API Security Risks and Resilience in Financial Institutions

Arpita Basak

Divya Tiwari

2025 Laurea

Laurea University of Applied Sciences

API Security Risks and Resilience in Financial Institutions

Arpita Basak; Divya Tiwari

Future-Oriented Project Management

Thesis

April, 2025

Laurea University of Applied Sciences **Abstract**

Degree Programme in Future

Oriented Project Management

Master of Business Administration

Arpita Basak; Divya Tiwari

Resilience in Cyber Security

Year 2025 Number of pages 68

Through the analysis of Application Programming Interface (API) vulnerabilities and the identification of critical risk variables that Impact API security, this study aims to increase cybersecurity resilience in financial institutions. The risk environment becomes more complicated as financial services depend more and more on APIs for tasks like open banking, payment processing and real-time data. Interchange. This study tackles these issues by identifying, evaluating and migrating vulnerabilities using a data-driven methodology. To assist security teams, organizations and regulators in bolstering their security and incident response plans, a methodological approach is suggested. The study uses R programming for data preprocessing, statistical analysis and visualization of vulnerabilities associated to APIs, utilizing publicly available data from the National Vulnerability Database (NVD). To investigate trends in severity, attack vectors, necessary privileges, and CWE (Common Weakness Enumeration) classifications, important analytical tools like trend analysis, Kruskal- Wallis tests, and Dunn's post-hoc comparisons are employed. According to the findings, there has been a discernible increase in high- and critical-severity vulnerabilities over time. Common threats include improperly set access controls, broken authentication, excessive data exposure, and injection attacks. The most common attack vector turned out to be network-based attacks, highlighting the necessity of strong authentication procedures and safe gateway configurations. The report suggests a cybersecurity resilience paradigm specifically designed for financial institutions based on these insights. It consists of elements like automated threat intelligence integration, risk-based access control, real-time monitoring, and best practices across the API Software Development Lifecycle (SDLC). Additionally, the platform encourages constant anomaly detection, frequent penetration testing, and secure code. Financial institutions can better protect sensitive financial data, adhere to laws like PSD2 and GDPR, and lessen their vulnerability to API-related dangers by using this strategy. The report also identifies areas for further research, such as practitioner- driven evaluations and AI-driven anomaly identification.

Keywords: API security, Cybersecurity Resilience, Financial Institutions

Contents

1. Introduction

Cybersecurity is now an important area in organizations in this current day's digitized world. IT systems and particularly financial institutions have been identified as being most at risk of an attack due to their use of digital technology for their work and the sensitivity of the data with which they work. Consequently, attacks on these systems can result in grand thefts, service dislocations, and eroded public confidence. Therefore, to manage these challenges, the idea of cybersecurity resilience has been developed as a proper approach to threats.

Cybersecurity resilience means an organization's capability to prepare for, maintain operating capability in the face of, and restore from cyber threats and cyber incidents. Unlike other frameworks it extends the scope of security to also include the continuous operation in spite of known attacks. For the financial industry, this means not only protection of the information but also the availability of vital services. Nevertheless, many organizations still fail to establish proper resilience measures, partially because these organizations do not receive enough practical advice based on research results. As such, this thesis aims to fill this gap by designing a model of cybersecurity resilience for financial institutions. The work is based on data that can be accessed online which includes the feed from the National Vulnerability Database (NVD). In the present study, relatively quantitative analysis of the data is carried out for the purpose to make practical suggestions related to the enhancement of the resilience programmes of the given organisations. Besides, this thesis seeks to study the uncertainties of dealing with vulnerable data especially APIs which are frequently attacked. Besides, this thesis seeks to study the uncertainties of dealing with vulnerable data especially APIs which are frequently attacked. In this thesis, the primary goal is expanding the insight into cybersecurity resilience and offering empirically grounded best practices for developing it. The paper's implications will be of interest to multiple organizations, not just financial institutions, and will also help extend work on organizational resilience to cyberizing attacks.

1.1. Background of the Study

With the virtual world becoming an increasingly integrated part of people's lives, the financial sector is, therefore considered to be in the crosshairs of cyber threats. This paper examines the threat that has emerged from innovative and advanced cyber risks to financial stability and security. As pointed out by the IMF, the financial sector has been a victim of nearly a fifth of reported cyberattacks for the last two decades, leading to about $12bn of direct losses (Böhme and Schwartz, n.d.). COVID has deepened digitalisation efforts in the financial services industry, thereby amplifying the attack vectors for unsavoury parties. According to the BIS, the financial sector endured the second largest proportion of COVID-19 related cyber- attacks, indicating the sector's weakness during crises (Ebsco.com, 2025).

Due to such emerging and enhancing threats, the authorities or the regulatory framework bodies have examined and advanced the cybersecurity perspective. Currently, the Federal Deposit Insurance Corporation is working on the reinforcement of cybersecurity in connection

with the financial sector and mainly focusing on the issue of risk management programs (Domingues, 2018). However, the following problems have still existed.

The financial industry in particular was shown to be generally improving in threat prevention by a 2024 report from Picus Security, however there are still critical weaknesses in detection suggesting that the financial sector still has work to do in the cybersecurity domain (Labs, 2024). The issue of cybersecurity is relatively recent, but currently, financial institutions cannot ignore the question of cybersecurity resilience. It includes the following; prevention, protection, detection, reporting and ability to respond, recuperate and learn to maintain the organization's business operations going even in the face of cyber operations. Hence, establishing strong and sound resilience strategies are crucial required for managing risks and improving public confidence on the financial system. The purpose of this work is to make a particular contribution to the existing literature on the improvement of the cybersecurity situation in the financial industry. Using data feeds from the National Vulnerability Database (NVD) and quantitative methods of study, this research aims to construct a model suitable to the problems of financial institutions.

1.2. Purpose and Aim of the thesis

The main goal of this thesis is to enhance cybersecurity defense in financial institutions with the help of analysing the open data sources to reveal key weaknesses and prepare suitable measures. Cybersecurity resilience on the other hand can be described as the improvement and preparedness of an organisation in respect of cyber security risks that threaten the ability of an organisation to deliver some of its key services. This thesis aims to provide solutions that financial organizations can leverage in meeting challenges of data security, operational continuity, and preserving the public's confidence. The purpose of this study is to investigate the patterns and trends of cybersecurity threats from the available data set which is NVDD. As a quantitative-deductive study, the research will determine the antecedents to resilience of financial institutions. In addition, the challenges of dealing with sensitive data as well as those characteristics of Application Programming Interfaces (APIs) concerning the execution of financial technologies and services, which are constantly under cyber threats, will be studied. The research also seeks to fill the identified theoretical-practical gap by proposing an applied resilience framework for the financial sector. This framework will deliver valuable intelligence and practical guidelines for strengthening the cybersecurity front line and helping organizations be better prepared for shifting threats. Finally, the results of this thesis will not only be valuable for financial institutions but also for enhancing the cybersecurity resilience concept's understanding and applying it elsewhere, where similar risks are present.

1.3. Research Questions

To fulfill the purpose and aim of this thesis, the research targets questions that relate to real challenges enhancing cybersecurity resilience in the financial industry. These questions are used as a framework in the study in order to make the analysis meaningful in achieving its objective. The research questions are as follows:

What factors most influence cybersecurity resilience?

This question is designed to show what aspects are most important in an organisation's capability to prepare for, cope with, and reset from and after a cyber incident. In undertaking a pattern analysis of the publicly available data, the study will determine the causes of resilience in the financial sector.

How can public data strengthen organizational resilience?

This question concerns itself with how basic data which is freely accessible, like the National Vulnerability Database (NVD), can be used to improve the cybersecurity mechanisms. It discusses aspects of how decision-makers can protect their organization's vulnerabilities and enhance the general state of defense.

These research questions help in articulating the study and are in accordance with its objectives. The answers to these questions will be used as a basis for the creation of a theoretical and practical guide for the construction of a cybersecurity resilience approach for the financial sector.

1.4. Scope and Delimitiations

The subject area of this thesis is dedicated to the possible options to improve the cybersecurity defense of the financial sector. This includes reviewing of datasets obtainable in the public domain to look for vulnerable indicators of resilience with a view of coming up with measures can be taken by financial institutions to safeguard sensitive data, and maintain operational functionality. Quantitative data is the focus of the study due to the availability of feeds in the National Vulnerability Database. These feeds give rich information about the present-day opportunities in cybersecurity and its threats; a quantitative treatment for these feeds would give one a pattern of the threats.

While the thesis aims to address critical aspects of Cybersecurity resilience, it has several limitations

Industry Focus: This research focuses on the financial industry in particular, because it is heavily reliant on digital systems and because cyberattacks are apt to happen to it. This is true considering that the research might not have afforded a perfect insight into other industries that have dissimilar operation and security vulnerabilities.

Data Sources: No proprietary data is used in the analysis, apart from the NVD feed data that is publicly available. Information contained in data samples cannot be owned by certain

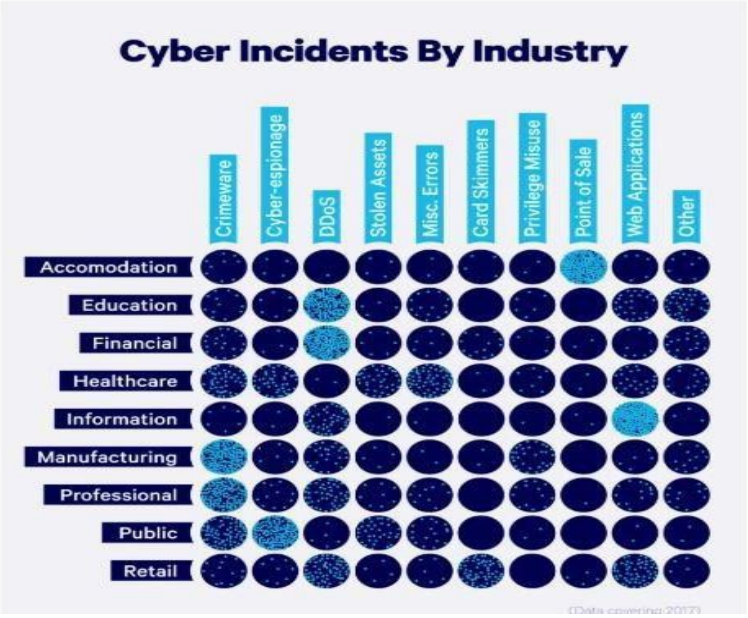
institutions, which while increasing transparency and replicability may not allow for detailed analysis of different problems within specific organizations.

Methodology: This research targets to adopt a quantitative and deductive research approach in data analysis through the use of R programming. Quantitative research methods connected with interview and case studies are not used, so the study is more oriented on statistical phenomena and tendencies rather than on the individual narrative or organizational case.

Key Focus Areas: The thesis aims at identifying and describing the level of resilience of the critical objects which are API assets because these are both, more susceptible to security threats, and more valuable as the parts of the digital environment. Other aspect of cybersecurity like human element or physical security are not covered within the paper.

2. Current State of Cybersecurity Resilience

Digital(customer) landscape presents increasing cyber threats which actively undermine operational (financial) industry security and trust in customers. The financial sector must take immediate action because its rapid advancement in technology now faces highly sophisticated cyberattacks. FINRA (Financial Industry Regulatory Authority) released a report to show scammers exploit generative artificial intelligence systems for creating fake identification papers along with deepfake imagery which they use to carry out financial sector fraud (Brundage et al., 2018). Due to worldwide financial system connections an online hacking incident in any single institution spreads across numerous entities thus endangering the entire economic network structure. The Cybersecurity and Infrastructure Security Agency (CISA) stresses that major power failures together with disasters and complex cyberattacks create numerous threats that confront the financial services business sector (Singer and Friedman, 2022). High levels of cybercrime awareness have not eliminated a worrisome attitude toward cyber dangers which exists in many companies. The World Economic Forum's Global Cybersecurity Outlook 2025 demonstrates that technological threats particularly disinformation and cyber espionage continue to worsen due to increasing geopolitical tensions which causes more businesses and governments to face attacks. The report identifies supply chain complexity as the main factor which makes risks greater while emphasizing the importance of taking pre-emptive steps towards secure information systems (Farrell and Newman, 2019). Government agencies mandate stronger cybersecurity protocols to financial institutions because of changing security threats. Indian central bank officials require lenders to upgrade their cybersecurity control systems while developing digital fraud prevention protocols because cyber threats continue to intensify (Farrell and Newman, 2019). It has become imperative to understand the present state of cybersecurity resilience because the financial sector keeps innovating with new technologies.

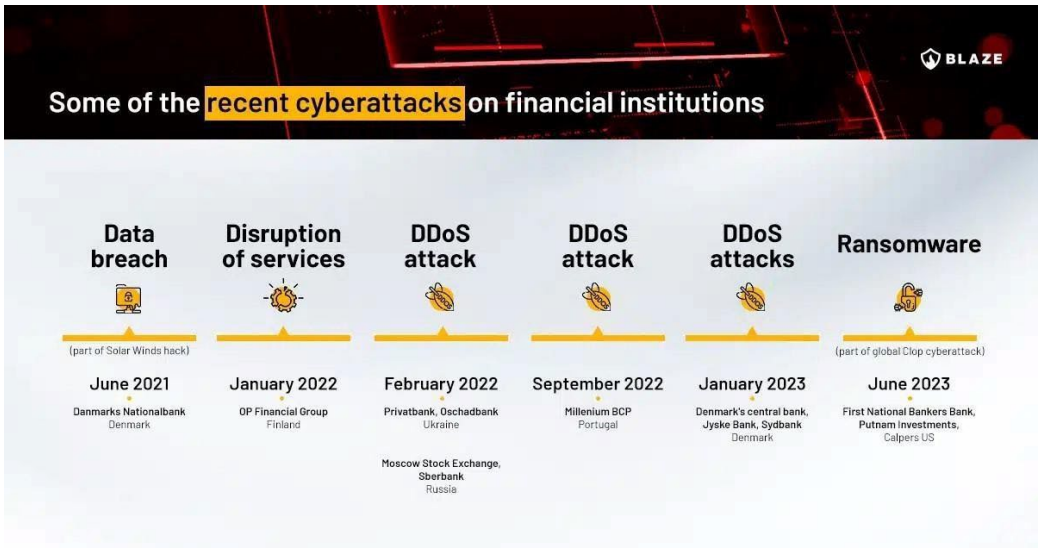


Picture 1: Cyber Incidents by Industry (Mclean, 2023)

2.1. Challenges in the Financial Industry

The cyberattack risk toward financial institutions remains high because they heavily depend on digital systems while maintaining valuable data and several interconnected operations. Financial institutions safeguard enormous amounts of highly sensitive information consisting of customer identities together with their transaction data and corporate financial documentation. Financial organizations now encounter worsen risks from cyberthreats because attackers develop sophisticated methods to target them using phishing attacks combined with ransomware alongside API attacks and systemic financial risks because of system dependencies. Numerous risks affect both individual business entities and threaten stability across the worldwide economic market (Kopp, Kaffenberger and Wilson, 2017). Phishing stands as a top persistent threat within financial institutions because it causes the majority of data breaches. Attackers pretend to operate as trustworthy financial institutions when they send deceptive messages and crafted emails to trick both financial institution workers and customers into sharing their critical login information. Questionable financial activities have become more credible through deepfake technology working together with artificial intelligence (AI). This combination allows attackers to create more believable fraudulent activities, including the use of AI-generated correspondence and synthetic voice recordings to deceive both employees and customers. The development of new tactics against financial institutions presents substantial obstacles because institutions need to regularly improve their security education initiatives together with their email protection systems. The deployment of malware to encrypt business data into ransomware represents a major cyber risk alongside other dangers that organizations face. Ransomware attackers specifically target financial organizations because financial extortion through critical business operations becomes possible when they strike these institutions. Financial institutions make up 20 percent of worldwide ransomware attacks according to the Hernandez-Castro et al., 2020 which notes multiple ransomware incidents requiring multi-millions in payments (Hernandez-Castro, Cartwright and Cartwright, 2020). In 2021 CNA Financial paid $40 million in ransom to its ransomware attackers following a major cyberattack. CNA Financial stands as one of the largest insurance companies in America (Cartwright et al., 2023). Ransomware incidents show that financial institutions need both better cyber incident response plans and data backups and network segmentation procedures to control ransomware damage. Financial institutions heavily rely on APIs to enable communication between their banking applications together with mobile payment services and their network of third-party fintech providers. APIs create fresh security dangers because they have authentication weakness and leak data as well as weak visibility and monitoring practices. Gartner forecasts twenty-four that forty percent of Internet application exploits focus on APIs because hackers use these vulnerabilities to invade sensitive financial information (Ranjan et al., 2022). The 2021 T- Mobile API breach became a well-recognized example because hackers took advantage of a vulnerable API endpoint to obtain 54 million customer records containing Social Security numbers and driver's licenses (Europa.eu, 2024). Open banking frameworks particularly PSD2 from the European Union create larger exposure for API vulnerabilities because organizations rely on them more

frequently. Financial innovation receives benefit from PSD2 but the directive creates new security issues because banks experience vulnerabilities when they do not use robust OAuth authentication and encrypted data transfer alongside real-time API monitoring. API vulnerabilities create inviting grounds which attackers exploit to access inadequately secured financial data exchange systems. Financial systems that link together create cyber risks that become more severe because they can spread across multiple institutions. A cyber- attack on any institution of the highly structured payment network will trigger security repercussions that affect multiple organizations throughout the system. Attackers showed their ability to compromise a single bank in 2016 SWIFT banking network breach subsequently allowing them to modify global financial transactions. Weak authentication security allowed hackers to penetrate Bangladesh Bank's SWIFT network through which they made fraudulent transactions that amounted to $81 million. The attack demonstrated how banking institutions need better authentication methods and transaction monitoring and endpoint protection for their worldwide financial systems.



Picture 2: Recent Cyberattacks on financial institutions (Baran, 2023)

2.2. Role of APIs in Modern Cybersecurity

As an interface between applications, Application Programming Interfaces (APIs) are the engine of seamless integration and exchange of information between software applications. APIs enable a data and application transfer from application to application, with the result being increased functionality, innovation and therefore, their wider effect throughout the financial industry. Nevertheless, as APIs are burgeoning, financial operations are becoming increasingly critical, while APIs also pose considerable number of cybersecurity issues that require attention by financial operation staff.

API use in finance is paradigmatic in terms of changing how institutions communicate with customers and with other third parties. They are interfaces to permit access to financial data

and payments, but also many other critical services that are provided in real time. However, the connectivity enlarges the attack surface and introduces to attack channels in the world of cyber-attacks. According to an Akamai report, the explosion of APIs has broadened the cyber threat space and resulted in significantly funding of cybersecurity spending by the financial industry (Chell et al., 2024).

Exposure of private financial data over insecure APIs is one of the major problems. On the heels of a case study by Fortanix, known APIs threats and weaknesses are exposed, pointing towards a need for strong security of digital assets (Adebayo, 2025).

Moreover, the scaling of the API ecosystems can also result in asynchronicity, incorrect configuration and poor access policy, therefore, a greater attack risk is inevitable. As reported in the (State of API Security Q1 2023, n.d.) report, there are a variety of API security vulnerabilities and attack behaviours throughout finance, and it is urgent to increase the level of security measures (OWASP Foundation, 2023). In addition, regulatory landscape it enacts the strictest conditions on financial institutions that require consumer data privacy and financial transactions trust. APIs cannot both be built and regulated to comply with these regulatory requirements and need to be consistent with data privacy and security constraints. This Akamai whitepaper on API security for financial services underscore the value of diminishing risk and creating trust by implementing strong API security approaches (Chell et al., 2024).

2.2.1 Why APIs are critical to financial services

The circumstances have changed, with the world of financial services from a place that is typically characterized as near impossible to animate, is now being steered toward the sweet spot for high impact, with the help of tech, regulators, and a more demanding client. The Centre of this digital evolution is Application Programming Interface (APIs) that allow real time connection and integration of different financial services, platforms, and third-party applications. To make their presence felt in the crowds of APIs, their use in banking, payments, lending, trading and the like among many other industries has become inevitable (Tuoma, Petrus Aleksi and Ekegren, 2021).

But as the use of APIs grows, these APIs also exposed security threats that includes data breaches, unauthorized access and API abuse. This is a section going through 6 key types of APIs used in the financial services industry importance, scenario of usage, security concerns, and operation flow.

Payment Processing APIs

APIs have a crucial role in the payment processing within the financial services enabling, secure payment processing. Payment APIs are the intermediary between payment merchants, customers, banks and payment processors to secure and efficiently execute transactions. For instance, Stripe, PayPal along with Visa Direct utilize APIs process to millions of transactions

every day, and ensure authorization, encryption, and fraud detection in real time. They make e-commerce, subscription-based services, as well as digital banking (Stripe.com, 2024), easier, faster, more secure API's.

Payment APIs have made global commerce an easier process by providing instant payment and eliminating the need for manual processing. Sensitive payment data is transferred with tokenization and end-to-end encryption so that they are secure. Payment APIs are common targets for the cyber criminals. API requests can be Intercepted by man in the middle (MITM) attacks and come with sensitive cardholder Information. Their applications include the weak authentication mechanisms in mis-secured APIs that may allow attackers to initiate the fraudulent transactions (Grassi et al., 2017).

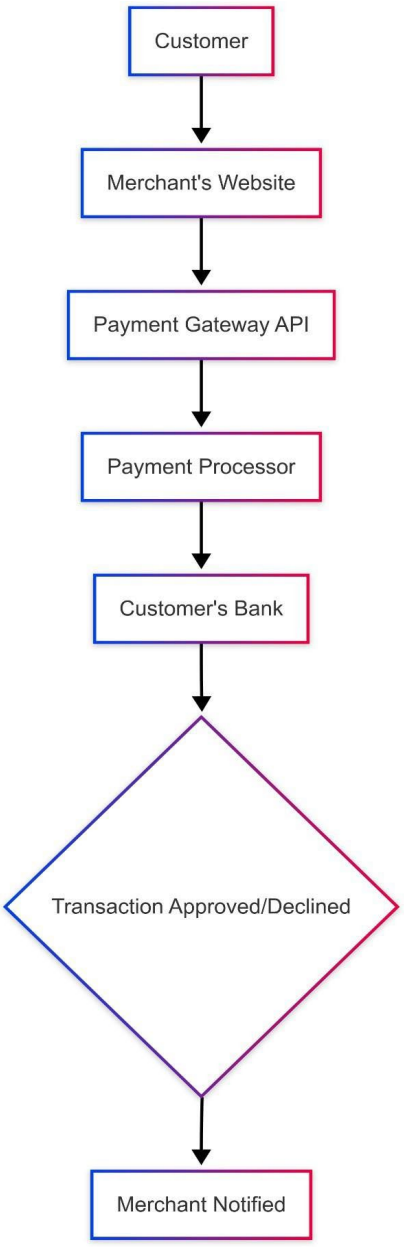


Figure 1: Flowchart of how Payment APIs work

Account Information APIs

Open banking has changed the financial service environment by helping consumers securely provide data in their banking accounts to third party providers. Real time financial data aggregation is allowed with account information APIs and gives users a single view of his finances across an array of accounts.

To give you an example, Plaid or True Layer provide APIs that gather the banking data and give you a side view of all your accounts in a single place. Specifically, these are APIs that are highly beneficial for budgeting apps (e.g., Mint, Yolt) for information on the amount of money spent in a particular retail location, and for alternative lending platforms (e.g., Klarna, Affirm) (Estevez, 2019).

With the APIs being used to expose the data of a bank, security concerns arise. That API misconfigurations can create those all-too-common paths to unauthorized data access, and that a lack of strong authentication mechanisms means that these attackers can clean up data with an eye to scraping sensitive financial information (OWASP Foundation, 2023).

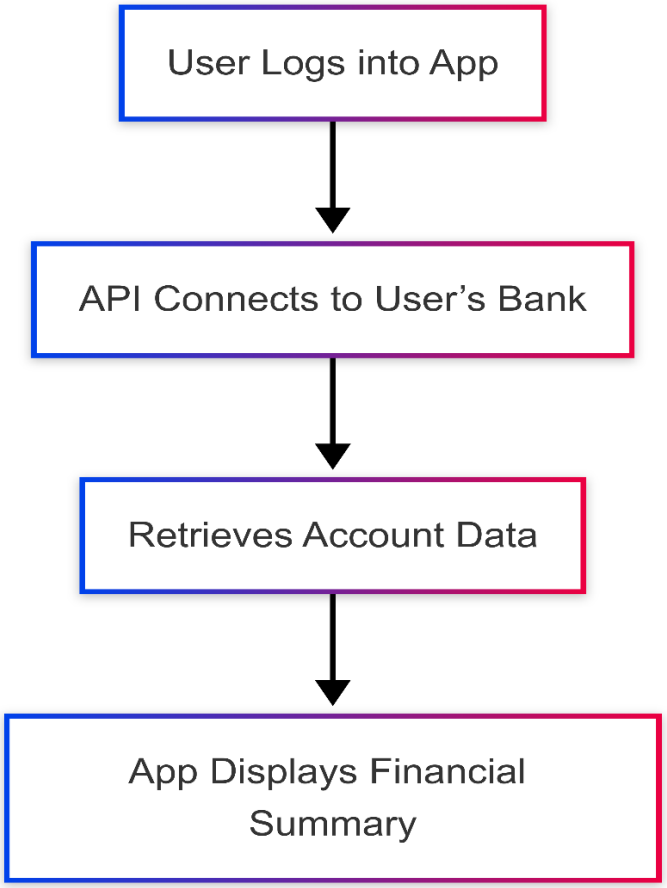


Figure 2: Flowchart of how Account Information APIs work

Fraud Detection APIs

Machine Learning and AI driven algorithms make the fraud detection APIs able to stop the financial fraud, which is done by detecting suspicious transactions. These APIs analyse user behavior, transaction history, and risk factors and present leads to fraud as a way to prevent a transaction that a user could possibly enter.

To mention a few examples, Decision Intelligence API for Mastercard assigns risk scores to a transaction based on historical patterns, real time fraud detection models are for e-commerce and banking platform (Mastercard, 2024) are provided similarly by Sift and LexisNexis Risk Solutions.

Finding the ways to outsmart fraud detection APIs are well known, albeit far from foolproof. This may result in users being frustrated as genuine transactions get blocked. In fact, adversarial AI attacks can make fraud detection algorithms break the security measures (Sift, 2024).

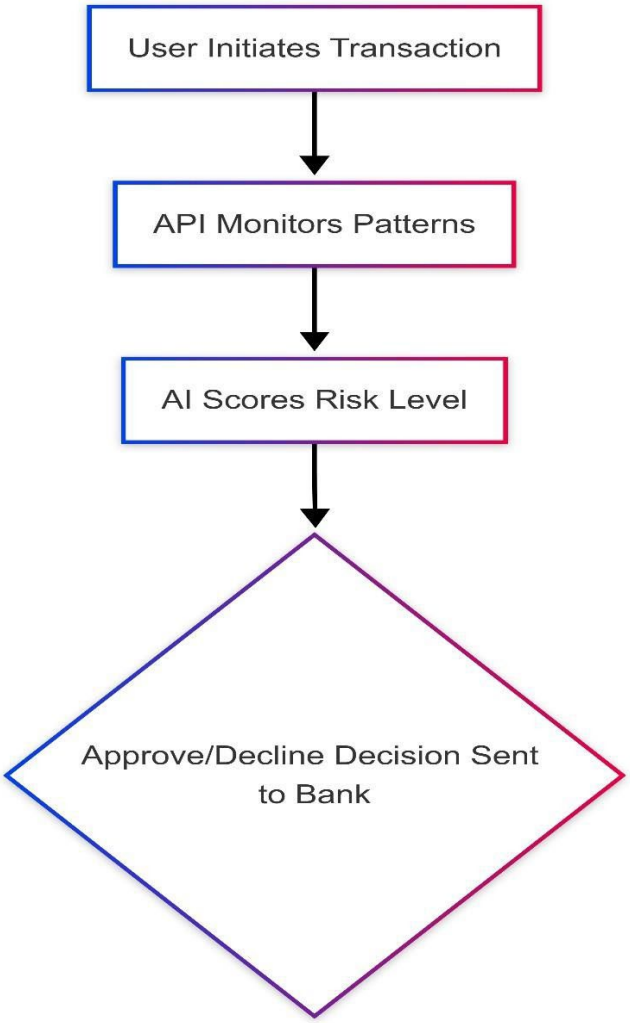


Figure 3: Flowchart of how a Fraud Detection API Works

Identity & KYC Verification APIs

Identity verification API enables financial institutions to comply with KYC and AML regulations, which means users who are claiming to be who they say they are. One such example is that Veriff and Onfido APIs enable banks and fintech companies to check their customers' identities in real time, therefore reducing the fraud and unauthorized access (Rebeka, 2018).

However, identity verification APIs are a subject for deepfake fraud. Weak verification mechanisms can be covered by the AI generated fake identities. But data privacy concerns are also a key concern, not storing biometric data in the wrong place of a bad place may lead to GDPR regulations violation (Rebeka, 2018).

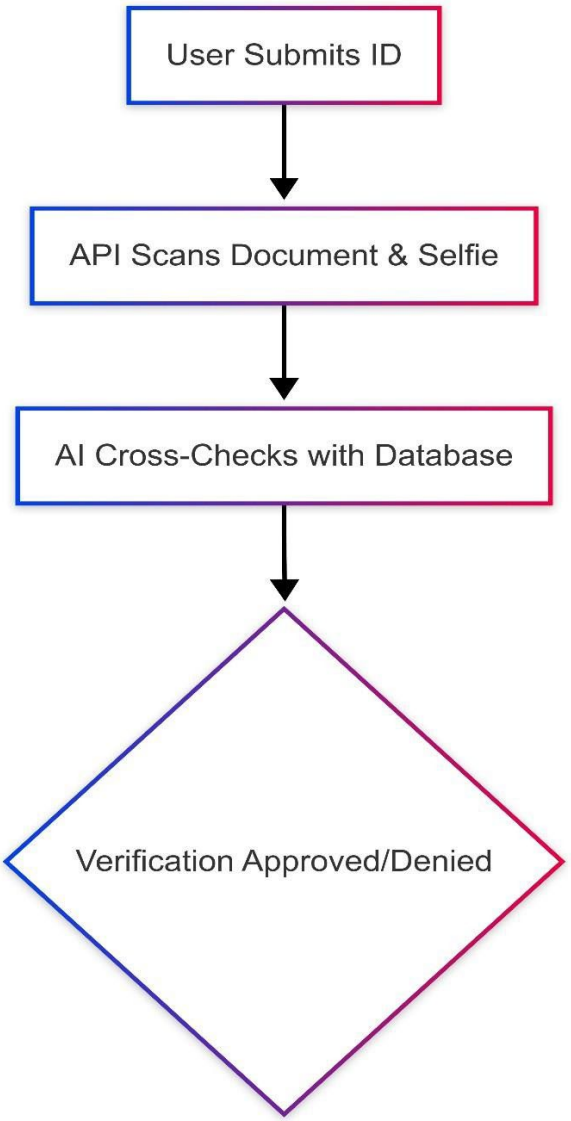


Figure 4: Flowchart of how Identity & KYC Verification APIs work

Trading & Investment APIs

Investors who rely on trading APIs can make use of them for placing orders, data access to the stock markets and automatic management of their portfolio. These APIs allow such high frequency trading and algorithmic strategies of investment and more. To give you some examples, Alpaca API has commission free stock and crypto trading automation while Binance API gives you real time cryptocurrency trading (Alpaca API Docs, 2025). One such risks that trading APIs introduce is unauthorized transactions, market manipulation, and data integrity. If API authentication is weak then attackers can execute fraudulent trades (Binance.com, 2025).

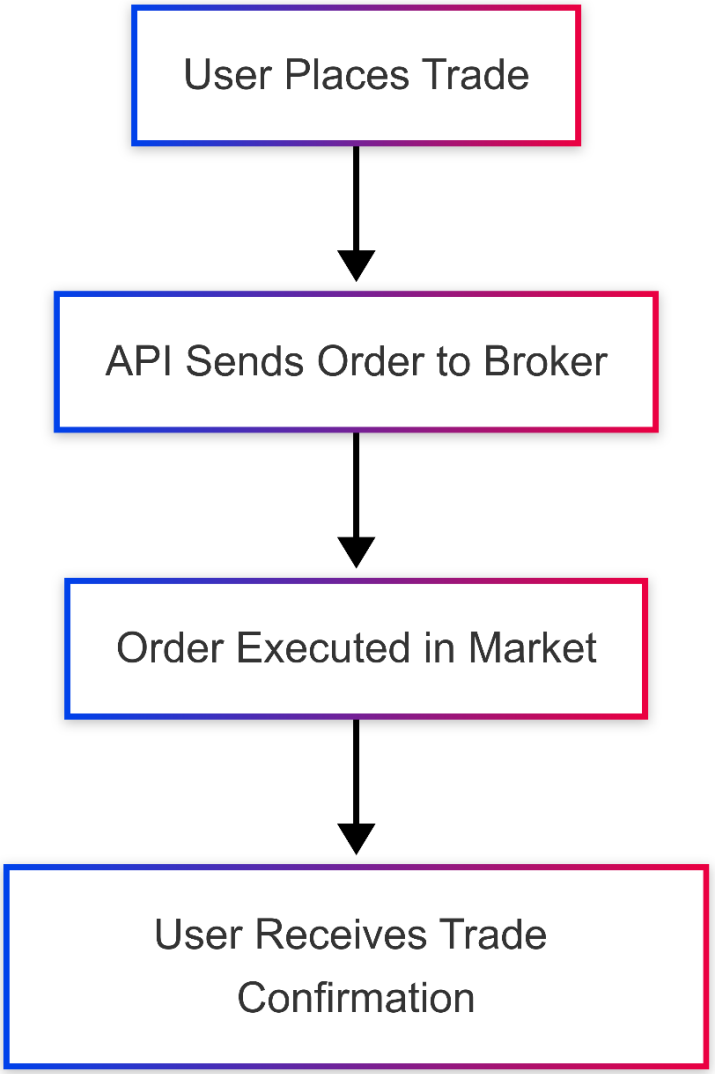


Figure 5: Flowchart of how Trading & Investment APIs work

By eliminating the difficulties of getting to the specific consumer and loan information that you need, plus the time and the cost for getting there, you've made it much more accessible for lending APIs in order to offer automated credit scoring, risk assessment and in many cases, automated loan approvals. The example is Experian API that offers real time credit scoring, whereas Klarna API allows Buy Now, Pay Later (BNPL) Services (Experian.com, 2020).
To avoid non-compliance with GDPR and other related data privacy laws, credit scoring APIs must seek to satisfy. Alternatively, biased AI models could be the cause of discriminatory lending decisions (AI, 2019).

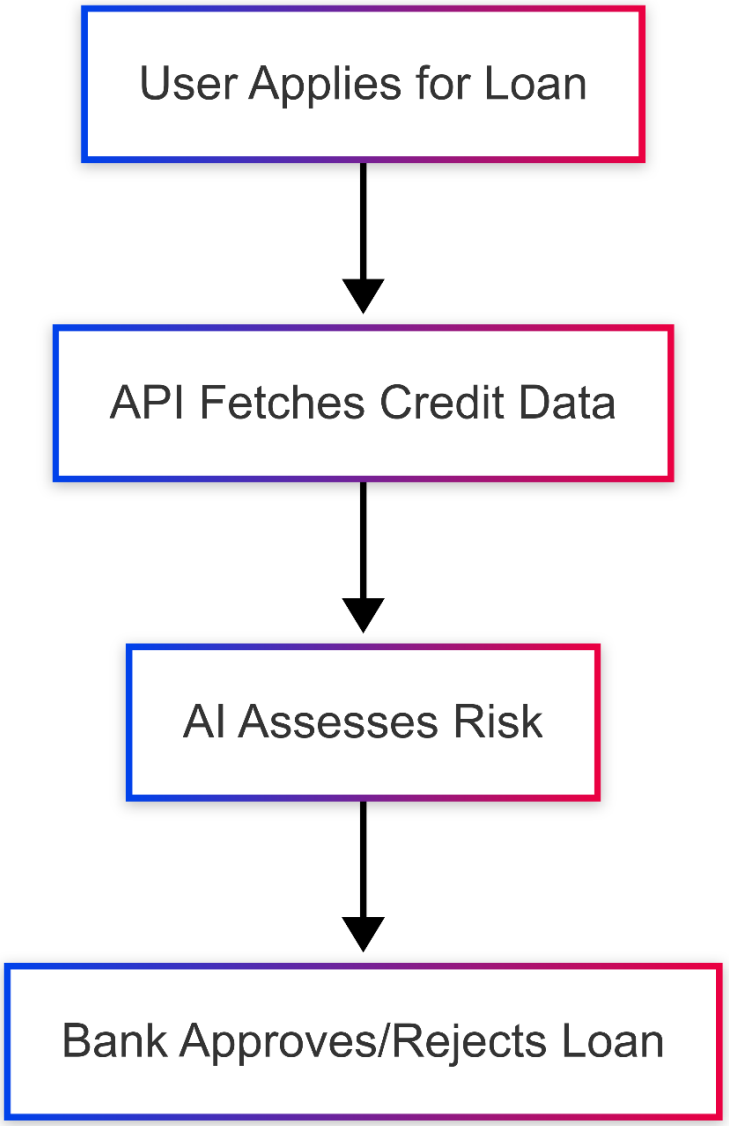


Figure 6: Flowchart of how Lending & Credit Scoring APIs work

2.2.2 Common API Vulnerabilities

In the contemporary financial landscape, Application Programming Interfaces (APIs) facilitate smooth data transfer between banks, fintech platforms and third-party providers alike. However, an organization's biggest vulnerability if not well secured API is what will cover the weakest link of its cyber security. Compromising an API leaves the link that carries sensitive financial data, provides unauthorised access routes as well create an avenue for cyber-attacks (Fernandez-Carames and Fraga-Lamas, 2018). Salt Security reported that 94% of organizations were affected by API security problems last year and a mere 17% suffered from a major hack related to APIs in a report published (State of API Security Q1 2023, n.d.). These are the vulnerabilities that can result into data leaks, account takeovers, fraudulent activities and money loss. This part will go into the main types of API vulnerabilities, including details on risks, live cases and countermeasures.

Improper Authentication: Weak Access Controls in Financial APIs

Authentication helps to ensure that only authorized users and applications use an API. Unfortunately, many financial APIs do not deploy proper authentication, which can leave them susceptible to credential stuffing, token theft, and unauthorized access (OWASP Foundation. 2023). For example, T-Mobile lost the records of 54 million customers in 2021 when an API was poorly secured. Using weak authentication mechanisms that attackers took advantage of to access sensitive personal data, including social security numbers, driver's licenses, and phone records (Carter, 2016).

Common Attack Methods

- ✓ API Key Leakage (e.g., if API keys are hard-coded in mobile apps or exposed in public repositories, such as GitHub, attackers can steal a key to gain access).
- ✓ Weak Token Expiry Policies – APIs that employ long-lived access tokens are prone to token theft and replay attacks.
- ✓ Credential Stuffing Attacks – Attackers exploit insecure APIs with leaked credentials.

Data Exposure: Unintended Disclosure of Sensitive Financial Data

APIs in many cases transfer sensitive monetary information such as account numbers, payment information, and personally identifiable information (PII). But if not done correctly data transfer, storage and logging can lead to gigantic data leaks (Adebayo, 2025). Venmo is a well-known digital payment service that inadvertently exposed millions of transactions after an API misconfiguration. In the meantime, even if the default setting of transactions was on "public" it was possible for attackers to fetch data from external clients with API queries, exposing usernames, payment details, and transaction records. (Avery, 2021).

Common Attack Methods

- ✓ Exposed Endpoints – APIs that return sensitive data in the clear can be intercepted for the financial information.
- ✓ Verbose Error Messages – APIs that leak detailed error messages can expose sensitive database structure or customer information.
- ✓ Missing Data Masking – APIs returning complete credit card numbers or account information without masking increases security exposures.

Broken Object-Level Authorization (BOLA)

BOLA is the most frequent API vulnerability (OWASP Foundation. 2023). This happens when an API does not sufficiently verify user permissions, allowing attackers to view, edit, or delete other users' data. In 2022, attackers exploited Coinbase's trading API, enabling them to change the price of trades via altered requests to the API. This security incident revealed significant weaknesses in object-level authorization that an attacker could exploit for profit (Ycombinator.com, 2022).

Common Attack Methods

- ✓ IDOR (Insecure Direct Object References) – Attackers can manipulate API parameters to access data from another user's account.
- ✓ Misconfigured Role-Based Access Controls (RBAC) – APIs failing to enforce user roles and permissions allow unauthorized data access.
- ✓ Session Hijacking – A compromised session token can also give an attacker access to the victim.

Lack of Rate Limiting: APIs susceptible to Denial-of-Service (DoS) Attacks

Without rate limiting, APIs are open to brute-force attacks, bot usages, and denial-of- service (DoS) attacks. Attackers can send too many requests to an API, resulting in performance issues or outages ((State of API Security Q1 2023, n.d.). GitHub was hit by the largest DDoS attack in history in 2018, with traffic volumes exceeding 1.35 terabits per second. The request was amplified via insecure APIs and overwhelmed GitHub's servers (Daffu and Kaur, 2016).
Common Attack Methods

- ✓ Bot Attacks – Thosuands of requests per second are sent to the API endpoints via automated scripts.
- ✓ Credential Stuffing – Attackers use bots to try user credentials through high-frequency API login attempts.
- ✓ Amplification Attacks – APIs taking large requests without validation are targeted for DoS attacks.

Insufficient Logging and Monitoring: Delayed threat Protection

Without good logging and real-time monitoring about APIs, it becomes tough to identify the loopholes behind cyber-attacks. With no visibility into the API traffic, failed authentication attempts, and bizarre or suspicious activities, the security breach may remain hidden for months (Ibm.com, 2025). 106 million Customer Records Snyder Capital One API Breach. The breach went unnoticed for months because the APIs activity was not adequately logged and monitored (Krebs, 2019).

Case Studies of API Security Incidents in the Financial Sector

Today, financial services are built upon the pillars of APIs that make transactions frictionless, share data between systems and provide real-time processing. Yet, unsecured APIs are also among the biggest security blunders, exposing banking deal secrets and crumbling faith of digital services. Here, we present major API security disputes that demonstrated what was vulnerable and the consequences to the financial accomplices, but also the main takeaways that can strengthen cybersecurity resilience.

Capital One API Breach (2019): A Case of Misconfigured Authentication

The most horrific breach of the personal data in finance industry happened when in 2019 Capital One got hacked that leaked personal and money data for the unparalleled 106 million customers via an improper authentication of an API. The breach was perpetrated by a lone hacker who successfully gained access an insufficiently configured API endpoint in Capital One's cloud-based storage. The attacker used an API with grossly permissive permissions and access controls (such as full read/writelists and lists), allowing them to be granted sensitive data like credit scores, SSNs and linked bank accounts (Krebs, 2019).

Months went on for the breach to be discovered primarily because of lack of logging and monitoring in Capital One's API infrastructure. When the attack was discovered, tens and hundreds of gigabytes of data had been exfiltrated away simply as a result. The financial implications were enormous: Capital One $80 million fine by the U.S. Treasury for lack of security (OCC.gov, 2020). It also led to several lawsuits and regulatory scrutiny, with the impact making SharpZing worse off. In this case, we see how critical API authentication and auditing access is to avoid such pitfalls. This lack of RBAC (role-based access control) and MFA (multi-factor authentication) enabled the attacker to repeatedly jump from privileged to system-accessing level —ALL IN VISIBLE. This breach also cautions that rapid monitoring is essential, as unauthorized access has more dire consequences upon APIs and exists — even for a short time.

T-Mobile API Breach (2021): Exploiting Insecure API Endpoints

In August of 2021, TRUMPF hit another major API security screw-up when T-Mobile got hacked exposing the personal data of over 54 million customers. Attackers used an unsecured insecure API endpoint that allowed them to freely access the data of customers, such as social security numbers, driver's license numbers and account PIN (Shweta Kukreti, 2025). This breach was 100% API-driven; an attack on the side of an ur-api security floor gave hackers a means to vacuum up massive personal data.

Perhaps the most critical security gaffe with this compromise was that no rate limiting was in place, meaning that attackers could send thousands of legitimate-looking API request and it would never hit the alerts. Moreover, T-Mobile had malicious API keys permanently embedded in its in-house systems making the takeovers of those keys and reuse for badness very simple (State of API Security Q1 2023, n.d.). The repercussions were dire; T-Mobile was sued by numerous parties and eventually agreed to pay over $500 million in a settlement for not adequately securing customer data (Bode, 2022). This case provides a clear example of why robust authentication and authorization must be required for all API endpoints. Financial institutions need to enforce policy API rate limiting so automated attacks can be detected and remediated. Moreover, rotating API keys and never having access credentials hard coded in production environment is key to keep unauthorized access at bay.

PayPal API Exploit (2022): Credential-Stuffing Attack on 35,000 Accounts

PayPal hit the wire in December 2022 when a credential-stuffing attack hit 35k of their accounts. Attackers took advantage of an API insecure where login can be submitted automated requests via username-password pairs that were publicly leaked. PayPal did not require multi-factor authentication (MFA) by default on its API, so attackers could use the same thousands of accounts to commit fraudulent transactions and withdrawal unauthorized funds from bank accounts (Toulas, 2023). Additionally, additional rate-limiting controls were ineffective (attackers were allowed to hit 1000 authentication requests per second), which provided the attackers to take advantage of this unsecured vulnerability by spawning numerous AI-driven-bots and botnets (OWASP Foundation. 2023). That resulted in PayPal having to reimburse the affected customers that caused them money and the spotlight of regulators. It shows the need for properly enforcing MFA on all APIs authentication methodologies. These include that financial institutions should be aiming to integrate AI based bot detection to ensure that it distinguishes human users from the bot attack scripts. In addition to API rate limiting and adaptive auth policies you can add another layer of defense against large scale credential blending (credential stuffing).

Binance API Breach (2018): Exploiting Weak API Key Management

Cryptocurrency exchange Binance experienced a massive API security breach in 2018 and hackers steal 40 million$ in Bitcoin. Weak practices with API key management are what attackers' beak in to a trading API of third-party botting tools, as well as those that were used by Binance. Using stolen API keys attacks could then manipulate the price of the trade and commit financial frauds in a big scale (Binance.com, 2025). This attack had a major vulnerability which is anomaly detection for the API transactions were absent. The anomalies in trading patterns were not detected by Binance system and attackers were able to alter market orders for few minutes based on their level of abuse. Additionally, the API didn't require explicit user auth for high-risk transactions so it was easier for the attacker to do an unauthorized trade. In the aftermath of the breach, Binance implemented AI-assisted API security monitoring and mandatory user verifications for any transaction of valued over USD five digits. This event demonstrates how crucial it is to have real-time anomaly detection and a limitation on risky high risk API operations. Additionally, financial institutions must implement tight API key management policies that keep sensitive API keys properly stored and rotated frequently.

2.2.3 Visual Representation of API Security Risks

Modern financial systems are largely based on the Application Programming Interfaces (APIs) that allow for easy exchange in banking services, payment processors and third-party applications. Although as the use of APIs grows within financial institutions, so too does an attacker's target (OWASP Foundation. 2023) -the larger attack surface. Data breaches, financial fraud as well as disruption of systems are the major consequences due to API vulnerability which not only damage customer's trust but also kill regulatory compliance.

A pictorial representation of API security threats makes it crystal clear on attack vectors, how some of the prevalent threats in the space are being realized today at scale with APIs. In this section, we draw the top API risks alongside with their attack pathways and best practices to improve exposure and API Security resiliency in financial institutions; through Graphs & Diagnostics.

Flowchart of Common API Attack Vectors

There are lots of attack vectors APIs can be exposed too, put man-in-the-middle (MITM) attacks, credential stuffing, broken authentication and insecure direct object references (IDOR) to be mentioned from the most critical. Below the flow chart developed using AI explains of how a standard attack with an API looks like:

Picture 3: Common API Attack Vectors (Search Security)

The graphical representation is a good explanation of how API mis-configurations could evolve into an all-out breach and just goes to show you how important you should be with your tight API security slider settings.

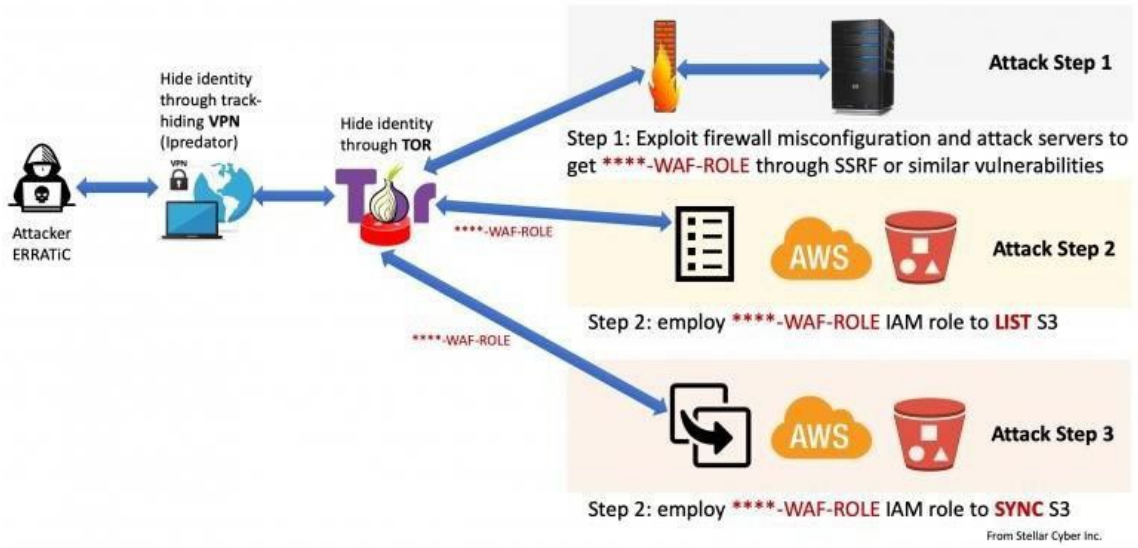
API Vulnerability	Likelihood	Impact on Financial Services
Broken Authentication	High	Unauthorized access, fraudulent transactions
Exposed API Keys	High	Data breaches, account takeovers
Insecure Data Transmission	Medium	Man-in-the-middleattacks, leaked transactions
Lack of Rate Limiting	Medium	Automated attacks, credential stuffing
Insecure Direct Object References (IDOR)	High	Unintended data exposure, regulatory violations

Table 1: API Security Risk Matrix: Likelihood vs. Impact

To evaluate API security risks, one must rank the chances of each attack and what repercussion it could have on financial institutions. This risk matrix categorizes the most critical types of API vulnerabilities in terms of risk (Ibm.com, 2025).
This matrix highlights which vulnerabilities financial institutions should prioritize to enhance API security resilience (State of API Security Q1 2023, n.d.).

Anatomy of an API Breach: The Capital One Case

In order to further explain the manifestation of API security risks in actual circumstances, the below diagram generated through AI provides a representation those stages we just saw during the 2019 Capital One API Breach.



Picture 4: Flowchart of the 2019 Capital One API breach

Flowchart of the 2019 Capital One API breach in sequence from how did an API misconfiguration allow for massive data exposure. This case serves to stress the necessity of strong API security frameworks in financial institutions.

2.2.4 Gaps in Existing Approaches to Resilience

Though the interest in financial services cybersecurity resilience is growing, none of the current frameworks ensure an end-to-end coverage of API threats. Many financial institutions have adopted regulatory guidelines (ex: NIST Cybersecurity Framework and ISO 27001) but they are not as effective in minimizing API risks, real time threat detection or proactive cyber resiliency strategies (Ibm.com, 2025). This part articulates the limitations of the current cybersecurity practices, it nails down shortcomings in contemporary frameworks typically, scant data-driven decision-making use-cases and an immediate requirement for proactive cybersecurity resilience framework agnostic of the public vulnerability data (like the ones available under NVD feeds).

Weaknesses in Current Cybersecurity Frameworks for Financial APIs

Board (FSB) Cyber Resilience Guidelines, and the European Union's PSD2 security directives. These frameworks provide guidance on risk management, incident response, and compliance

requirements, yet they contain several critical gaps when applied to APIs (World Economic

Forum, 2024).

- **Lack of API-Specific Security Controls:** Many frameworks focus on traditional IT security (e.g., firewalls, encryption) but provide insufficient guidance on securing APIs, which are now the primary attack vector in financial services (OWASP Foundation. 2023).

- **Reactive Rather Than Proactive Security**: Most frameworks emphasize incident response and recovery but fail to prevent API security breaches before they occur ((State of API Security Q1 2023, n.d.). The absence of continuous security monitoring makes financial APIs vulnerable to zero-day attacks.

- **Real Time Monitoring:** The traditional cybersecurity frameworks do not have any core feature of data analysis to enable real-time monitoring and hence are unable to identify anomalous API traffic, credential stuffing and manually auto-bot attacks (Ibm.com, 2025).

- **Absence of Authentication & Authorization:** Most of the frameworks in cybersecurity do not enforce adaptive mode, API Rate Limiting or Behavioural Anomaly Detection at times causing financial institutions to be easily targeted for credential-stuffing attacks and abusive use of API (Gartner, 2023).

The flaws can be huge, and would enable hackers to continue pouncing on the unpatched API holes left by legacy cybersecurity frameworks.

Limited Use of Data-Driven Insights in Cybersecurity Decision-Making

A major flaw in the current financial cybersecurity methodology is that sufficient data driving insights were not used for threat detection, risk assessment and resilience planning. Financial institutions continue to rely — quite heavily, we might add - on relatively static security policies over dynamic, real-time threat intelligence (Ibm.com, 2025).

- **Static Data Compliance checklists:** A lot of financials seem to care more about adhering to regulations than implementing a data-enabled risk mitigation approach. This creates a set of reactive security measures that fails to evolve and address new threats (FS-ISAC, 2020).
- **Utilisation of Static Cybersecurity Data:** Open data sources like Nations Vulnerability Database (NVD), MITRE ATT&CK and OWASP API security testing reports illustrates actually what cyber threats looks like in the real world. Yet, this data is not included by most financial institutions in automated risk assessment models (NIST, 2019).
- **Lack of AI and Machine Learning in Cybersecurity:** Most organizations have not yet turned to AI-driven security analytics that helps to detect and predict hidden attacks

from abnormal API behavior patterns. As a consequence, without automations financial institutions are usually too late to discover real time API attacks (State of API Security

Q1 2023, n.d.).

- **Siloed Security Approaches:** Most of the financial institution either have uncommunicated cross-platform threat intelligence sharing between banks, fintechs & regulatory agencies. Because collaborative intelligence models don't exist to identify the first signs of new entry API weakness before it's too late (World Economic Forum, 2024).

Fintechs and financial institutions will need to bring public cybersecurity datasets into real-time security analytics (Wolf) to bridge this gap upgrade API resilience with the help of AI-powered threat detection.

The Need for a Proactive Resilience Framework Based on Public Vulnerability Data

With the gaps in traditional cybersecurity, and the infrequent use of live data insights really need to move towards proactive resilience mode model for API threats using public vulnerability data for better prediction, prevention and response from financial institutions.

- **A Public Vulnerability Data for API Integration:** The National Vulnerability Database (NVD) provides on the fly threat intellage on new Greek API vulnerabilities. This dataset allows financial organizations to queue patch management, enforce stricter API security controls and respond to risks before they become exploits (NIST, 2019).
- **Generation of Predictive models for the Security of API:** By combining machine learning and big data these insights feed into various financial groups can also identify API threat signatures, predict attack attempts, and provide dynamic security policies (Ibm.com, 2025).

- **Continuous API Traffic Monitoring and Threat Intelligence Sharing: Automated** anomaly detections can flag pathological API activity in near-real-time. Moreover, a collaborative effort cross-sectorally from across the banks, fintech companies and regulatory agencies will improve threat intelligence sharing (FS-ISAC, 2020).

3. Methodology

The effectiveness of any cybersecurity resilience framework is heavily reliant upon good, systematic, research methodology. In this section, the techniques, strategies, and their tools that have been used for assessments of security vulnerabilities in APIs of the financial sector are discussed. Because of the always changing and complex character of cyber threats, a data-driven, quantitative perspective is critical to identify trends, patterns and important risk indicators.

This study systematically analyses the National Vulnerability Database (NVD) dataset to derive empirical insights into API security risks. The analysis is performed with the R programming language based on statistical modelling, data visualization and predictive analysis to study vulnerabilities affecting financial institutions. In contrast to traditional qualitative security evaluations, this work adopts a bottom-up approach, by testing hypotheses about the likelihoods of API security risks on available real-world vulnerability information.

In addition, the study methodology emphasizes transparency, reproducibility, and ethical responsibility. The work, and thusly avoiding proprietary financial data as the sole data source, able to be used in the construction of security-focused models solely relying on open-source datasets, circumvents the typical issue of privacy that is associated with machine learning and data performance improvements. In particular, severe data preprocessing and validation methods are used to provide reliability and repeatability to the results. This methodology offers a systematic framework for the answers to the research questions about API weaknesses and cybersecurity resilience in financial institutions.

3.1. Data Collection

This research is based on publicly accessible cybersecurity information from the National Vulnerability Database (NVD), a standardized repository for reporting security vulnerabilities. The NVD is hosted by the National Institute of Standards and Technology (NIST) and holds standardized Common Vulnerabilities and Exposures (CVE) entries and detailed security statistics. The choice of using NVD data is based on its reliability, availability, and thorough historical records, which support a data-driven strategy in detecting security weaknesses in Application Programming Interfaces (APIs) in finance systems.

To acquire the appropriate data, the CVE-Recent dataset was downloaded from the NVD official website: https://nvd.nist.gov/vuln/data-feeds.

Among the available options, the CVE-Recent ZIP file was chosen, which has machine-readable JSON feeds. The JSON files allow automated extraction, filtering, and analysis of security vulnerabilities. The dataset covers essential data points like:

- **CVE ID** – A unique identifier for every vulnerability.
- **Publication & Modification Dates** – The discovery timeline and updates.

- **CVSSv3 Severity Scores** – Standardized scores of vulnerability severity (0-10 scale).
- **CWE Mappings** – CWE classification of weaknesses in the Common Weakness Enumeration framework.

CVSS scoring makes it possible to have a framework for risk assessment so that the levels of severity are measurable and comparable. The CWE mappings additionally enable the vulnerability to be further classified into clear security flaws like authentication, injection attacks, and data exposure.

As this study is specifically focused on API vulnerabilities in financial organizations, automated screening was used for extracting only those vulnerabilities related to APIs. The same was achieved through R programming, which facilitates structured processing along with keyword screening. The procedure of extraction had the following chief steps:

- **Download & Parse JSON Data** – The CVE-Recent database was imported to R for analysis.
- **Keyword-Based Filtering** – API-specific keywords like "API," "authentication," "OAuth," "token," "JWT," "rate limiting," "CORS," and "server-side request forgery (SSRF)" were employed to filter API-specific vulnerabilities.
- **Data Cleaning & Preprocessing** – Retrieved records were processed to maintain data quality, integrity, and usability for statistical analysis.
- **Ethical Implications** – Utilizing open-source vulnerability information promotes transparency and replicability, and obliterates privacy issues surrounding proprietary financial information.

By adopting a methodical and automated strategy toward data gathering and filtering, this research constructs a robust empirical basis for examining trends, patterns, and prominent risk factors in API security for financial institutions.

3.2. Data Analysis Using R Programming

A structured and systematic approach to data analytics is needed in order to gain meaningful understandings of the outcomes from the National Vulnerability Database (NVD) data set. This study uses R Programming to conduct data preprocessing in order to gain statistical significance of the breaches of API vulnerabilities affecting financial institution. The first step into data analysis entails preprocessing of the NVD JSON files extracted data to achieve accuracy and consistency, including data that is raw and contains nested structures and unstructured nested text data. This data is then parsed and transformed into a structured tabular format where the dataset is cleansed. Data is however converted by dealing with date portions into a standardizable format, missing values are handled and API related vulnerabilities are filtered and identified using keyword-based filtering on the description for fields when API related vulnerabilities are identified. Where missing CVSS scores are present zero values are replaced as critical errors are to be prevented and API related vulnerabilities are identified this pre-processing step ensures that only incidents relating to authentication flaws, token mismanagement, injection vulnerabilities, or data exposure remain for further analysis.

Following preprocessing the dataset undergoes descriptive statistical analysis to identify major trends, patterns and severity distributions further both the frequency of vulnerabilities across time by aggregating CVE records in the descriptor per year and insight into whether or not API related security issues with API related vulnerabilities in financial institutions have been enlarged over time. Both the severity distributions of the vulnerabilities are looked at using CVSS Score IV scores categorizing into LOW, MEDIUM, HIGH, and CRITICAL levels. And lastly Group vulnerabilities by CWE classification to identify recurring patterns in security weaknesses. This is generally used to determine the most prevalent attack vectors such as improper authentication where insufficient access controls and injection-based exploits also occur. To provide insight into more detailed information about how API security vulnerabilities are trending, visualization techniques such as line graphs, bar charts and density plots are employed to show how security flaws in a API, can be depicted in order to discern trends; it is also possible for ease of analysis, that safe representation of attack vectors and exploitability scores are provided to show the impact that the various security flaws being depicted has on the API resilience in a finance system. Insights into the API attack characteristics such as distribution of attack complexity, whether it be of the LOW or HIGH variety compared with vulnerabilities of NONE (LOW) or HIGH (HIGH) privilege exploitation that involve a dependency upon user input are determined by using grouped statistical summaries and Kruskal Wallis comparisons in order to check if significant differences exist between the different affected vulnerability categories. In addition to statistical hypothesis testing and effect size analysis, assessments of differences in severity of vulnerability across different types of APIs (Internal APIs, Web APIs, Local APIs and other APIs) are undertaken. Non-parametric tests of the Kruskal - Wallis test and Dunn's post hoc comparisons are employed in order for specific API categories to determine whether specific API categories possess significantly different risk profiles. Cliff’s Delta effect size measurement is then employed to quantify the magnitude of differences between the API categories. In the instance of applying rigorous statistical analysis and visualization methods this research provides a data driven assessment of API security risks within financial institutions. Use of R programming is increasingly efficient which is getting positively valued to perform data handling, transformation and interpretation in a meaningful way which in turn allows financial cybersecurity practitioners to identify actionable insights with regards to increasing the resilience of APIs in financial institutions. The outcomes from this analysis will form the basis of a proactive framework that is going to be applied by financial institutions to mitigate the impacts of API vulnerabilities by acting in an effective measure.

3.3. Ethical Considerations

Ensuring that ethical integrity and transparency is fundamental to the research conducted in this study in that the study focusses on the analysis of cybersecurity vulnerabilities in financial institutions. The research adheres to strict ethical guidelines which includes but not limited to reliance on publicly available data; the use of the national vulnerability database (NVD) feeds explicitly from the National Vulnerability Database. By using open-source datasets as opposed

to proprietary or confidential financial data, the research seeks to avoid privacy concerns and ethical issues pertaining to handling sensitive information. In financial institutions there are too often restrictions on access to internal cybersecurity data due to confidentiality agreements which exists alongside a range of regulatory requirements.

This will also allow other researchers as well as cybersecurity professionals to perform a replication of the findings and to validate the results as well as be able to extend the analysis to different datasets or industries. The use of open-source tools and statistical models ensures that the research process remains accessible, verifiable and adaptable. An additional crucial ethical principle that is followed by the research work is that of responsible reporting of cybersecurity vulnerabilities. The study investigates patterns and trends in API security risks but does not give away specific vulnerabilities that could be exploited for malicious purposes. This study has a primary focus to understand security trends that may show weaknesses of particular financial institutions or API implementations in question but rather to understand broader trends in security. Another major ethical concern is the responsible reporting of public vulnerability databases. While the study explores the patterns in API security risks it does not identify any specific vulnerabilities which could be exploited for malicious purposes. The major focus of the study is not to identify security weaknesses in particular financial institutions or implementations of API, but rather study the general security trends that may not be exposed by vulnerabilities in specific financial institutions or API implementations. Furthermore, by adhering to ethical research principles the study ensures that its findings are valuable, responsible and transparent and to be beneficial to the cybersecurity community.

4. Results and Analysis

Scalability and the level of cybersecurity resilience in financial institutions should be understood through a close examination of API vulnerabilities, their patterns and the effect that they can have on security as a whole. In this particular section the findings are gathered from the technical data of the National Vulnerability Database in feeds collected in R programming. The analysis which is carried out is made through statistical and visualization methods in R and a breakdown of vulnerability related information is included. The results provided offer a series of findings on the frequency of API related vulnerabilities, how severe they are, and the key influencing factors on cybersecurity resilience for financial institutions. The analysis opens up by looking at general trends in vulnerabilities over time to see if the number of reported incidences is increasing or decreasing. A further classification is carried out categorising vulnerabilities based on their severity level (LOW, MEDIUM, HIGH, CRITICAL) which in this case uses the CVSS system for scoring of which allows the resultant virulence of an attacker to themselves the financial institution in question. The attempt to look at a series of classifications will add further details into the most common types of API security flaws, such as improper authentication, data exposure and injection attacks. Beyond these classifications, this section will evaluate attack characteristics to see the attack vectors, attack complexity, required privileges and user interaction. All these are factors that are crucial in determining whether an API vulnerability can be exploited with ease or if what the potential impact if exploited. Scalability and the level of cybersecurity resilience in financial institutions should be understood through a close examination of API vulnerabilities, their patterns and the effect that they can have on security as a whole. In this particular section the findings are gathered from the technical data of the National Vulnerability Database in feeds collected in R programming. The analysis which is carried out is made through statistical and visualization methods in R and a breakdown of vulnerability related information is included. The results provided offer a series of findings on the frequency of API related vulnerabilities, how severe they are, and the key influencing factors on cybersecurity resilience for financial institutions. The analysis opens up by looking at general trends in vulnerabilities over time to see if the number of reported incidences is increasing or decreasing. A further classification is carried out categorising vulnerabilities based on their severity level (LOW, MEDIUM, HIGH, CRITICAL) which in this case uses the CVSS system for scoring of which allows the resultant virulence of an attacker to themselves the financial institution in question. The attempt to look at a series of classifications will add further details into the most common types of API security flaws, such as improper authentication, data exposure and injection attacks. Beyond these classifications, this section will evaluate attack characteristics to see the attack vectors, attack complexity, required privileges and user interaction. All these are factors that are crucial in determining whether an API vulnerability can be exploited with ease or if what the potential impact if exploited. In addition, vendor analysis provides a way of identifying which technology providers are the most prone to API vulnerabilities, and also provides a firm insight into which platforms or software packages require a more robust level of security. The API security risks

will be examined via a statistical test and comparative analysis. The research incorporates several approaches such as the Kruskal-Wallis test and Dunn's post hoc test to determine whether certain API categories, such as web APIs and internal APIs show statistically significant differences with respect to their security risk levels. In addition to this effect size measurements are used to quantify the magnitude of difference between these API categories. By structuring data-driven, systematic analysis for this section critical analyses of cybersecurity resilience in financial institutions will provide the foundations for a proposed framework of cybersecurity resilience that will allow financial institutions to adopt data-driven strategies that will make APIs safe and to mitigate risks efficiently.

4.1. Analysis of Cybersecurity Vulnerabilities

The analysis on the documented vulnerabilities of API services yields significant insight into the cybersecurity risks that are faced by the financial institutions. By using the National Vulnerability Feed data from the National Vulnerability Database (NVD) this study identifies quick trends in the severity levels of the threats, attack vectors and common weaknesses that are generated by the APIs security. With the increasing development of cloud services, third party integrations, and the use of Open Banking it is becoming ever more appealing to cybercriminals to exploit APIs new vulnerabilities. The output of this section is derived by extensive data processing, statistical analysis and visualisation using R programming language. Results of an analysis show an increase in the frequency of APIs vulnerabilities that are being reported, an increase in the severity of the threats that the vulnerabilities pose and the factors ascribing to an increase in risk levels that are posed by today’s cybersecurity threats. The first major aspect of the analysis of the dataset is the steady increase in API vulnerabilities present over time. Annual increases in reported security flaws have been confirmed from the data shown below and over the period of the dataset it is believed that the number of financial services based around APIs have increased while at the same time these also appear to be more exposed by the increasing number of vulnerabilities and security flaws during the year. As seen in Figure 1 and the number of reported vulnerabilities also show an increase over recent years which when we look at the observed growth, we can see that in recent years as financial institutions begin to adopt more API based services in order to their services then they are also creating risk with regards to the security of their services. order to their services then they are also creating risk with regards to the security of their services. The increasing trend in the graph also provides evidence of the need for continuous assessments of their security flaws and proactive mitigation strategies.

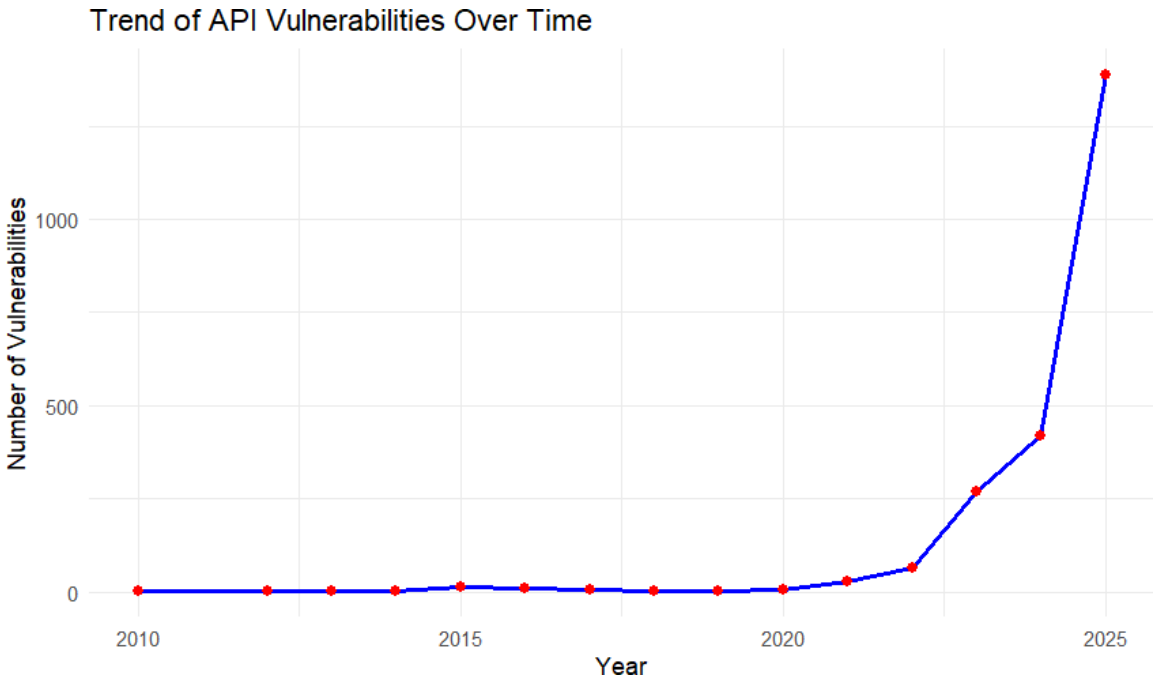


Figure 7: Trend of API Vulnerabilities Over Time

By examining the severity levels of the vulnerabilities that can be found within the API one can get much deeper insight into their potential impact. The Common Vulnerability Scoring System (CVSSv3) classifies the vulnerabilities into four categories of LOW, MEDIUM, HIGH and CRITICAL. The figures provided in Figure 8 show that a large proportion of API vulnerabilities falls into the HIGH and CRITICAL severity ranges, as depicted in Figure 8. The existence and injection of vulnerabilities with high severity suggests that many financial APIs are vulnerable to exploitation which could result in data breaches, unauthorized access and financial fraud. Given that there is a high proportion of severe vulnerabilities this highlights the urgency for organizations to implement a range of security methods such as multi - layered authentication, rate limiting and API monitoring to mitigate off possible threats.

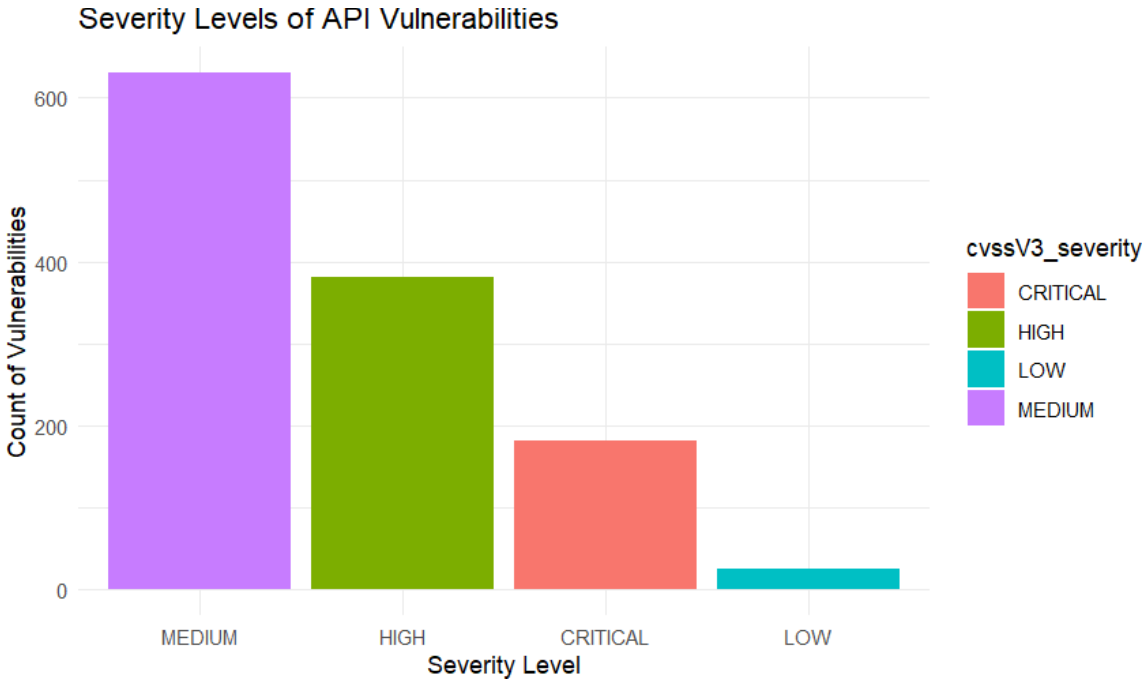


Figure 8: Severity Levels of API Vulnerabilities

A long examination of the Common Weakness Enumeration (CWE) classifications shows most of the recurring security flaws that impact financial APIs. The results of this examination show that Cross-Site Scripting (CWE- 79) and SQL Injection (CWE- 89) are the foremost vulnerabilities, this correlation is followed in order by Out-of-Bounds Write (CWE- 787) and Missing Authorization (CWE- 862 in financial API security. These results are shown in Figure 3 and show that through poor input validation, improper 4 authentication mechanisms and weak access controls are the principal causes of many vulnerabilities found attackers that impact API. Continued presence of these flaws suggest that many financial institutions are either not using secure coding practices or fail to develop adequate API security testing. Addressing these issues requires SDLC, penetration testing, and strict API security policies.

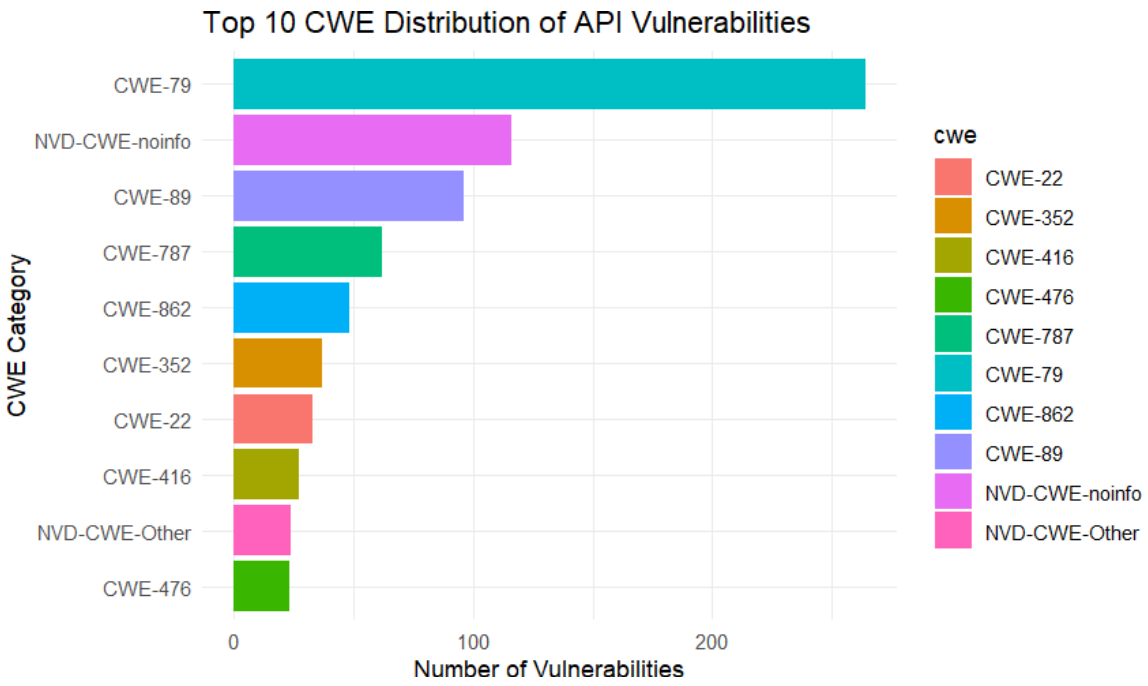


Figure 9: Top 10 CWE Distribution of API Vulnerabilities

By analyzing the attack vectors for API vulnerabilities, it further illustrates the various means by which an attacker can exploit security flaws. The data shows that NETWORK-based attacks are the most prevalent, as shown in Figure 10. The finding implies that remotely exploitable API vulnerabilities pose the greatest risk to financial institutions, as attackers can take advantage of such vulnerabilities without the need to have any access to internal systems. Infrequently occurring local and adjacent network-based vulnerabilities signify that remote exploitation will be a primary concern. Since the inception of API development, it has created an avenue for secure authentication, encrypted API communications, and filtering with a firewall.

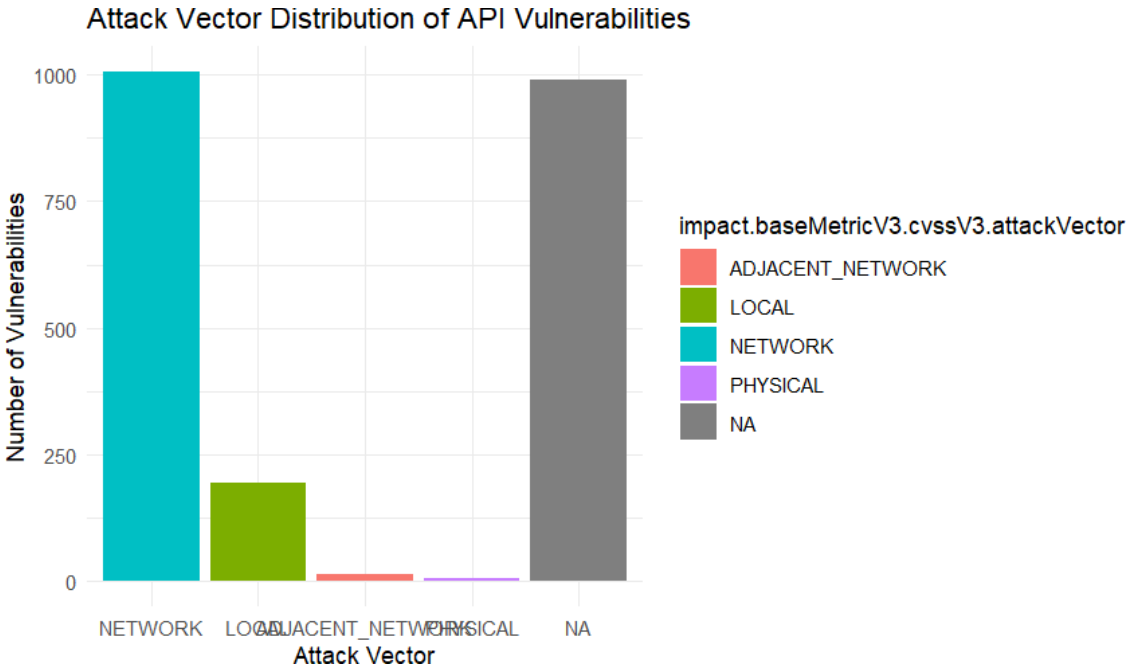


Figure 10: Attack Vector Distribution of API Vulnerabilities

Attack complexity, alongside attack vectors, is important in determining the ease with which it is possible to exploit a vulnerability. Analysing it based on the levels of attack complexity shows that most API vulnerabilities fall into the LOW complexity score range, meaning that exploiting them does not require complex techniques. This is depicted in Figure 5, indicating that several vulnerabilities can be exploited by attackers with only a small amount of technical knowledge. An ever-increasing number of such vulnerable spots compiles a burgeoning risk of automated attacks which, in turn, augments the risks associated with large-scale cyber threats for financial APIs.

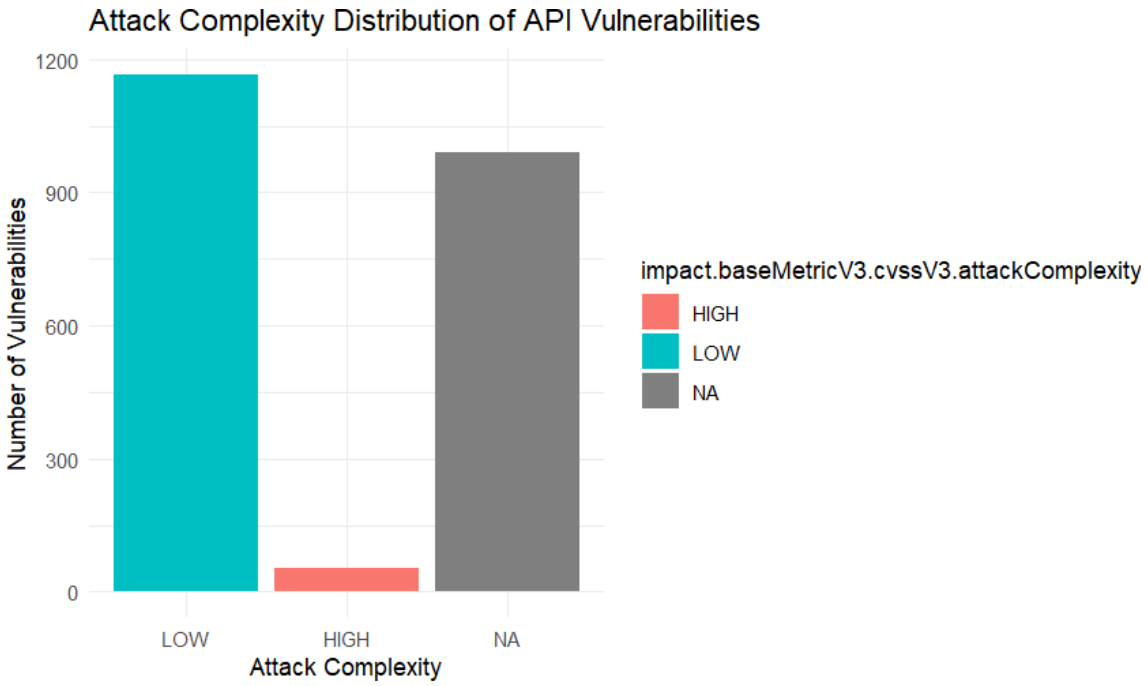


Figure 11: Attack Complexity Distribution of API Vulnerabilities

The subsequent analysis of the privileges needed for exploitation notes that many API vulnerabilities do not require authentication or special privileges. Attackers can exploit some of those weaknesses without needing any type of access credentials, which makes it highly dangerous. There are also API vulnerabilities that do not need any form of user interaction, meaning they can be done even without the victim having to do anything. These findings include what Figure 12 describes about role-based access control (RBAC), strict authentication mechanisms, and security gateways reducing the risk of unauthorized access.

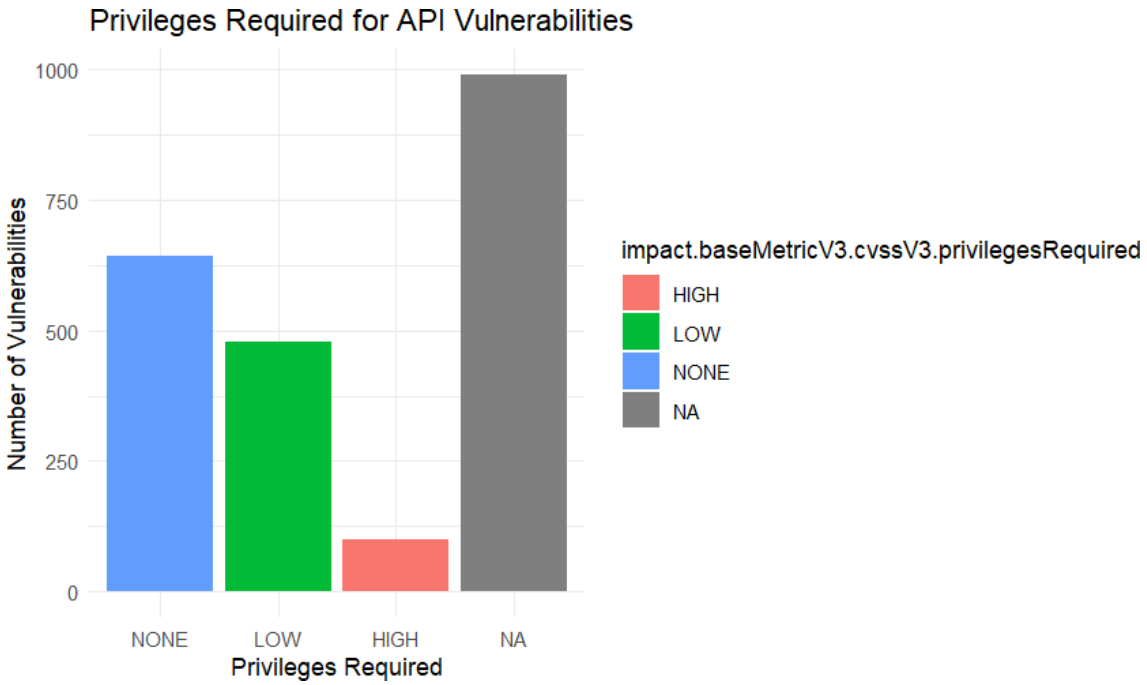


Figure 12: Privileges Required for API Vulnerabilities

Vendor-level analyses of API vulnerabilities offer further insight into the risks that financial institutions face from their third-party vendors. The findings highlight that API vulnerabilities are not confined to one vendor but rather spread across many technology providers. Shown in Figure 13 are several security issues with APIs reported by vendors such as IBM, Google, and Red Hat, which emphasize the importance of vendor security assessments and regular updates for software and hardware. Since financial institutions use third-party API integrations, they must carefully track vendor security advisories and apply patches in a timely manner to better mitigate exposure to cybersecurity threats.

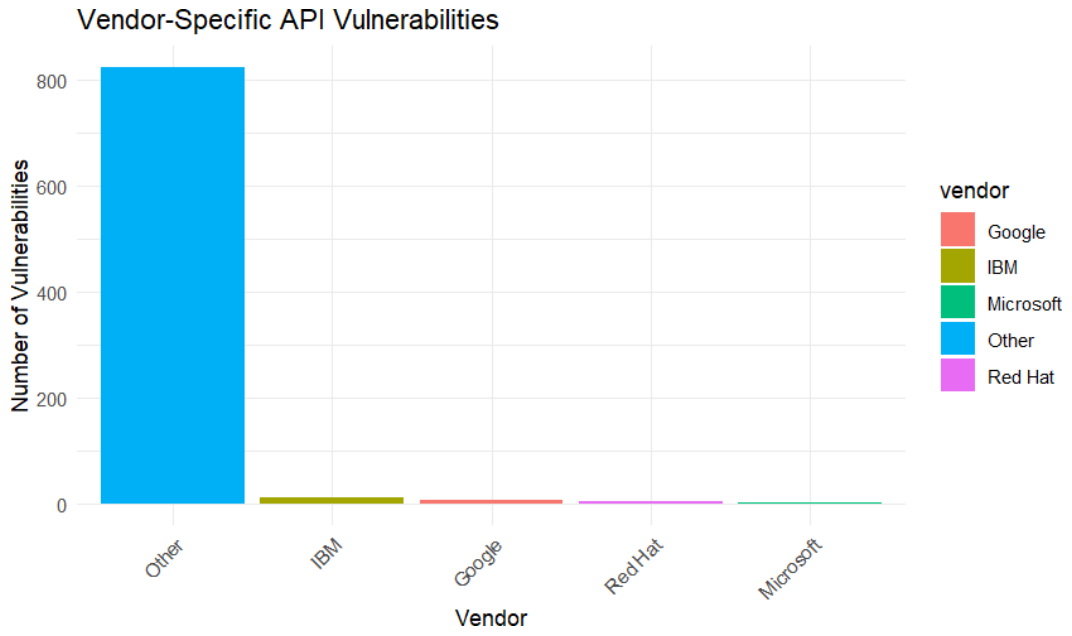


Figure 13: Vendor-Specific API Vulnerabilities

The examination was concluded with the discovery of the urgent need for API security enhancements in finance. Because of rising vulnerability count, high severity, and easy exploitability, API security should take precedence in the cybersecurity strategy of financial institutions. To achieve this goal, businesses should approach API security proactively by undertaking the following measures:

- Routine automated vulnerability assessments
- The setup of a strong authentication and access control system
- Encryption of API communication to avert data leakage
- Proactive continuous monitoring and anomaly detection

The information in this section serves as a platform for further discussion that investigates the factors affecting the cybersecurity resilience of financial companies discussed next. The trends, severity levels, and exploitability of API vulnerabilities give important information about an urgent need for a data-driven, proactive security framework. Elevating API security shall be, therefore, a substantial advancement in strengthening the overall cybersecurity resilience of financial institutions.

4.2. Key Factors Affecting Resilience

Factors like severity of vulnerabilities, conditions for exploitability, attack vectors, and vendor roles in mitigation essentially define the financial institutions' resilience against API security threats. The previous analysis emphasizes key trends concerning API vulnerabilities, and this feeds into organizational cybersecurity preparedness and its risk mitigation strategies.

API vulnerability severity is one of the major factors influencing resilience. A large number of high and critical vulnerabilities in the dataset indicates that these severe threats must be prioritized for mitigation by financial institutions. High frequencies of CWE-79 (Cross-Site Scripting) and CWE-89 (SQL Injection) reinforce the criticality of secure coding practices, periodic vulnerability assessments, and fortification of authentication mechanisms to minimize risks.

Another factor that affects resilience against cyberattacks is the exploitability of such vulnerabilities based on their respective attack vectors. Many of the analysed vulnerabilities arise from networked attack vectors, bringing to focus very widespread risks due to remotely exploitable API weaknesses. According to the result, APIs that are exposed via the internet (like Web APIs) are more vulnerable to unauthorized access, data leaks, and injection attacks. This calls for reinforced access controls, technical security checkpoints through API gateways, and regular penetration testing to proactively identify security gaps.

Privileges needed to launch an attack are another significant determinant of resilience. A number of vulnerabilities in the dataset used a low or non-existent privilege level, suggesting that attackers can undermine API security without needing administrative access to an API. This implies that financial institutions need to reinforce their combinations of authentication mechanisms, ensure their policies concerning the principle of least privilege (PoLP), and ensure that sensitive API operations maintain requisite multi-factor authentication (MFA).

Some scenarios of security attacks rely on user interaction and further introduce a dimension to consideration on resilience. For instance, the types of software vulnerabilities that rely on user actions, such as phishing API exploits, suggest that these organizations invest in cybersecurity awareness, enhancing phishing detection systems and introducing behavioral anomaly detection systems for consideration of social engineering attacks with less risk.

In a broader perspective, the role of vendors as well as third-party software is another fundamental aspect of resilience. The analysis pointed out several major vendors associated with API vulnerability incidents (for instance, IBM, Google, Red Hat) and indicated that financial institutions should verify the security metrics for their technology providers. High-level vendor security assessment expectations, regulatory compliance toward cybersecurity, and urged timely patch management from those vendors will certainly bolster API security posture.

4.3. Patterns and Trends Identified from Data

Presenting a data-driven analysis of vulnerabilities in APIs offers an insightful take on the cybersecurity world that is evolving in a financial organization space. Through an examination of patterns over time, common weaknesses, and statistical distributions, it points to the trends critical in allowing organizations to build more effective security strategies. While the other sections have sufficiently established the severity and exploitability of vulnerabilities, here we discuss how these vulnerabilities have arisen, which attack methods remain ubiquitous, and how organizations in financial institutions can prepare for forthcoming dangers.

Among the most cases observed through the analysis was a continual increase in API vulnerabilities across time. The data showed a generally upwards trajectory of API vulnerabilities reported on a yearly basis, indicating that with changing trends in user-adopted APIs against financial services also comes increasing sophistication amongst cyberattacks against the same. In tandem with enterprise digital transformations across the board-be it through open banking, mobile banking, or cloud-based financial services-APIs are at the heart of it. As financial institutions ramp up on their digital services, attack surfaces are actively under pursuit by adversaries in their mission to gain access, keeping the cycle of continuous monitoring and proactive security in check.

Another singular trend is in the counts of all vulnerabilities clustered according to their severity. The high amounts of high and critical vulnerabilities as seen from the data are indicative of persistent weaknesses needing to be fixed proactively before they present threats that risk being exploited. The clustering of CWEs like SQL Injection (CWE-89), Cross-Site Scripting (CWE-79), and Broken Authentication (CWE-287) speaks to a situation where API security hassles still coexist with poor coding syntax and misconfigurations. This continues to cast doubt on the effectiveness of developed security tools since, after all this happening at all, the misconfiguration of authentication and improper input validation still remain pernicious risk factors.

The attack vectors applied in the exploitation of APIs underline major security risks that should be paid attention to. Most of the analyzed vulnerabilities are amenable to remote exploitation (i.e., network-based attacks), with some others requiring physical or local access to affect the systems. This helps to agree with the said threat scenario of external attacks including API endpoint scanning, unauthorized data extraction as well as injection-based exploits. Most of the time, the exploited API vulnerabilities are user-interaction independent, but when they depend on user interaction, most often, they depend on social engineering techniques such as phishing or malicious link injections. This indicates that strong fortifications coupled with user training need to be employed by the financial institutions in an attempt to combat the risk of exploit vectors.

Additionally important to note is the involvement of vendors in API security incidents. Several large technology providers including IBM, Google, and Red Hat are among those vendors connected to known API vulnerabilities. This puts into better focus the fact that financial institutions need to go beyond their internal security assessments and scrutinize their third

party vendors with the utmost rigor. With increasing reliance on third-party software and cloud services, vendors form an essential component in the overall stability of cybersecurity initiatives. Beyond these broad patterns, it can be observed from the statistical analyses that certain categories of API—for example Web APIs—are more prone to suffer higher severity vulnerabilities than others, such as internal or local APIs. Under the Kruskal-Wallis statistical test that examined the variations in CVSS severity scores on various API types, some variance between these types was noted, although not on every occasion reaching statistical significance. It is however compelling that the pairwise results would find out that the Web APIs recorded higher CVSS scores and are more often under attack than any other API categories. This reinforces the need for enhanced security controls on any externally-facing API, especially on that concerned in authentication, payments, and sensitive data transfer.

These trends and patterns emerging from the dataset show the risk that the security APIs present in the financial institutions as being on the increase, interdependent, and requiring a data-driven multi-layered defense approach. Utilizing insights from previous vulnerability data, severity distributions, attack vectors, vendor-specific risks, and statistical analyses, financial institutions would additionally foresee emerging threats in this pro-active approach to API security governance.

4.4. Insights Related to API Vulnerabilities

The analysis of API vulnerabilities in financial institutions provides valuable lessons on how, why, and where these security flaws originated. By utilizing historical data from the National Vulnerability Database (NVD), key takeaways should inform risk mitigation, cybersecurity resilience planning, and proactive defenses. This section consolidates the understandings gleaned from the previous analyses and brings attention to the most pressing security concerns besetting financial institutions.

The most significant insight gleaned from the data is that the weaknesses constantly fall under the same domain of insecurity. With the improvement of cybersecurity awareness and regulatory compliance, problems like SQL Injection (CWE-89), Cross-Site Scripting (CWE-79), and Broken Authentication (CWE-287) keep recurring. This constant occurrence postulates that some basic hygiene problems in terms of security-such as improper input validation, weak authentication mechanisms, and insufficient access controls-continue to pose major concerns. Since these vulnerabilities lead to unauthorized access, data breach, and financial fraud, it is essential for financial institutions to establish secure coding practices, perform penetration testing on a perennial basis, and conduct security audits on API security.

Another critical point is the disparity in API security from different vendors and technology providers. The dataset also indicates that some of the vendors with vulnerabilities in APIs are IBM, Google, and Red Hat. This growing concern about third-party risks involves the fact that many financial institutions rely on external software providers, cloud-based services, and API integrations to advance their offerings in digital banking and fintech. The added dependencies

increase external risks from integrated partners, thus making supply chain security a non-negotiable element of API security strategies. Financial institutions should apply vendor risk assessments, monitor their APIs for dependencies, and enforce compliance with cybersecurity best practices to mitigate risks originating with third-party providers.

Further statistical analysis of API categories and severity levels gives more information in this respect. Web APIs tend to have greater CVSS severity ranking than internal or local APIs, which are widely used for internet banking, mobile payments and financial data exchanges. This implies that public APIs are inherently more susceptible to attack because they are more accessible and fed into supporting services. Given this trend, financial institutions will have to adopt stronger authentication checks, rate limiting, and state-of-the-art encryption protocols for all their externally facing APIs.

This analysis evidently demonstrates that there is not just an increase in the number of financial APIs but also a serious increase in their level of sophistication, calling for a proactive rather than reactive security posture by the institutions. Traditional security frameworks based on firewalls and perimeter security are unhelpful to securing APIs better against specific threats Rather, financial institutions need to now introduce continuous monitoring, behavioral analytics, and AI-based anomalous detection to ensure timely detection and annihilation of the threats before this turn systematic.

Another insight is the regulation and compliance impacts. Many of the identified vulnerabilities found in APIs may truly harm the financial institutions regarding legal culpability, regulatory fines, and reputational damage. Worldwide, financial regulators and authorities, such as the European Banking Authority (EBA), Financial Stability Board (FSB), and US Office of the Comptroller of the Currency (OCC), now tend more to emphasize the API security governance as part and parcel of the financial cybersecurity framework. These include the General Data Protection Regulation (GDPR), Payment Services Directive (PSD2), and Open Banking Standards, and financial institutions must ensure that their API security practices adhere to compliance obligations.

5. Proposed Framework for Cybersecurity Resilience

The growing reliance on APIs by financial institutions has significantly improved digital banking services, real-time transactions, and seamless customer experiences. However, the widespread use of APIs has also widened the attackers' surface, exposing financial institutions to data breaches, fraud, and cyber threats. The report draws out research findings from such critical security loopholes as weak authentication mechanisms, unnecessary access controls, and external attack vectors, which call for a structured and data-driven approach towards cybersecurity resilience.

As an administrative response to these concerns, this section proposes a full-on framework intended to solidify API security and boost the cybersecurity resilience of financial institutions. The proposed framework bases its recommendations on empirical insights derived from National Vulnerability Database (NVD) analysis that aligns security measures with trending real- world API vulnerability trends. The primary aim is to achieve a proactive and adaptive cybersafety framework that could enable a financial institution to cope with risk, elevate API protection message, and, thus, comply with given industry standards.

Importantly, the critical element of the framework relies on data. Unlike conventional cybersecurity approaches that resort to barriers based on firewalls and old network-based defenses, the new model emphasizes constant monitoring, real-time threat intelligence, and API-specific security controls. By integrating other modern technologies like automated security testing, anomaly detection, and risk-based access management, financial institutions can in advance be ready to tackle emerging threats targeting technical financial services that should retain integrity, confidentiality, and availability.

It also takes into account security-risk-based controls aimed at different sets of API categories: internal APIs, external APIs, and third-party integrations. With publicly exposed Web APIs being more prone to attacks than internal APIs, a tiered model introduces context-sensitive security to that extent that more-litigious APIs are subjected to a more stringent security assessment, access control circumvention, and thus encryption.

Additionally, the proposed framework will tackle issues regarding regulatory compliance in-line with other global standards on financial security like PSD2, GDPR, and the NIST Cybersecurity Framework. Incorporating these security policies, which are compliance-driven, into their API security strategies will minimize financial institutions' exposure to the legal risk which regulators tend to guard against and the penalties that are imposed. Protecting these measures will maintain the loyalty of their customers.

This final outline will offer a flexible, scalable, and adaptive framework aimed at allowing financial institutions to address increasingly sophisticated cyber threats without sacrificing performance, innovation, and user experience. The following subsections will highlight the major pieces of this framework, consisting of tailored cybersecurity strategies; implementing tactics; and real applications in strengthening financial resiliency against cyber threats.

5.1. Development of New Methods or Solutions

Creating a strong cybersecurity framework for financial institutions, which ensures security, requires data-driven insights, proactive threat detection, and adaptive security mechanisms. Based on findings from the analysis of the application programming interface (API) vulnerabilities laid out in the National Vulnerability Database (NVD), this section proposes a systematic methodology for enhancing cybersecurity resilience.

A multi-layered framework incorporating continuous risk assessment, API security measures, and automated vulnerability management is proposed in the solution. With the model implemented, financial institutions will be able to limit exposure to high-risk vulnerabilities, ensure strong API authentication mechanisms, and prevent unauthorized access to sensitive financial data.

Key Components of the Proposed Solution

Risk-Based API Security Model

APIs in financial institutions are used for different purposes and with varying levels of exposure. In this regard, in order to allow for efficient risk management, the proposed framework segments APIs into three security tiers:

API Category	Risk Level	Security Measures
Internal APIs	Low Risk	Role-based access control (RBAC), authentication tokens, logging mechanisms
Partner APIs	Medium Risk	API gateway security, rate limiting, mutual TLS authentication
Public APIs	High Risk	Strong encryption, OAuth 2.0, anomaly detection, API firewalls

Table 2: Risk-based API category and security measures

This implementation allows for heightened security measures when higher-risk APIs are concerned, thereby reducing the possibility of cyber-attacks.

There is wide recognition that one of the greatest challenges in API security is to detect those vulnerabilities before an exploit occurs. That being said, the proposed solution implements real-time vulnerability intelligence sourced from NVD to produce an automated risk scoring mechanism.

Process Workflow:

Step 1: Extract API-related vulnerabilities from NVD feeds.
Step 2: Assign risk scores according to CVSS severity levels.
Step 3: Map each API vulnerability to its Common Weakness Enumeration (CWE) category.
Step 4: Alert security teams about high-risk vulnerabilities and recommend patches.

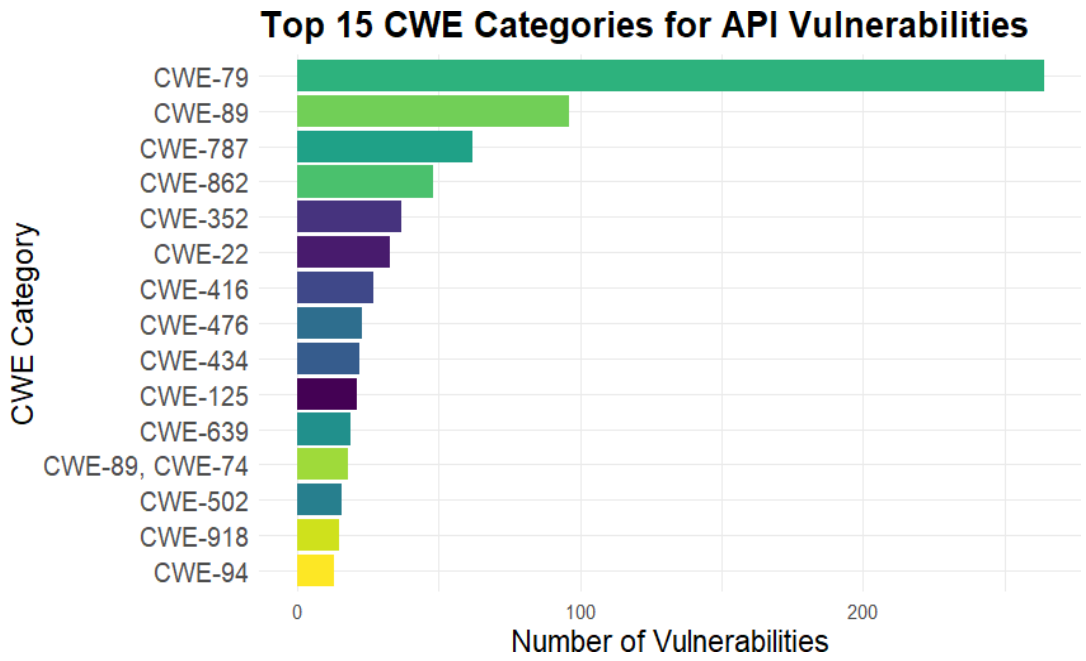


Figure 14: Top 15 CWE Categories for API Vulnerabilities

This visual illustration provides a perspective based on the data indicating which types of vulnerabilities are most prevalent in financial APIs. This aids institutions in setting their security fixing priorities.

Implementing Stronger Authentication & Authorization Controls

Another important cause for API breaches in financial institutions is inadequate authentication mechanisms. The proposed framework proposes adding to it the following:

- ✓ OAuth 2.0 and OpenID Connect (OIDC) - Secure token-based authentication.
- ✓ Mutual TLS (MTLS) - Programming Communication with Encryption.

- ✓ Fine-grained Access Control - Ensuring least privilege access for API consumers.

Why This Is Important

Our analysis reveals that many security weaknesses come from set up login systems and weak access rules in public APIs. Using strong login methods greatly cuts down the risk of API abuse and data breaches.

Continuous Security Monitoring & Incident Response

To fight off new cyber threats, banks and other money-related businesses need to keep a close eye on their systems. This watching should be part of their plan to bounce back from cyber-attacks.

Key Ways to Watch:

- ✓ **Check API Logs in Real Time:** This spots odd API calls or tries to get in without permission.
- ✓ **Anomaly Detection using Machine Learning:** This marks possible API misuse by looking at how traffic flows.
- ✓ **Fix Problems Automatically:** This uses NVD feed updates to fix known weak spots before they cause trouble.

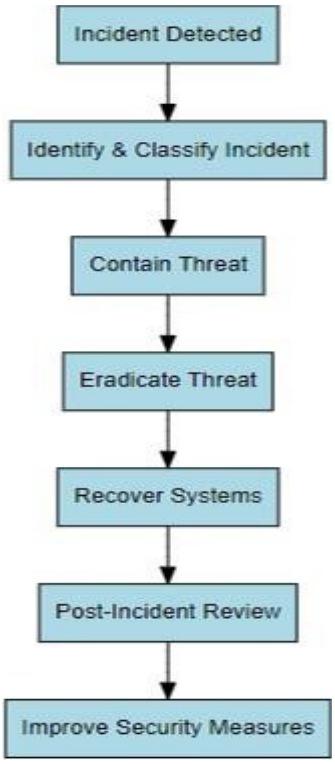


Figure 15: Incident Response Workflow

This plan shows how banks and such can find check out, and deal with API security threats. It helps them react faster and lose less money when something goes wrong.

Secure API Development Lifecycle (API-SDLC)

Banks and other financial companies need to weave security measures into every step of creating APIs. The framework suggests:

Security Measures at Different API-SDLC Stages

Design Phase Threat modeling, API security architecture review
Development Phase Secure coding standards (OWASP API Top 10 compliance)
Testing Phase Automated security testing, penetration testing
Deployment Phase API gateway security, token expiration policies
Monitoring Phase Continuous API security scanning, anomaly detection

Figure 16: Security Measures at Different API-SDLC Stages

This makes sure API weak spots are dealt with at each point in the process instead of trying to fix them after launch when the chances of attacks are greater.

Expected Impact of the Proposed Solution

When banks and other money-related companies put this new cyber safety plan into action, they can look forward to:

- Fewer API Break-ins: Tougher checks on who can use APIs make it harder for bad guys to get in where they shouldn't.
- Better Spotting of Dangers: Using NVD info to check for weak spots helps to handle risks before they become big problems.
- Quicker Reactions to Problems: Watching everything as it happens lets security teams stop threats before they get out of hand.
- Following Safety Rules: Making sure API safety matches what NIST, GDPR, and PSD2 say keeps the company on the right side of the law.

5.2. Strategies for Enhancing API Security

As financial institutions depend more on APIs for digital transactions, the necessity to establish strong API security measures has emerged as a critical concern. The potential threats of data breaches combined with unauthorized access and financial fraud demand the implementation of proactive security strategies. This section examines essential approaches to bolster API security by emphasizing advanced authentication methods alongside rate limiting techniques and the incorporation of machine learning with NVD data for real-time threat detection.

Strengthening API Authentication and Access Control

The exploitation of APIs frequently occurs through attack vectors that stem from inadequately designed authentication processes combined with deficient access control systems. Financial institutions need to deploy complex authentication mechanisms alongside stringent authorization protocols to safeguard APIs against unauthorized access.
Key security measures include:

1. OAuth 2.0 with OpenID Connect: This system uses tokens to ensure only authorised users can access financial APIs securely.
2. Mutual TLS (Mttls): It checks both the lient and server identities before allowing access, keeping communication encrypted and private.
3. JSON Web Tokens (JWT) with Short Expiration: these tokens are temporary and have strict expiration time, which boosts session security.
4. Role-Based and Attribute-Based Access Control (RBAC & ABAC); Access to APIs is controlled based on user roles and specific attributes such as location, device type, and risk level.

Financial institutions can protect transactions and block unauthorized access by implementing strong security measures.

Implementing Rate Limiting and API Throttling

When API requests lack proper control then denial-of-service (DoS) attacks occur alongside abuse from malicious bots and unintended disruptions of service. API traffic regulation happens through rate limiting and throttling systems which protect against system overload caused by excessive requests.
Best practices include:

- Users and applications face rate limitations using Token Bucket for their API interactions during time-based intervals.

- The platform enforces both IP-Based and User-Based throttling which manage how many API requests each IP address and each authenticated user can execute.
- Geo-Blocking in combination with Risk-Based API Access control blocks users from accessing the API according to their geographical region and unusual request patterns for detecting anomalies.
- The system uses AI to adapt request-limit settings according to user conduct and server utilization levels and time-dependent actions.

Certain financial institutions utilize stringent API rate limits in order to defend APIs from automated attack attempts also achieving system stability alongside equal resource utilization.

Integrating Machine Learning with NVD Data for Real-Time Threat Detection

Current API security methods use detection rules that struggle to detect new security threats together with unreported system vulnerabilities. The combination of machine learning (ML) with NVD data enables financial institutions to discover abnormal API behavior while forecasting attack scenarios which allows them to take ahead-of-time risk action processes.
Machine learning technologies serve several vital security functions for API protection through the following applications:

- API Traffic Anomaly Detection through ML models analyzes API request patterns to reveal abnormal activities that could signal either malicious intent or fraudulent operations.
- AI-based systems scan NVD feeds through automated tools that detect financial API vulnerabilities at risk levels using CVEs (Common Vulnerabilities and Exposures).
- Existing security event data help ML algorithms to forecast API vulnerabilities before they are subject to exploitation.
- An API user monitoring system conducted by AI-based methods automatically separates regular patterns from abnormal activities and rapidly detects risky transactions along with unauthorized access attempts.

Financial institutions can utilize machine learning to detect threats swiftly while enabling automated security responses and maintain security across evolving cyber threats.

5.3. Practical Applications in the Financial Industry

Financial organizations require Application Programming Interfaces (APIs) to enable digital transactions as well as open banking functions and quick financial services delivery. The continuing growth of APIs has created larger vulnerability areas which financial institutions must implement strong security protocols to defend themselves successfully. Security experts will examine methods financial institutions can implement to use the resilience strategies from this work for enhancing API protection and safeguarding customer information while meeting regulatory standards.

Securing Open Banking APIs

Financial institutions now implement API exposure toward third-party providers (TPPs) for enabling payment services along with financial analytics and account aggregation. The advance in innovation through better customer experience simultaneously creates security risks which allow unauthorized data breaches as well as API abuse between financial institutions and their third-party partners.

Customers need their open banking APIs to be protected by the following practices:

- All systems must utilize Strong Customer Authentication through the combination of multi-factor authentication (MFA) and transaction authorization procedures.
- Financial organizations should utilize API gateways which offer a centralized security system that includes authentication features together with rate limiting capabilities and real-time monitoring functions.
- Every API request requires authentication through a verification system based on device identity combined with behavioral and risk assessment of both device and user.
- Financial organizations should use National Vulnerability Database (NVD) feeds to track down API weaknesses and automatically fix security vulnerabilities through proactive measures.

Open banking transactions become secure through these measures which also maintain compliance with PSD2 (Revised Payment Services Directive) and GDPR (General Data Protection Regulation) regulations.

Enhancing Fraud Detection in API Transactions

The processing of real-time payments along with digital banking operates through APIs as its central functional element. Financial organizations face substantial security threats because of fraudulent activities combined with unauthorized system intrusions along with improper API management practices. Enterprise API security becomes stronger when staff use machine learning anomaly detection algorithms in security operations.

The detection of financial fraud improves through these measures by financial institutions:

- Machine learning models detect irregular API requests because they analyze patterns of usage along with unusual failed authentication attempts and abnormal transaction sizes.
- The implementation of historical fraud data analysis and behavioral analytics lets institutions create API transaction risk scores for real-time prevention strategies.
- Special security systems analyze vulnerabilities reported through the National Vulnerability Database to detect emerging attack patterns which enable them to modify security rules automatically.

- When security triggers run automatically, they can block suspected API keys to stop fraud from occurring.

The identification of fraudulent API interactions in advance enables financial institutions to protect API security while lowering both fraud losses and improving customer trust.

Regulatory Compliance and API Risk Management

Financial institutions which operate within the industry must show evidence of robust API security measures because they must follow strict data protection and cybersecurity regulations. The failure to meet cybersecurity regulations leads to severe monetary penalties together with destructive brand image deterioration alongside business system breakdowns.

The compliance with regulatory requirements needs financial institutions to follow the following steps:

- API security testing should follow Secure API Development Lifecycle (SDLC) Practices by being performed in all stages beginning from design until end deployment.
- Companies should implement three security frameworks namely NIST Cybersecurity Framework along with ISO 27001 and OWASP API Security Top 10 which ensures organizations follow international security standards.
- Organizations must track all API requests together with authentication trials and data movements through complete logging systems to fulfill auditing and compliance demands.
- Every communication through financial APIs should use encryption to shield sensitive payment data from unauthorized interception.

The implementation of rigorous governance together with compliance and monitoring frameworks allows financial institutions to prevent regulatory vulnerabilities while improving their API security resilience.

6. Conclusion

APIs play an essential role in the financial industry because they both open new innovation possibilities and create major cybersecurity threats. Automated Program Interfaces (APIs) make possible easy data transfer between systems while supporting open banking standards and also increase transaction speed. APIs have gained extensive deployment in industry sectors but this extensive usage exposes them to an enlarged attack area which criminal actors' exploit. The research investigates API security vulnerabilities through NVD data while presenting measures that strengthen financial institution cybersecurity resilience.

This research used quantitative methods to identify patterns together with risk factors that impact API security and developed trends from the analysis. API security vulnerabilities continue to increase in both volume and severity according to the findings which show that many weaknesses pose high or critical risks to systems. The major security vulnerabilities organizations experience stem from injection attacks combined with broken authentication and insufficient access controls and system misconfigurations although these issues are responsive to proactive security interventions. The proposed framework developed by this research created specific cybersecurity resilience components for financial institutions which incorporated preventive measures and added detective functions together with response capabilities. The framework underlines critical elements including protected API programming along with data security models as well as real-time threat data and automatic security implementation. A promising fraud detection solution involves the implementation of machine learning models for API transaction pattern analysis to identify both fraudulent activities and abnormal system behaviors as well as new attacks beforehand. Financial institutions can track evolving security threats using NVD feeds which enables them to implement immediate security patch applications.

The main discovery during this research showed financial institutions face challenges when trying to maintain usable security measures that also provide good performance results. The implementation of serious security protocols including multi-factor authentication and encryption and rate limitation results in better protection yet produces both operational complexity and performance delays. Organizations need to use risk-based principles to automatically readjust security measures according to transaction risk levels together with user actions and instant threat information.

The study established regulatory compliance as an essential factor for securing Application Programming Interface security. The security requirements specified by PSD2 and GDPR and NIST cybersecurity guidelines constrain financial institutions to implement robust security solutions that defend both customer information privacy and fight financial fraud. API security policies based on compliance requirements help institutions reduce legal exposure through strengthened customer trust and established institutional reputation.

The produced research contains significant findings yet several constraints need to be taken into consideration. The research worked predominantly with quantitative data collected from public vulnerability sources without interviewing security experts in financial institutions or security practitioners. This study omitted an evaluation of individual security defense systems used by companies despite their potential variations from standard industry security practices. Further studies should combine expertise interviews with concrete case investigations and AI simulation exercises of cybersecurity threats to boost the practical application of their results.

REFERENCES

Adebayo, H. (2025). *Digital Banking and API Security: Best Practices for Secure Financial Transactions*. [online] Available at: https://www.researchgate.net/publication/388589352_Digital_Banking_and_API_Security_Best_Practices_for_Secure_Financial_Transactions.

AI, Z. (2019). *There's A Fix To The Problem Of Biased Algorithms in Lending*. [online] Zest AI. Available at: https://www.zest.ai/learn/blog/theres-a-fix-to-the-problem-of-biased-algorithms-in-lending/.

Alpaca API Docs. (2025). *Welcome*. [online] Available at: https://docs.alpaca.markets/docs/getting-started

Avery, A. (2021). After the disclosure: measuring the short-term and long-term impacts of data breach disclosures on the financial performance of organizations. *Information & Computer Security*, 29(3), pp.500–525. doi: https://doi.org/10.1108/ics-10-2020-0161.

Baran, E. (2023). *Biggest Cyber Threats For Financial Institutions In 2023*. [online] www.blazeinfosec.com. Available at: https://www.blazeinfosec.com/post/cyber-threats-for-finance-2023/.

Binance.com. (2025). *Developer Center*. [online] Available at: https://developers.binance.com/en

Böhme, R. and Schwartz, G. (n.d.). *Modeling Cyber-Insurance: Towards A Unifying Framework WORKING PAPER* *. [online] Available at: https://infosecon.net/workshop/downloads/2010/pdf/Modeling_Cyber-Insurance:_Towards_A_Unifying_Framework.pdf

Brundage, M., Avin, S., Clark, J., Toner, H., Eckersley, P., Garfinkel, B., Dafoe, A., Scharre, P., Zeitzoff, T., Filar, B., Anderson, H., Roff, H., Allen, G.C., Steinhardt, J., Flynn, C., hÉigeartaigh, Seán Ó, Beard, S., Belfield, H., Farquhar, S. and Lyle, C. (2018). *The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation*. [online] arXiv.org. Available at: https://arxiv.org/abs/1802.07228.

Domingues, V. (2018). *Finance and Cybersecurity Risk Management*. [online] Available at: https://www.researchgate.net/publication/344711134_Finance_and_Cybersecurity_Risk_Management.

Carter, W. (2016). *Forces Shaping the Cyber Threat Landscape for Financial Institutions*. [online] Ssrn.com. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3047730.

Cartwright, A., Cartwright, E., MacColl, J., Mott, G., Turner, S., Sullivan, J. and Nurse, J.R.C. (2023). How cyber insurance influences the ransomware payment decision: theory and evidence. *The Geneva Papers on Risk and Insurance - Issues and Practice*. doi: https://doi.org/10.1057/s41288-023-00288-8.

Chell, S., Sainath Chakare, Sohan, P. and S. Sandosh (2024). Real-Time Threat Detection and Mitigation in Web API Development. [online] pp.1–9. doi: https://doi.org/10.1109/iceect61758.2024.10739333.

CISA (2023). *Financial Services Sector | Cybersecurity and Infrastructure Security Agency CISA*. [online] www.cisa.gov. Available at: https://www.cisa.gov/topics/critical-infrastructure-security-and-resilience/critical-infrastructure-sectors/financial-services-sector.

Daffu, P. and Kaur, A. (2016). *Mitigation of DDoS attacks in cloud computing*. [online] IEEE Xplore. doi: https://doi.org/10.1109/WECON.2016.7993478.

Dela Luna, C. (2024). *Cyber Security in Banking: Threats, Solutions & Best Practices*. [online] eSecurity Planet. Available at: https://www.esecurityplanet.com/cloud/cyber-security-in-banking/.

Ebsco.com. (2025). Available at: https://openurl.ebsco.com/EPDB%3Agcd%3A14%3A19896558/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A164891477&crl=c&link_origin=scholar.google.com

Estevez, E. (2019). *Open Banking*. [online] Investopedia. Available at: https://www.investopedia.com/terms/o/open-banking.asp.

Europa.eu. (2024). *ENISA Threat Landscape: Finance Sector | ENISA*. [online] Available at: https://www.enisa.europa.eu/publications/enisa-threat-landscape-finance-sector

Everbridge (2021). *The Impact of Cybersecurity Risks on Financial Services*. [online] Everbridge. Available at: https://www.everbridge.com/blog/the-impact-of-cybersecurity-risks-on-financial-services/

Experian.com. (2020). *Home | Experian Global Developer*. [online] Available at: https://developer.experian.com/

Farrell, H. and Newman, A.L. (2019). Weaponized Interdependence: How Global Economic Networks Shape State Coercion. *International Security*, 44(1), pp.42–79.

Fernandez-Carames, T.M. and Fraga-Lamas, P. (2018). A Review on the Use of Blockchain for the Internet of Things. *IEEE Access*, 6, pp.32979–33001. doi: https://doi.org/10.1109/access.2018.2842685.

Financial Times, (2024). *ECB finds 'shortcomings' in banks' ability to cope with cyber attacks*. [online] @FinancialTimes. Available at: https://www.ft.com/content/1e69ab42-813c-4729-aec9-89fe49853fa0.

FS-ISAC, I. (2020). *Financial Services Information Sharing and Analysis Center*. [online] Fsisac.com. Available at: https://www.fsisac.com/.

Gartner (2023). *Gartner: Fueling the Future of Business*. [online] Gartner. Available at: https://www.gartner.com/en

Gong, Y., Zhu, M., Huo, S., Xiang, Y. and Yu, H. (2024). Enhancing Cybersecurity Resilience in Finance with Deep Learning for Advanced Threat Detection. *arXiv (Cornell University)*. doi: https://doi.org/10.48550/arxiv.2402.09820.

Grassi, P., Fenton, J., Newton, E., Perlner, R., Regenscheid, A., Burr, W., Richer, J., Lefkovitz, N., Danker, J., Choong, Y., Greene, K. and Theofanos, M. (2017). *DRAFT NIST Special Publication 80063B Digital Identity Guidelines Authentication and Lifecycle Management*. [online] Available at: https://csrc.nist.gov/CSRC/media/Publications/sp/800-63/3/draft/documents/sp800-63b-draft.pdf.

Hartman Advisors (2021). *Hartman Executive Advisors*. [online] Hartman Executive Advisors. Available at: https://hartmanadvisors.com/how-data-breaches-impact-financial-industry/?utm_source=chatgpt.com

Kopp, E., Kaffenberger, L. and Wilson, C. (2017). Cyber Risk, Market Failures, and Financial Stability. *IMF Working Papers*, 17(185). doi: https://doi.org/10.5089/9781484313787.001.

Krebs, B. (2019). *Capital One Data Theft Impacts 106M People — Krebs on Security*. [online] Krebsonsecurity.com. Available at: https://krebsonsecurity.com/2019/07/capital-one-data-theft-impacts-106m-people/.

Hernandez-Castro, J., Cartwright, A. and Cartwright, E. (2020). An economic analysis of ransomware and its welfare consequences. *Royal Society Open Science*, 7(3), p.190023. doi: https://doi.org/10.1098/rsos.190023.

Labs, P. (2024). *Financial Services Cybersecurity: 2024 Performance in Banking, Financial Services, and Insurance (BFSI)*. [online] Picussecurity.com. Available at: https://www.picussecurity.com/resource/blog/financial-services-cybersecurity-performance-2024

Mastercard, 2024. *Payment Fraud Prevention Solutions for Issuers | Mastercard*. [online] Available at: https://www.mastercard.us/en-us/business/issuers/business-payments/fraud-prevention.html.

Mclean, M. (2023). *2021 Must-Know Cyber Attack Statistics and Trends - Embroker*. [online] Embroker. Available at: https://www.embroker.com/blog/cyber-attack-statistics/.

Mehrotra, K. and Turton, W. (2021). *CNA Financial Paid Hackers $40 Million in Ransom After March Cyberattack*. [online] Bloomberg.com. Available at: https://www.bloomberg.com/news/articles/2021-05-20/cna-financial-paid-40-million-in-ransom-after-march-cyberattack?utm_source=chatgpt.com

Natalucci, F., Qureshi, M. and Suntheim, F. (2024). *Rising Cyber Threats Pose Serious Concerns for Financial Stability*. [online] International Monetary Fund. Available at: https://www.imf.org/en/Blogs/Articles/2024/04/09/rising-cyber-threats-pose-serious-concerns-for-financial-stability.

NIST (2019). *NVD*. [online] Nist.gov. Available at: https://nvd.nist.gov/. OCC.gov. (2020). *2020 News Releases | OCC*. [online] Available at: https://www.occ.gov/news-events/newsroom/news-issuances-by-year/news-releases/2020-news-releases.html

OWASP Foundation. (2023). *API Security Top 10*. https://owasp.org/API-Security/

Pimentel, B. (2024). *Cybersecurity risk management: An overview*. [online] Thomson Reuters Law Blog. Available at: https://legal.thomsonreuters.com/blog/cybersecurity-risk-management-an-overview/.

Ranjan, P., Akhil Khunger, Chalamayya Batchu Veera Venkata Satya and Sumit Dahiya (2022). Threat Modeling and Risk Assessment of APIs in Fintech Applications. *ESP Journal of Engineering & Technology Advancements (ESP-JETA)*, [online] Volume 2(Issue 2), pp.44–61. Available at: https://www.espjeta.org/jeta-v2i2p108#

Rebeka, P. (2018). *KYC and AML — the Difference and Best Practices | SumSub.com*. [online] Sumsub. Available at: https://sumsub.com/blog/kyc-and-aml/.

Scott, M. (2025). ESG Watch: Companies 'complacent about cybercrime', despite rise in risk from AI. *Reuters*. [online] 3 Feb. Available at: https://www.reuters.com/sustainability/sustainable-finance-reporting/esg-watch-companies-complacent-about-cybercrime-despite-rise-risk-ai-2025-02-03/.

SearchSecurity. (n.d.). *What is attack vector?* [online] Available at: https://www.techtarget.com/searchsecurity/definition/attack-vector.

Sift, 2024. *Digital Trust & Safety: Go beyond fraud prevention with Sift*. [online] Available at: https://sift.com/.

Singer, P. W., & Friedman, A. (2022). *Cybersecurity and Cyberwar: What Everyone Needs to Know*. Oxford University Press. https://www.google.co.in/books/edition/Cybersecurity_and_Cyberwar/B88ZAgAAQBAJ?hl=en&gbpv=1&dq=Cybersecurity+and+Cyberwar:+What+Everyone+Needs+to+Know&pg=PP1&printsec=frontcover

State of API Security Q1 2023. (n.d.). Available at: https://content.salt.security/rs/352-UXR-417/images/SaltSecurity-Report-State_of_API_Security.pdf

Statista. (2024). *Cost of a data breach in financial sector worldwide 2022*. [online] Available at: https://www.statista.com/statistics/1324063/cost-of-data-breaches-in-financial-industry-worldwide/.

Stempel, J. (2025). *Robinhood to pay $45 million to settle SEC charges over recordkeeping, other violations*. [online] USA TODAY. Available at: https://www.usatoday.com/story/money/2025/01/14/robinhood-45m-settlement-sec-violations/77694796007/

Stripe.com. (2024). *What are financial APIs? Here's what to know | Stripe*. [online] Available at: https://stripe.com/in/resources/more/financial-apis-explained-what-they-are-how-they-work-and-how-they-are-changing-fintech

Terranovasecurity.com. (2024). *Top 11 Cybersecurity Concerns in the Finance Sector in 2024*. [online] Available at: https://www.terranovasecurity.com/blog/cybersecurity-concerns-finance-sector.

The Times & The Sunday Times (2024). *Get ready for your own CrowdStrike, City regulator tells firms*. [online] Thetimes.com. Available at: https://www.thetimes.com/business-money/companies/article/get-ready-for-your-own-crowdstrike-city-regulator-tells-firms-tp0t57pst.

Tuoma, Petrus Aleksi and Ekegren, W.E. (2021). *Banking-as-a-Service and the transformation of the finance industry: An empirical investigation*. [online] Lub.lu.se. Available at: https://lup.lub.lu.se/student-papers/search/publication/9053388

Vanderford, R. (2025). *GenAI Increasingly Powering Scams, Wall Street Watchdog Warns*. [online] WSJ. Available at: https://www.wsj.com/articles/genai-increasingly-powering-scams-wall-street-watchdog-warns-a6592d54

World Economic Forum (2024). *Global Cybersecurity Outlook 2024 J A N U A R Y 2 0 2 4 In collaboration with Accenture*. [online] Available at: https://www3.weforum.org/docs/WEF_Global_Cybersecurity_Outlook_2024.pdf.

Ycombinator.com. (2022). *Binance is trying to calm investors, but its finances remain a mystery | Hacker News*. [online] Available at: https://news.ycombinator.com/item?id=33949162

Figures

Tables

Pictures

Appendices

Appendice 1: Parsing JSON data using R

This section describes how the raw vulnerability dataset from NVD is **downloaded**, **parsed**, and **converted** into a structured format in R.

```
> library(jsonlite)
> library(dplyr)
> json_file <- "C:/Users/hp/Desktop/Thesis/Arpita Thesis/nvdcve-1.1-modified.json"
> if (!file.exists(json_file)) {
+     stop("File not found! Check the file path and try again.")
+ }
> nvd_data <- fromJSON(json_file, flatten = TRUE)
> str(nvd_data, max.level = 2)
```

This step ensures that the **CVE vulnerabilities dataset** is successfully imported into R for further processing. The dataset contains key fields such as **CVE ID, description, CVSS severity scores, CWE mappings, and references**.

Appendice 2: Keyword-Based Filtering

To isolate vulnerabilities relevant to **API security**, the following R script **filters the dataset** based on key API-related terms.

```
> library(stringr)
> api_keywords <- c("API", "authentication", "token", "OAuth", "data exposure", "injection",
+                   "REST", "GraphQL", "endpoint", "authorization", "JWT", "CORS",
+                   "server-side request forgery", "rate limiting", "access control", "CSRF")
> api_vulnerabilities <- vulnerabilities_clean %>%
+     filter(str_detect(str_to_lower(description), str_c(str_to_lower(api_keywords), collapse = "|")))
> nrow(api_vulnerabilities)
[1] 845
> table(api_vulnerabilities$cvssV3_severity)

CRITICAL     HIGH      LOW   MEDIUM
     113      149        2      232
```

Appendice 3: Data Cleaning & Preprocessing

The following script extracts key attributes such as CVE ID, Description, CWE, CVSS Scores and handles missing values.

```
> library(purrr)
> library(tidyr)
> vulnerabilities_clean <- nvd_data$CVE_Items %>%
+     select(
+         cve_id = cve.CVE_data_meta.ID,
+         publishedDate,
+         lastModifiedDate,
+         description_data = cve.description.description_data,
+         references = cve.references.reference_data,
+         cvssV3_score = impact.baseMetricV3.cvssV3.baseScore,
+         cvssV3_severity = impact.baseMetricV3.cvssV3.baseSeverity,
+         cwe_data = cve.problemtype.problemtype_data
+     ) %>%
+     mutate(
+         publishedDate = as.Date(substr(publishedDate, 1, 10)),
+         lastModifiedDate = as.Date(substr(lastModifiedDate, 1, 10))
+     )
>
> vulnerabilities_clean <- vulnerabilities_clean %>%
+     mutate(description = map_chr(description_data, function(x) {
+         if (!is.null(x) && length(x) > 0 && is.data.frame(x)) {
+             desc <- x %>% filter(lang == "en") %>% pull(value)
+             if (length(desc) > 0) return(desc[1])
+         }
+         return(NA_character_)
+     })) %>%
+     select(-description_data)
```

```
> vulnerabilities_clean <- vulnerabilities_clean %>%
+     mutate(cwe = map_chr(cwe_data, function(x) {
+         if (is.list(x) && length(x) > 0) {
+             first_entry <- x[[1]]
+             if (is.data.frame(first_entry) && "description" %in% names(first_entry)) {
+                 cwe_desc <- first_entry$description[[1]]
+                 if (is.data.frame(cwe_desc) && "value" %in% names(cwe_desc)) {
+                     return(cwe_desc$value[1])
+                 }
+             }
+         }
+         return(NA_character_)
+     })) %>%
+     select(-cwe_data)
> sum(is.na(vulnerabilities_clean$cwe))
[1] 2210
```

Appendice 4: API Vulnerabilities Trend Over Time

This appendix provides a yearly count of API vulnerabilities, showcasing how they have increased over time.

Year	Number of Vulnerabilities (n)
2025	943

```
> library(dplyr)
> library(lubridate)
> library(jsonlite)
> json_file <- "C:/Users/hp/Desktop/Thesis/Arpita Thesis/Extracted/nvdcve-1.1-recent.json"
> nvd_data <- fromJSON(json_file, flatten = TRUE)
> cve_trends <- nvd_data$CVE_Items %>%
+     transmute(year = year(as.Date(substr(publishedDate, 1, 10)))) %>%
+     count(year)
> write.csv(cve_trends, "C:/Users/hp/Desktop/Thesis/Arpita Thesis/Extracted/API_Vulnerabilities_Trend.csv", row.names = FALSE)
```

The table above presents the trend of API-related vulnerabilities reported over time. The data is derived from the NVD dataset, highlighting the increasing frequency of security flaws in financial APIs.

Appendice 5: API Vulnerabilities Trend Over Time

This appendix classifies vulnerabilities into **Critical, High, Medium, and Low** based on **CVSS v3 scores**.

CVSS Severity	Number of Vulnerabilities (n)
CRITICAL	18
HIGH	72
MEDIUM	96
LOW	25

The severity levels of API vulnerabilities are categorized based on the CVSS v3 scoring system. A significant number of vulnerabilities are classified as 'High' or 'Critical,' emphasizing the need for financial institutions to enhance their security measures.

```
> library(dplyr)
> library(jsonlite)
> severity_distribution <- nvd_data$CVE_Items %>%
+     filter(!is.na(impact.baseMetricV3.cvssV3.baseSeverity)) %>%
+     count(impact.baseMetricV3.cvssV3.baseSeverity)
> write.csv(severity_distribution, "C:/Users/hp/Desktop/Thesis/Arpita Thesis/Extracted/API_Vulnerabilities_Severity.csv", row.names = FALSE)
```

Appendice 6: Attack Vector Distribution of API Vulnerabilities

This appendix shows **how API vulnerabilities are exploited** (e.g., Network-based, Local, Physical).

Attack Vector	Number of Vulnerabilities (n)
NETWORK	186
ADJACENT NETWORK	17
LOCAL	5
PHYSICAL	3

This table outlines the attack vectors used to exploit API vulnerabilities. The majority of threats originate from network-based attacks, indicating that APIs exposed to the internet are at higher risk.

```
> library(dplyr)
> library(jsonlite)
> attack_vector_distribution <- nvd_data$CVE_Items %>%
+     filter(!is.na(impact.baseMetricV3.cvssV3.attackVector)) %>%
+     count(impact.baseMetricV3.cvssV3.attackVector, sort = TRUE)
> write.csv(attack_vector_distribution, "C:/Users/hp/Desktop/Thesis/Arpita Thesis/Extracted/API_Vulnerabilities_Attack_Vector.csv", row.names = FALSE)
```