspect of secure coding in API design. The case described here perfectly illustrates why there ought to be a clear strategy on how the input should be validated, and the web servers made more secure to prevent such input from infiltrating the system. Other security bugs should also be scanned and checked in the code reviews before any attacker has a chance to take advantage of the flaws. In addition, the general habits and the adherence to some of the coding frameworks and best practices reduce the probability of having these flaws. The observation that can be drawn from the Plaid API vulnerability is that there should be steady progress made towards safeguarding APIs as well as ongoing feedback from threats and the recent available data on preventive measures to shield customers' data.

Table 4: Case Studies

Incident	Description	Lessons Learned
Capital One Breach	Misconfigured web application firewall led to data exposure	Regular configuration audits, secure API settings
Plaid API Vulnerability	Improper input validation led to data exposure	Secure coding practices, thorough testing

D. Discussion:

Thus, the obtained findings could not be more indicative when it comes to stress on the need to implement increased protective characteristics of APIs within Fintech solutions. As information security is the key factor to the company's success, and given the rank of the financial data, together with the increased level of cyber threats, the systematic approach to identify potential security risks and their mitigation is necessary. Thus, the threat of security violations, which may potentially endanger the customers' personal data or affect the operations of the financial companies, can be mitigated. Nevertheless, the perfection of API security solutions is not an activity that is done independently because it needs a coordinated effort. The measures that analyze the protection against unlawful access and data modification include encryption, Multi-Factor Authentication (MFA), and rate limiting. The first difference is encryption, which means that the content of the data should be safe before the transmission while it is in the process of transmission and right after it; the second is MFA because the user has to identify himself/herself with more than one factor. Several controls can be achieved with rate limiting including stopping abuse since the API requests are limited and hence no denial of service attacks. In addition to such technical actions, organizational measures are also important. The security policy should be created with high-level security measures to be achieved for the constant awareness of the employees, together with proper response measures when an incident happens. Also, periodic security audits are necessary to ensure that the existing safeguards are still effective, especially over time, and to confirm that the most recent threats cannot outsmart the applied security measures. That is how a more global approach, inside which an organization can develop itself, becoming prepared for countering contemporary threats and opportunities, can be effective for construction.

E. Future Work:

As for further research, predictive approaches should be identified to solve the problem quickly without incompliance with the security assessment process. The problem with manual assessments is that they are very detailed and thus take a lot of time and other resources. The use of these automated tools can increase efficiency because the threats and vulnerability assessment can be swiftly conducted. These tools can solve intricate and time-consuming problems, such as using preset parameters and analyzing numerous possible aspects by applying machine learning methods that can be missed during a manual examination. Thus, more research papers are required to discuss how the Fintech value chain is affected by emerging technologies and API security. Thus, some technologies, such as block chain, can be highly beneficial for improving data reliability and openness. Blockchain, owing to its structure, can also help in making the records of transactions and the transactions themselves much more credible thus making it more secure. Likewise, artificial intelligence (AI) can also be used in the improvement of threat identification and crisis management. AI patterns can analyze large volumes of data to recognize obvious patterns of security breaches to occur before they happen. This paper examines these technologies with the aim of obtaining fresh ideas and techniques that could be used in enhancing API protection in Fintech apps. Such research can lead to ideas for protecting the systems from cyber threats to maintain the integrity of the sensitive data of finance.

a) Automated Threat Modeling Tools:

This is why the appearance of automated threat modeling tools can be considered a great step forward in the field of cybersecurity, especially when it comes to API protection in Fintech. These tools use highly technical methods that include formulaic and machine learning to analyze system structures and discover risks and weaknesses. Sophisticated tools used in

threat modeling help in cutting down significantly the time needed for effective risk assessment. Cuts out the check and routine work, which saves a lot of time for security professionals to focus on more complex analysis and planning. Moreover, the use of automated tools leads to more precise threat evaluations owing to constant system surveillance of the newest threat descriptions. This continuing assessment guarantees that threat models are updated with the current security intelligence and the current systems being used, thus enhancing the overall threat counters measures. Introducing automated threat modeling tools as part of the security structure increases the organization's capability to remain a step ahead of new evolving threats.

b) Blockchain Technology:

AI Blockchain also makes a new approach to changing API security by using the features of decentralization and non-tampering. In a system with the help of this technology, all exchange of values or, in simpler terms, all transactions along with the overall communication between the users is completed and recorded on what is known as the ledger, which is extremely difficult to counterfeit. It is for this decentralization that no party can own full control of the entire dataset thus eradicating the instances when data might be hacked or fake. Referring to the concept of the blockchain it creates an unchangeable log of all the API activities, which might be useful sometimes for data security. Besides, the use of smart contracts as the type of contracts that allow users to define the conditions in code and the application of this code by the API will contribute to API security as well. Smart contracts also assist in the effectiveness of the transaction process, free from errors from individuals and fraud. This way, Fintech applications can be made more secure by adding on the benefits of blockchain technology to make all the transactions and data exchange more secure and absolutely transparent.

c) Artificial Intelligence:

Artificial Intelligence (AI) and Machine Learning (ML) are among the critical attributes of modern cybersecurity since they potentially contribute to the enhancement of threat detection and response. After the training phase, the algorithms can take into account the impacts in terms of the standard or variations that indicate a threat; this process is done very fast in order to scan a lot of data in a short amount of time. They also help to see risks when they are in their infancy, and as such, it becomes easier to gain control of risks to prevent their occurrence. Stating behaviors in the application of AI can monitor both the activity of the user and the system to identify their mainstream activities and a race towards a malicious activity. For instance, several times a day, one can notice certain specific and suspicious login activity or such transactions which deviate from the average. Another noteworthy application of AI is risk analysis and assessment, which analyzes the existing and historical data using statistics and other tools and estimates the likelihood before the risk materializes. Thus, AI helps to deepen the analysis of previous implications of accidents and new trends in the process of creating, preventing and enhancing enterprises' security. Thus, there is an effective use of Artificial Intelligence in managing API touchpoints and protection, and an enhancement of the dynamic control against cyber threats.

Table 5: Proposed Future Research Areas

Research Area	Description
Automated Threat Modeling Tools	Development of tools to automate the threat modeling process
Blockchain Technology	Exploration of blockchain for enhancing API security
Artificial Intelligence	Use of AI for detecting and mitigating security threats

V. CONCLUSION

First of all, it should be mentioned that APIs play a critical role in all Fintech applications as they help to manage interactions and data sharing processes between various systems and services. This permits the financial institutions, payment processors as well as other participants to offer seamlessly integrated services that add value to the end user and optimizes the underlying operations. However, APIs being open and available to the public are a serious concern for security threats that can occur. These vulnerabilities can cause serious impacts such as data leakage, loss of money, and tarnishing of the organization's reputation. Thus, this paper has discussed the elements of the threat modeling and risk assessment process in the context of the usage of APIs in Fintech applications to expose the extent of the security risks and the challenge of their protection.

The framework described in this paper starts with a critical examination of the threats and their significance, which comes out during threat modeling. Through the use of methods like the STRIDE and DREAD, one is able to assess risks relating to different threats, including spoofing, tampering and DoS. The detailed threat identification and risk evaluation make it possible to solve the most critical security issues. Thus, for protecting important financial data and the integrity and

confidentiality of transactions, the effecting implementation of mitigation measures by encryption, multi-factor authentication, rate limiting, etc, has to be carried out properly.

Thirdly, the paper also rushes to examine the implication of integrating technical control with organizational control in order to achieve the aim of protecting API. Encryption of data and the use of intrusion detection systems form the base of the information technical controls that lock out unauthorized users and discourage negative conduct. Another crucial facet is the policies and procedures, which are as vital as the technology infrastructures; these are having good security policies, conducting awareness programs for employees, and having good incident handling processes. Therefore, it states that security has to be monitored frequently and that the correct procedure is to carry out security check-ups periodically.

The real-life examples of the Capital One data breach and the Plaid API vulnerability also highlight how careful one must be when it comes to security. These scenarios support the author's approach to identifying weaknesses in API configurations and coding best practices, as they show that numerous breaches stem from such issues. Thus, when Fintech applications' tasks involve similar operations, they can learn from these incidents and apply measures for their prevention as recommended by the research.

Summarizing, the proper utilization of threat modeling and risk assessment of Fintech applications is critical to increasing the security profile of the developed solutions and protecting consumers' sensitive financial data. As a result, the conceptual model suggested in this paper provides a set of guidelines for the assessment of potential threats stemming from API usage and the means to address them to enhance Fintech systems' protection levels on the whole. The increase in Fintech services, continual innovations in technology, and new regulations will require more research and development in security practices to address the emergent threats that may threaten the applications. Besides, strengthening API security plays a significant role in avoiding security incidents and enhancing the users' confidence in Fintech services, which, in turn, contribute to the permanent development of the industry.

VI. REFERENCES

[1] Jha, A., & Sharma, R. A risk assessment of fintech adoption in the Indian financial services industry. Developments and trends in the banking and finance sector, 66.

[2] The Importance of API Security for Protecting Financial Cloud Apps, fintechweekly, online. https://www.fintechweekly.com/magazine/articles/the-importance-of-api-security-for-protecting-financial-cloud-apps

[3] Fintech API: Everything You Need to Know, yellow systems, online. https://yellow.systems/blog/fintech-apis

[4] Chakraborty, S. (2018). Fintech: Evolution or Revolution, Business analytics research lab India.

[5] OWASP. (2020). OWASP API Security Top 10. Online. https://owasp.org/www-project-api-security/

[6] Shostack, A. (2014). Threat modeling: Designing for security. John Wiley & Sons.

[7] Microsoft. (2005). The STRIDE Threat Model. Online. https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)

[8] National Institute of Standards and Technology (NIST). (2012). NIST SP 800-30: Guide for Conducting Risk Assessments. Online. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf

[9] International Organization for Standardization (ISO). (2022). ISO/IEC 27005: Information Security Risk Management. Online. https://www.iso.org/standard/80585.html

[10] Capital One. (2019). Cyber Incident. Online. https://www.capitalone.com/facts2019/

[11] What Is Threat Modeling?, cisco, online. https://www.cisco.com/c/en/us/products/security/what-is-threat-modeling.html

[12] Protecting FinTech APIs, Salt, online. https://content.salt.security/rs/352-UXR-417/images/SaltSecurity-SolutionBrief-ProtectingFinTechAPIs.pdf

[13] Threat Modeling, fortinet, online. https://www.fortinet.com/resources/cyberglossary/threat-modeling

[14] Xiong, W., & Lagerström, R. (2019). Threat modeling–A systematic literature review. Computers & Security, 84, 53-69.

[15] Shevchenko, N., Chick, T. A., O'Riordan, P., Scanlon, T. P., & Woody, C. (2018). Threat modeling: a summary of available methods. Software Engineering Institute| Carnegie Mellon University.

[16] Saini, V., Duan, Q., & Paruchuri, V. (2008). Threat modeling using attack trees. Journal of Computing Sciences in Colleges, 23(4), 124-131.

[17] Threat Modeling Methodology: STRIDE, iriusrisk, online. https://www.iriusrisk.com/resources-blog/threat-modeling-methodology-stride

[18] DREAD Threat Modeling: An Introduction to Qualitative Risk Analysisk, ECcouncil cybersecurity-Exchange, online. https://www.eccouncil.org/cybersecurity-exchange/threat-intelligence/dread-threat-modeling-intro/

[19] Threat Identification, warditsecurity, online. https://warditsecurity.com/threat-identification/

[20] Fintech risk management: Strategies for success, BPM, online. https://www.bpm.com/insights/fintech-risk-management/

[21] Risk Management And Financial Technology: Strategies for Success, dashdevs, online. https://dashdevs.com/blog/risk-management-in-fintech-strategies-for-success-dashdevs/

[22] Matt High, Threat identification: cybersecurity and user experience, 2020. Online. https://fintechmagazine.com/venture-capital/threat-identification-cybersecurity-and-user-experience