EAI Endorsed Transactions
on Cloud Systems

Research Article EAI.EU

Application Programming Interface (API) Security in Cloud Applications

F. A. Qazi[1,*]

[1]Department of Cybersecurity, Capitol Technology University, Laurel, MD 20708

Abstract

Many cloud services offer an API gateway to users through API platforms such as Platform as a Service (PaaS), Software as a service (SaaS), Infrastructure as a Service (IaaS) and cross-platforms APIs. APIs are usually designed for functionality and speed by developers who write only a small portion of code. The code that the developers write has visibility that they may consider secure, but the code created from third-party software or libraries has no visibility, which makes it insecure. As a result, APIs are the most vulnerable points of attack, but many users do not recognize their insecurity. Organizations that use APIs, but which do not maintain them are likely to have a large attack surface, leading to security breaches. This paper reviews (1) API security in cloud applications and (2) discusses details of API vulnerabilities, existing security tools for API security, and machine learning techniques to mitigate API attacks. This study shows that most users are unaware of API insecurity, that most organizations lack resources and training to educate users about APIs, and that most organizations depend on the overall security of the network instead of the security of standalone APIs.

Keywords: Cloud API, Cloud/API Security, Cybersecurity, zero trust

Received on 31 January 2023, accepted on 26 September 2023, published on 17 October 2023

doi: 10.4108/eetcs.v7i23.3011

1. Introduction

The dramatic growth of cloud computing is a result of the popularity of APIs, as well as the increasing use of smart phones and tablets interfaced with other connected devices. In fact, the first cloud application programming interface (an API), launched by Amazon in 2006, stored or retrieved large amounts of data from Amazon servers [1]. APIs provided the needed communications between cloud applications or the infrastructure of the cloud, where infrastructure APIs could handle virtual machines and their workloads in the cloud. Examples of cloud provider services include Amazon Web Services, Microsoft Azure, the Google Cloud Platform, and other cloud platforms that use APIs to control all user-related operations. The recent Covid-19 pandemic increased the creation of cloud APIs in such sectors as healthcare. This increase was helped by medical professionals, researchers, and government experts who used cloud APIs to access real time data on drug availability and patients. There was a 56% increase in the demand for API requests between October 2020 and October 2021. Users of the Postman API platform, for example, made 855 million API requests, created 30 million collections (up by 30%) and signed in from 234 different countries and geographies including Antarctica [2]. The pandemic also helped to increase e-commerce activity where cloud AP promoted transfer of funds between customers and companies. It is predicted that the cloud API market may grow at a compound average growth rate (CAGR) of 23.2% to reach US$ 3.71 billion by 2030 [3].

Cloud APIs are widely used to access sensitive data and applications in the currently globally connected world, and so they become more vulnerable to malicious attacks. An average-sized business normally manages 363 APIs, and these APIs experience an average of 26.7 vulnerabilities in any one year [4]. Security risks increase as the number of APIs increases because API developers tend to focus more on

*Corresponding author. Email: faqazi@captechu.edu

functionality than on the necessary security. Thus, APIs become insecure because of their design. According to the 2023 Q1 Salt Security report, API attacks on Salt customers increased by 400% compared to the previous six months. In addition, 31% of customers experienced sensitive data exposures and 17% suffered data breaches from API security gaps [5]. In June 2022, malicious API traffic was 2.1% of overall traffic and API attacks were reported to be 26.46 M malicious calls compared to 12.22M the year previous. Data attacks on consumers are increasing every month [6]. It was reported that US companies face a combined $12 billion to $23 billion in losses from API breaches, which occur with the increased use of cloud services and DevOps-style development methodologies [7]. Our qualitative study showed that many organizations do not know how many APIs they have or even whether they have any at all. This author has suggested various techniques to mitigate attacks from insecure APIs, including better training for staff and implementing security protocols at every step of the API design process. The author has also suggested methods based on machine learning and artificial intelligence to analyse API breech data to prevent any future data attacks [8]. This paper is divided into eight parts. The first part contains the introduction which outlines the purpose for securing APIs in the cloud. The second part consists of two sections; (1) API Security in the Cloud; and (2) Cloud API security protocols. The third part consists of (1) Vulnerabilities of APIs; (2) Common API attacks and mitigation; (3) API breaches. The fourth part discusses the best practices for API security. The fifth part details a qualitative case study methodology for API security, performed by the author, and consists of five sections: (1) qualitative case study for API security; (2) survey questions to the participants; (3) levels of inductive coding; (4) sample population and themes; (5) mitigation of API vulnerabilities and practical implications. The sixth part consists of (1) Machine learning for cloud API security (2) utilization of machine learning tools for cloud security and (3) detecting attacks in cloud APIs using machine learning. The seventh part consists of (1) artificial intelligence for cloud API security and (2) applications of AI in cloud API security. Finally, the eight parts consists of the conclusion and future research work.

2. API security in the cloud

As a result of continuous growth in cloud computing services, many organizations are migrating their data to a public or private cloud infrastructure. This shift is called “cloud migration” and is accomplished by using APIs that connect clients to cloud infrastructures. An API is a type of software interface that allows digital devices, software applications, and data servers to communicate or connect with each other. Many cloud services use an API gateway positioned in front of an API in the network infrastructure as a single-entry point that is used to verify and identify users. The API gateway, designed as a security gateway, monitors and centralizes identity management [9].

Cloud APIs are types of APIs that allow for the creation of cloud infrastructure, software, services, and applications for various cloud platforms. The operation of such APIs is shown in figure 1.

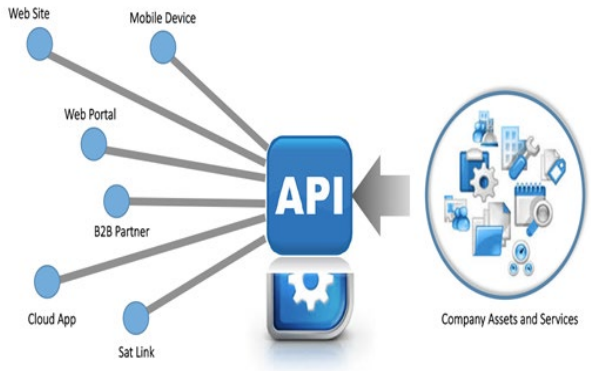

Figure 1. Operation of API Source

The current working environment easily tolerates hacking and allows attack surfaces to increase. So, it is important to strengthen APIs even in a secure environment. Although networks have become safer, partly with the application of zero trust security and DevOps, vulnerabilities in access and use of applications persist through the APIs. Instead of implementing security at each stage of the coding process, unsafe API coding practices continue. In fact, the APIs are designed mainly for functionality and speed rather than security. Thus, hackers can enter a system using a backdoor and gain access to the digital keys of a related system [10]. GitHub, the encrypted digital source code management system, is accessed by hackers who can regularly download the digital access keys to Amazon web services and other systems [11]. System developers routinely leave configuration files, user data, and company assets inside source code for easy access to applications, which encourages hackers to use these credentials. In the case of the Panera Bread’s data breach, the API executed by Panera Bread was unauthenticated, giving malicious users the opportunity to gain access to code used to retrieve the last four digits of customers’ credit card numbers, which were stored in plain text [12]. APIs are also exposed to the risk of bot cyberattacks in which hackers use bots to conduct DDoS, DoS and ATO (account takeover) attacks as well as input fuzzing, inventory hoarding and vulnerability scans. According to PerimeterX, 75% of login attempts from API endpoints are malicious. In 2020, 98% of organizations experienced attacks against their applications/websites, and 82% of these reported to be bot-related cyberattacks [13].

Cloud services host API servers, and many built in features are managed and maintained by cloud service providers. There are more than 15 cloud services providers worldwide, the most popular global cloud service providers being Amazon Web Services, Google Cloud, and Microsoft Azure. It is critical to safeguard the APIs with a security strategy, considering the increased numbers of malicious attacks on web applications over the internet. This scrutiny requires controlling the API level of access to different users and in

different locations. The three top cloud service providers all set rules and limit usage, but they use different security controls, as is explained below.

- Amazon Web Services (AWS) offers two kinds of tools to set rules and limits for securing and managing API, which are (1) API gateway, and (2) identity and access management (IAM). API security under API gateway method includes AWS web application firewall (WAF), throttling in AWS API gateway, and AWS shield. AWS WAF protects API and web applications from attacks by enabling them to configure a set of rules that allow or block requests from specified addresses and inspects a range of addresses from certain countries or regions. AWS WAF provides protection against common attacks such as SQL injection and cross-site scripting as well as safeguarding against content scrapers. API gateway throttling, which is configurable, can limit the response to requests sent by a client. AWS shield provides protection and detection service for distributed denial of service (DDoS) attacks. AWS WAF also contains Bot Control for visibility and control over bot traffic, and it blocks the rare limit- pervasive bot, including scrappers, scanners and crawlers. AWS IAM is an alternative way of using API keys to access APIs through its permission and is used for allowing access to internal employees and contractors [14, 15, 16].
- The Microsoft Azure approach for API security on setting rules and limits on usage is successful mainly through Azure WAF and Azure API management. Azure WAF and many of its offerings for API security are offered under the PaaS Application Gateway that provides API security to various cloud applications. Azure Application Gateway is a web traffic load balancer that manages traffic to web applications that protects against XSS (Cross-Site Scripting) attacks and threats such as SQL injection and command injections. It also detects bots and web crawlers in addition to HTTP protocol violations and anomaly detection that can take many such forms as HTTP request smuggling or HTTP response splitting. Azure API management can secure API by such different methods as Authorization Keys, OAuth 2.0 and JSON Web Tokens (JWTs), and Client certificate authentication. Azure WAF also prevents bad bots, scanning for vulnerabilities in web applications. Stopping them will avoid wasting resources and services [14, 15, 17].
- The Google Cloud Platform (GCP) approach for securing API is similar to AWS and Azure on setting rules and limits on usage. GCP provides Cloud Armor, API gateway and Apigee API management tools for securing APIs. Cloud Armor, which is essentially GCP WAF, can provide DDoS protection, rate limiting, IP based and geo-based control. It also offers WAF to protect against SQL injection cross -site scripting and remote code execution. The GCP API gateway can secure and manage all API calls for Google Cloud backend. Apigee, an API management tool brings a service that can increase API security using OAuth 2.0, API keys and Security Assertion Markup Language (SAML) policies. SAML policies are integrated with Apigee and can authorize and authenticate applications via SAML tokens. Apigee advanced API security capabilities can also detect malicious bot attacks and identify API misconfigurations. [14, 15, 18].

2.1. Cloud API security protocols

The security of Cloud APIs involves protecting APIs from attacks. Cloud API security is a key component of current web applications. The API gateway, designed as a security gateway, uses monitoring and centralized identity management [9]. A security gateway functions like a firewall by offering protection for the cloud service, the user, and applications from external malicious network traffic. Macy likened the gateway to a hotel room key. After the room is rented out and the customer gains access to the key, the customer's responsibility is to ensure the safety of personal data assets or digital information [19]. However, APIs are the points of vulnerability that hackers exploit in safe cloud environments. Because an API is designed to carry the data payload of an application, the form of the API depends on whether the payload needs more resources, or it remains lightweight. A cloud API can be offered as an (1) infrastructure as a service (IaaS) that helps provision computing and storage, (2) as software-as-a-service (SaaS) API, which connects to software and applications, or (3) as a platform-as-a-service (PaaS) API, which can create applications and software.

APIs follow a specific set of rules that implement how applications or databases can connect and communicate with each other. Simple object access control (SOAP) and representation state transfer (REST) are different approaches that define how to build APIs that allow data communication between cloud applications. GraphQL is a new query language API that can be used as an alternative to REST. The details of each are given below.

2.2. SOAP API protocol

SOAP, a protocol for web services, was designed by Microsoft as a web communication protocol and released as XML-RPC in 1998. In May 2000, it became the official protocol developed by a joint IBM and Microsoft through the World Wide Web consortium (W3C). It uses only XML for exchanging information in its message format and has been used for private and public APIs as a standard protocol. SOAP has been dominated by REST in recent years, and is more popular in large business enterprises, and health organizations of all sizes for its being more secure. SOAP uses web service (WS) security in addition to SSL, which prevents unauthorized users from accessing the system. Because SOAP is a messaging protocol, security of SOAP API is focused on preventing unauthorized access to the received and sent messages from SOAP APIs containing messages and user's information. SOAP is the standard API used for complex and high-security applications such as

financial transactions, banking, and payment gateways. SOAP can employ various protocols of the application layer. Authenticity and error handling are built into this protocol, making it more secure than the REST API [20].

2.3. REST API protocol

REST API was developed by a group of experts led by Roy Fielding in 2000 for building web and mobile applications through the evolving global internet. REST is a set of architectural constraints, not a protocol or standard like SOAP. The constraints include client server, cacheable, code on demand, layered system, and uniform interface. The use of REST APIs has increased over the last few years; many were used in public web services from 2000 until 2018. The REST API uses HTTP and JavaScript object notation (JSON) for a lightweight data-interchange format [21]. It utilizes secure HTTP for transmitting data via a secure transmission channel and does not have the built -in security capabilities, or extensions like SOAP. As a result, its security depends on the design of the APIs themselves, which can be designed with certain security mechanisms that ensure that only authenticated and authorized users can access them. HTTP basic authentication, JSON web tokens, OAuth, and API keys are the basic authentication methods of REST APIs.

2.4. GraphQL language

GraphQL is a new API standard invented and developed by Facebook in 2012 as its internal project for their native mobile application usage. GraphQL differs from REST in providing only a single endpoint by the server and responding with the precise data that a client may ask for. Since GraphQL is a query language, it is implemented through a set of tools on the server side by modeling the databases as graphs in which objects are represented through nodes and relationships are represented by edges on a graph. It also allows developers construct requests that pull data from multiple data sources in a single API call. GraphQL, however, is slow to perform large or complex queries, which imposes an additional time cost to process each request [22]. GraphQL is based on HTTP which can be accessed over a single endpoint and requests made by the HTTP GET and PUT commands. It is prone to complex security implications by allowing malicious queries or long queries by the legitimate users that can take the GraphQL server down by a Denial of Service (DoS) attack. Some security measures for GraphQL include API server timeout, blocking or limiting complex queries, throttling all client users, and keeping track of query costs [23].

3. Vulnerabilities of APIs

APIs are critical building blocks of a user access to information resources and are centers of communication, including in data processing. Therefore, attackers usually look for API vulnerabilities that compromise API operations. In the second quarter of 2022, the Wallarm (Platform for DevSecOps) research team examined 88,241 records for Top of Form and found 184 API-related vulnerabilities. In the first quarter of 2022; a total of 48 API-related vulnerabilities were found. This increase amounted to +268% (about 2 per day) in the first quarter of 2022. In the second quarter of 2022 the highest risk vulnerabilities were related to OWASP A03 Injection, and then followed by BOLA (Broken object level authorization) [24]. In 2021, the SALT survey reported that 55% of corporations in the previous 12 months had found vulnerabilities in their APIs. In addition, 19% of corporations had discovered their sensitive data to be exposed, 39% found authentication problems, 23% suffered denial of service attacks, 16% experienced brute force attacks or credential stuffing, and 12% faced scraping [25].

The increase in API-related vulnerabilities or flaws gives the attacker opportunities of access to an API, resulting in security threats. As a result, Open Web Application Security Project (OWASP) has developed a list of the top 10 API security risks to APIs, their risks, impacts, and countermeasures. It applies directly to API use and helps organizations address the most serious API security issues that often were overlooked in the design process. OWASP was formed in 2001 by an online community of volunteers who publish articles, methodologies, documentation, tools, and technologies in the field of web application security and provides free tools to organizations for minimizing security risks in their web applications. In 2004, OWASP foundation was established as a 501c3 nonprofit organization in the United States that supports its infrastructure and projects. OWASP is also registered as a nonprofit organization in 2011 in Belgium, and it was called OWASP Europe VZW. The latest OWASP Top 10-2021 is published with a new graphic design because of recent research based on comprehensive data compiled from more than 40 partner organizations. OWASP's list of vulnerabilities is published every year and updated every 4 years as shown below. There are three new categories, four categories with naming and scoping changes, and some consolidations in the top 10 for 2021 [26].

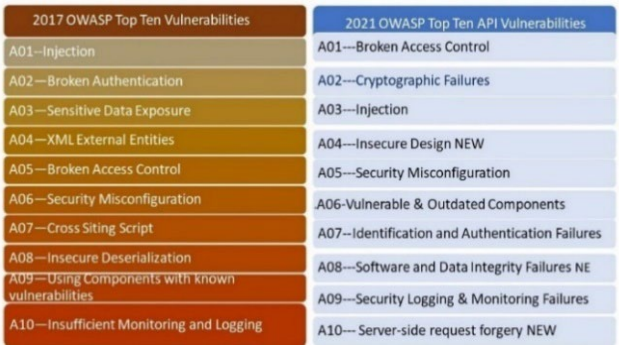

Figure 2. Changes in the top10 vulnerabilities of OWASP from 2017 to 2021

In the 2021 update, Insecure Design, Software and Data Integrity Failures, and a group for Server-side Request Forgery (SSRF) attacks have been added as new categories. XML External Entities (XXE) from 2017 OWASP top 10 vulnerabilities has been added to 2021 OWASP top 10 vulnerabilities. Security Misconfiguration category, and Cross -Site Scripting (XSS) have been added to the Injection section. In addition, Insecure Deserialization is now part of Security Logging and Monitoring Failures. Each vulnerability is listed with Common Weakness Enumeration (CWE), which is a community-developed list of software weaknesses and vulnerabilities [27, 28].

3.1. API attacks with mitigation

API attacks are the result of API security vulnerabilities, such as poor authentication, lack of encryption, and other flaws. Most of the corporations, unfortunately, do not know the number of APIs they have or how to secure them from attacks. According to a recent study by Gartner, API attacks will become the most common issue in 2022 becoming the primary cause of data breaches for web applications, business software, and others. These API attacks will be in addition to API insecurities that are currently damaging many businesses in the world. Gartner also found that approximately 40% of the web-enabled applications will deal with API attacks instead of interface attacks that will get worse in the long run [29]. The most common API attacks are (1) API injection attacks, (2) Distributed Denial of Service (DDos) attacks, (3) Man in the middle attacks, (4) credential stuffing, (5) insecure API key generation; (6) incorrect server security [30,31].

- API injection attack includes SQL injection (SQLi) and Cross-site Scripting (XSS) which are the most common types of API injection attacks because of their wide attack surface. According to an Akami report, 18 months of attack traffic between January 2020 and June 2021 showed that SQLi injection attacks was one of the tops of 6.2 billion web attacks, followed by local file inclusion with 3.3 billion attacks, and XSS with 1.19 billion attacks [32]. Injection attacks can be mitigated by validating and sanitizing all data in API requests and limiting response data to avoid unintentionally leaking sensitive data. Cross-site scripting attacks can be mitigated by validating input, using character escaping, and filtering.
- DDos Attack is one of the most common attacks in which the attacker floods a server with internet traffic to overwhelm API memory and to keep users from accessing online services and connected sites. InfoSecurity magazine reported 2.9 million attacks in the first quarter of 2021 attacks, which is a 31% increase from the same period in 2020 [33]. DDos attacks can be mitigated by rate limiting and limiting payload size.
- A man-in-the -middle attack is another common attack in which an attacker intercepts and possibly alters the communication or eavesdrop sensitive information between a client and server. By issuing an API request to an HTPP header between a session token, a hacker acts as a man in the middle. Different types of man in the middle attacks include rogue access point, address resolution spoofing (AARP), multicast domain system (mDNS), and DNS spoofing. This attack can be mitigated by encrypting the traffic in transit.
- Credential Stuffing is another common cyber-attack, by which the use of stolen credentials on API authentication endpoint gains unauthorized access. This attack can happen because many people reuse usernames and passwords on multiple or on all accounts [34]. One way of mitigating this attack is to use authentication without a password by which users show some other form of identity instead of entering a password or answering security questions. This form of evidence can be biometric in nature including fingerprints, a proximity badge, or a hardware token code.
- Insecure API Key Generation
  Most of the APIs are secured by JWT (JSON Web Token) or through API keys. Insecure API Key Generation is a technique that hackers utilize to bypass API defenses which are usually generated from API security and detection tools. Hackers can create these insecure API keys from many users. One mitigation of this would be to strengthen the API key generation tools.
- Incorrect Server Security
  Organizations protect valuable data and assets held on their servers, as well as protecting the server's resources. Servers that are not configured correctly can be used by hackers as a launchpad to data leaks and breeches. SSL certificates that are misconfigured need to be configured correctly so that the servers do not have backdoors for hackers to steal data.

3.2. API breaches

Attacks on APIs continue to increase every year and cause data breaches. API breaches are a major security problem which can have disastrous consequences leading to exposing millions of sensitive user records. Securing APIs can be complex because applications including web applications, mobile applications, and applications on laptops use different web browsers to access different insecure APIs. As a result of this, these insecure APIs can access data and establish many connections that would make the system vulnerable to hackers [35, 36]. The following are examples of API breaches [37].

- In December 2022, there was a data breach at 3Commas in which malicious users with the help of compromised API keys, stole approximately $20 million from users of the service. 3Commas operates crypto trading and offers traders worldwide the best tools for every bot type of market.
- In November 2022, hackers gained access to Dropbox's GitHub internal code repositories which was the result of a phishing scam. This scam allowed the hackers to access

130 GitHub repositories, some of which contained user data and API keys.

- Optus, the third largest Australian telecommunications company experienced a data breach in September 2022, affecting nearly 9.7 million current and former customers. This breach included access to many customers' names, home addresses, phone numbers, date of births, emails, driving licenses and passport numbers.
- In January 2022, another API breach exposed 1.8 million records of Texans' insurance claims which exposed personal information such as addresses, date of birth, phone numbers, social security numbers and information about workers' injuries. Fortunately, none of the exposed data, according to the Texas Department of Insurance, was misused.
- In December 2021, Twitter suffered one of the biggest API breaches of the year exposing over 5.4 million user records. This data breach was caused because of API vulnerability and contained both public information (usernames, Twitter IDs, locations) as well as private information (including email addresses and phone numbers).

4. Best practices for cloud API security

APIs are often a source of security concerns and threats because many businesses and organizations either are not aware of their existence or of their own insecurity. An insecure and poorly maintained API can be an easy target for hackers to gain access to an otherwise secure system. Because APIs work as a backend framework for mobile and web applications, one must protect sensitive data transferred between systems. To secure APIs and enhance API security, one must follow the following practices [38, 39, 40, 41].

- Identify vulnerabilities and associated risks: The first important way to secure APIs is to know about their vulnerabilities and risks by referring to OWASP's top 10 security vulnerabilities presented earlier. By identifying these vulnerabilities early in the API design stage, some vulnerabilities can be fixed before too much damage. Using thorough security testing, vulnerable API threats can be discovered. API security testing can also help identify parts of the API that are vulnerable to known threats.
- Authentication and Authorization: Strong authentication followed by authorization is an important best practice for API access control. Authorization uses OAuth, which is a token-based protocol that allows information to be addressed by third parties without exposing login credentials. Authentication is necessary for securely verifying the user of the API, and authorization is concerned with what data they have access to. API authentication encourages restriction or removal of users who abuse the API. API authorization usually starts after the identity is confirmed through authentication and then it verifies whether users or applications have permission to access the API.
- Tokens Usage: The use of tokens to restrict API resources to the few users who should be allowed to access them is a relatively simple but effective API security best practice. This step is possible after using a token-base for the API calls on the client-side applications that are made to the service, which can then validate that token and get user information from it. OAuth can create token-based authentication for API calls.
- Encrypt Data: All data transfer from the user to the API server or vice versa must be encrypted using the Transport layer security (TLS) protocol. As TLS does not encrypt data sitting behind the API, the sensitive data in the database layer should also be encrypted. A signature from authorized users should be required by the developer while users decrypt and modify data.
- Rate limiting and Throttling: With the increase in the number of APIs, some rate limiting, and throttling can help to avoid attacks. Rate limiting limits the number of requests an API accepts within a given period and it rejects requests that exceed the limit. Throttling is another common way to practically implement rate-limiting by allowing the API to assess each request. When the throttle is triggered, a user may be disconnected or have bandwidth reduced, causing a reduction in response rate.
- Use of an API gateway: An API gateway is an API management tool that lies between a client and collection of backend services and acts as the major point of enforcement for API traffic. The role of the API gateway is to authorize the incoming request, to direct them to appropriate backend services on defined policies, and to allow organizations to authenticate traffic, as well as to control and analyze how APIs are used.
- Implement a zero-trust philosophy: Zero Trust policy is based on the principle that no traffic inside a modern network, either internal or external traffic, is trusted unless verified by multiple sources. The policy should be applied to both authorized API endpoint, authenticated clients and to unauthenticated and unauthenticated entities.
- Validate Input: Input validation should always ensure that incoming data are legitimate and will not cause harm because requests sometime from perfectly valid sources may attempt hacking. The process of validating input involves checking all user-supplied entries such as username or a credit card number, in a web that meets the field data requirement. It should be strictly adhered to. Input should never be passed from an API through to the endpoint without validating it as early as possible, preferably when the data is received from the external party.
- Expose only limited data: APIs often reveal such information as passwords, keys, and other details

about the API endpoints in addition to such sensitive data as gender, location, financial standing, and medical history. APIs should expose only data necessary to fulfill their operation. To limit accidental exposure, organizations should incorporate scanning tools into their DevSecOps processes.

- Web Applications and API Protection (WAAP): Traditional firewall and API gateway cannot prevent attacks on APIs calls from web and mobile apps carrying sensitive information. As a result, Gartner analysts Jeremy D'Hoinne and Adam Hils developed cloud based WAAP that is specifically designed to protect web applications and APIs [38]. WAAP, located at the outer edge of a network in front of the public side of web applications, analyzes incoming traffic. It centers itself on around four capabilities--including DDoS, Web Application Firewall, Bot Management, and API protection.

5. Qualitative case study methodology for API insecurity

The author conducted a qualitative case study [8] to investigate why the insecurity of APIs is so often overlooked, to understand the vulnerabilities of APIs, and to find ways to mitigate vulnerabilities. The problem of insecurity of APIs is well suited for exploratory case study design because policies for the security of APIs consist of many complicated issues, and the back-end process of APIs provides abstraction and limitations for case study design. A survey was used to collect responses from developers, engineers, managers, and users of APIs. The theoretical background used to anchor the study was the API Security Maturity model, which is based on the Richardson REST Maturity model [20] and describes four levels for REST APIs: Level 0, 1, 2, and 3 [42]. The goals of this study were to research the phenomenon of the insecurity of APIs and the unawareness of insecure APIs among users and developers. The case for this study is that APIs are mostly insecure since the developers focus more on usability than security to save time before deployment. Users, developers, and organizations fail to see this insecurity when utilizing APIs. The sections below illustrate the methodology used for this study.

5.1. Survey questions

The following survey questions were presented to the participants for this study [43, 44, 45, 46].

- How many APIs are in your organization and are they public or private APIs?
- How many of the APIs in your organization are secure versus the insecure ones?
- Are the APIs authenticated via passwords, Auth 2.0, or neither?
- Does your organization provide information about API breaches (if any)?
- How is an encryption key transferred to the user and via URI?
- Do you think that the organization's security team is adequate for the security of APIs?
- Are you aware of the various API attacks such as injection attacks, DDOS attacks and replay attacks?
- If a data or API breach occurs, are you notified about the details?
- Does the security in your organization rely on web-based security methods such as firewalls?
- Are you aware of hidden APIs and onions?
- Since APIs are not user friendly, do you overlook them in web applications and security assessments?

The data analysis process for this qualitative case study is iterative or cyclic beginning from the survey's initial stage and ending at the response analysis. At the final response analysis phase, recurring themes were identified and coded accordingly. Qualitative purposeful sampling of population was used for this research because information-rich cases were identified and selected [47].

The survey tool and method for APIs that was used for conducting the survey among the sample population is direct contact through e-mail. Some APIs for application were also accessed and studied by the author. The data was interpreted by the author and established the importance of analysis using the respondents' and coded data [48, 49].

To summarize, this qualitative research used a single one-shot, one method survey with only one cycle consisting of research question-data collection-analysis-report. The survey was designed by the author and was based on a comprehensive research review of the subject [50, 51]. The author and fourteen participants of the survey were randomly chosen from users and organizations formed a partnership in discovery and interpretation of findings [52, 53, 54].

5.2. Levels of inductive coding

Before this study was conducted, there were only a handful of analysis research studies performed on insecurity of APIs and an inductive content analysis was used in this study [55]. The first level of three-step coding process involved segments of data and was summarized into various descriptive codes which provided the basis of a higher-level coding [56]. Second level coding is more inferential, focuses on pattern coding and yields categories which are "broad units of information which consist of several codes aggregated to form a common idea. "Themes emerged at the third level of the three-step process of coding, which required interpretation from the author at a more abstract level and thus, required more intellectual and effective content [57; 58].

Three levels of analysis were followed in this study based on thematic analysis. The first level of coding or axial coding was used on the respondents' data from the survey to organize

into similar code words to eventually identify themes [59; 60; 61]. After line-by-line coding and constant comparison technique. Each manual code was entered into NVivo13 (the main statistical software used) and resulted in forty-three codes. The word query and source code data tools were then used to illustrate the data in a graphical format. Next the codes were grouped together into categories. Finally, themes were developed using selective coding [62; 63].

5.3. Sample population and themes

The population sample group for this study consisted of developers, users, and technical personnel of various organizations. Some of the respondents work with cyber security while others have software engineering experience. All of them were either users or developers of various APIs.

Three levels of analysis were performed in this study and based on thematic analysis [64,65,66]. The first level of coding or axial coding analysed the respondents' data from the survey to organize the data into codewords and categories to identify themes eventually, as shown below.

Categories	Themes
No transparency	
API security not adequate	**Insecure APIs**
No resources for API security	
Unaware of API number and security	
Lack of authentication	**API Vulnerabilities**
Security risks	
Reliance on network security	
Insufficient notifications	**False sense of security**
Sparse training	
Improving API security	
Evaluation and testing	**API security improvement**
Information and selection	

Figure 3. Categories and Emerging Themes

The following four themes were developed using selective coding [66] based on the main categories:

- Theme 1: To increase awareness of insecurity of APIs and use existing security technology capabilities to secure APIs.
- Theme 2: Vulnerability of APIs; Mitigate the vulnerabilities of APIs by the proper usage of authentication and authorization by managing keys, rate limiting, and securing access to the end points of APIs.
- Theme 3: False Sense of Security; Introduce and track security infrastructure locally for the APIs separate from the central security of an organization.
- Theme 4: Improvements in security of APIs; Implement improvements in security of APIs.

This study verified that APIs are insecure, organizations lack resources and training to educate users about APIs, and users depend on the overall security of the network instead of standalone API security. The study also showed that most users and organizations are unaware of the APIs that they use, and they rely as well on third-party providers to control their APIs. This practice can be dangerous in that the user or organization has no control over their APIs, which can lead to third-party providers misusing their customer's APIs leading to nontransparency in API design. Given the increasing number of APIs in organizations over the years, it is crucial to ensure that API security protocols be followed. API security must be implemented at each step of the design process and user training practiced in organizations to make sure employees and other users of APIs use APIs correctly and safely.

5.4. Mitigation of API vulnerabilities and practical implications

Mitigating API vulnerabilities is an important practice in reducing API insecurities. Mitigation helps to ensure the success of organizations with both internal and external clients. In addition, the use of existing tools for authentication and authorization can be made available to the users at the time of access or call to APIs to mitigate API security vulnerabilities. The resources must be made available in the form of API security and education.

The findings from the study showed that education is needed to increase the visibility of APIs. Existing technology can increase awareness of API insecurity. Security tools for the use of third-party software by the developers and designers should be developed to secure code used in the API design and deployment. It is also suggested that a form of certification like Transport Layer Security (TLS) exchange using simplified TLS API be used when either a file is uploaded or code for a third-party software application is downloaded. A POSIX socket, which is a communication endpoint, for a simplified TLS API is used to avoid complex TLS libraries [67]. The code can be certified by making the code secure and convenient to use by an agency at the time of data exchange. Another suggestion would be to provide encryption, both homographic and traditional, to enable the storage and transmission of secure software for APIs.

The practical implications of the results from this study were the mapping of the results to the newly released National Institute of Standards and Technology (NIST) standards SP 800–204. The only specifications for API security that exist at the time of writing this paper are NIST specifications for microservices [45]. To map the themes which resulted from this study to the practical aspects of the NIST specifications, 8 of the 13 specifications were selected that were applicable to API, which included micro services APIs security: MS–SS–1, 2,5, 8, 9, 10, 12. The following

table maps the four themes from the study for practical implications to the standards set by NIST SP 800–204.

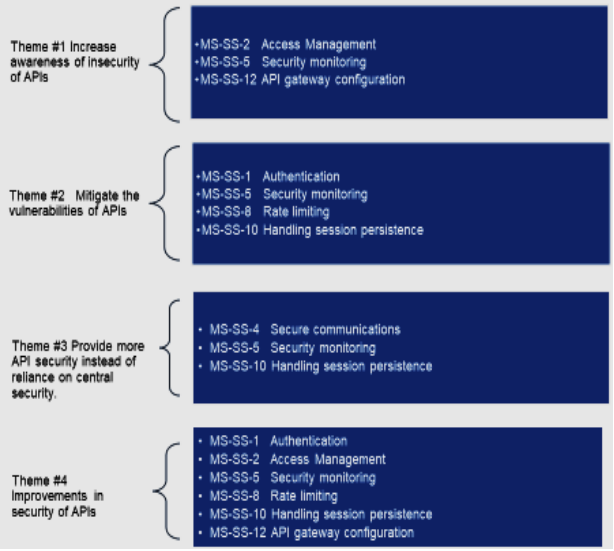

Figure 4. Practical Implications Mapping

6. Machine Learning (ML) for cloud API security

Machine learning (ML) is an application of artificial intelligence that gives computers the capability to learn without traditional programming but with the use of algorithms. It allows computers to learn automatically without human intervention by using samples of historical data. The traditional measures for API security such as API access using authorization, authentication, rate limiting and network privacy, are powerful tools. However, these tools do not address specialized threats like API-specific denial of service (DoS), data and log-in attacks. ML can be used to identify various types of threats, suspicious IP addresses and abnormal behavior by analyzing data faster and more efficiently than traditional means of security. It can be used in businesses to analyze and respond to threats faster and more efficiently than traditional means of security [68].

ML can be used in many areas, including for security in cloud computing. Businesses can use it to analyze and respond to threats faster and more efficiently than traditional means of security. For example, Amazon has developed and launched a service to use ML to analyze data stored on its cloud storage [69]. In the case of problems encountered by cloud server application performance that are flagged by the ML algorithm, the security staff would check the logs to detect the application or the infrastructure that caused the problem(s). This process is time-consuming because the log data files in a public cloud can become voluminous. One type of log data is the cloud operation logs system. For example, these logs include EC2 host logs, application logs, perimeter (VPN packet Trace) logs, CI/CD DevOps builds and release logs, and API access logs [70].

APIs expose data and information in the cloud to the actual application running in the cloud so that, if the API is not secure, then the data is exposed to hackers [20]. A. Walker recalls this concept as giving away one's door keys to a malicious person. Further, he explains that cloud providers have an API gateway. Because of this single point of entry, a secure API gateway has limited access. If the gateway or front door access is initially insecure, then security will be hard to manage after new exploits are discovered. Using APIs with a poor security infrastructure leads to massive data breaches, such as the data breach at Panera Bread in 2014 [20, 71]. As mentioned above, APIs can rely on machine leaning (ML) algorithms for their security. Most ML algorithms have been based on supervised learning on large data sets, whereby they learn by passing the information through a set of interconnected nodes, or neurons. These neurons have adjustable weights so that the information flowing through the layers of these nodes can predict the correct output. ML can serve as an enhancement to cloud computing, leading to what is called the "intelligent cloud" [72]. With the aid of ML in the cloud, the capabilities of the intelligent cloud could increase mainly from just storage to learning from the data stored there. This will help in prediction analysis and allow for more efficient use of the cloud. ML is prevalent in various applications, ranging from APIs for certain tasks (i.e., online shopping) to a car's online interface.

6.1. Utilization of machine learning tools for cloud API security

With increased computer security and staff to maintain security, data centers have become harder for hackers to access. The points of entry for malware have moved to the end points of networks. In addition, computer networks have become overly complex with the usage of public clouds. It will become even more complex when the Internet of Things (IoT) is fully operational. With this complexity comes the burden of sorting through massive amounts of log data, which will be difficult for humans to process alone. ML can process large, repetitive amounts of data. By sorting through the data, ML can learn about the identity of the hackers and their methods, identify new attacks, and suggest mitigation strategies. However, human interaction is needed to combine all the inferences that ML collects to interpret and put the data into a context to identify any correlation that multiple attacks have. Although ML tools generally make attacks difficult to implement, sophisticated hackers can still overcome them. Researchers noted that traditional intrusion detection systems are not effective for cloud computing. The authors introduced some ML tools for intrusion detection and discussed the disadvantages of using artificial neural networks (ANN). They further suggested the use of extreme learning machine (ELM) for intrusion detection, which may reduce the number of false alarm rates that defines an anomaly (other than the normal patterns) and misuse (known attack patterns) to propose a combination of ANN, artificial bee colony (ABC),

and fuzzy clustering algorithms so that more speedy and accurate weights can be determined to find anomalies in network traffic.

6.2. Detecting attacks in cloud APIs using machine learning

To detect previously unknown attacks, changes are necessary in the patterns of behavior in a network and finding anomalies can only be found by ML algorithms due to the high number of attacks. [73] worked on anomaly detection in cloud infrastructures and workflows through web services; the authors developed a rule-based approach to labeling normal and anomalous clusters by using supervised learning, identifying new signatures from monitoring as a service anomaly detection in a public cloud. [74] gave a comprehensive survey of using ML algorithms to prevent attacks, analyze endpoints, enhance human analyzes, automate repetitive security tasks, and close zero-day vulnerabilities. Drinkwater gave examples of Google using ML algorithms to analyze attacks against mobile endpoints and Amazon using ML to analyze, sort, and store data on the S3 cloud storage system. The applications and services used to provide the cloud infrastructure, equipment, software, and databases are accessed by a cloud API [75]. Amazon developed Amazon ML and Google developed Google Cloud Machine Learning by TensorFlow, an open-source ML development software. IBM developed IBM Watson Analytics, using the Alchemy API to connect to networks. Microsoft's Microsoft Azure gives users access to ML algorithms.

Cloud security management tools based on ML are offered by AWS and Microsoft's Azure Security Center. Amazon Macie guards the network from malicious activity when large amounts of data are downloaded and uncommon login patterns appear or when the data show up in an unexpected location [76] reviewed the methods of dynamic resource management for virtual machines in cloud computing; he reviewed three resource management methods: linear regression (LR), support vector machine (SVM), and artificial neural network (ANN). Fiala found that SVM and ANN outperformed LR when it comes to saving energy by turning off idle machines and workloads [77] in 2018 compared conventional cyber defense tools to ML tools by citing the Defense Advanced Research Projects Agency (DARPA)'s Cyber Grand Challenge. In this challenge, the use of AI algorithms could fix security bugs in seconds whereas conventional cyber security programs would take several months to find and patch them. In the older systems, hackers can simply insert malicious code into the traditional cyber security software programs because these systems were built to look for matches to previous malicious code. A second type of data tool is called behavioral analytics, which analyses IP addresses, user logins, and user sessions. Because behavioral analytics is performed by ML algorithms, the volume of raw data in log files can reach several gigabytes or even a few terabytes.

7. Artificial Intelligence (AI) for cloud API security

Artificial intelligence is a branch of computer science that deals with building smart machines to simulate human intelligence, with emphasis in machine vision, speech recognition, natural language processing and expert systems. It requires a foundation of specialized hardware and software for writing and training machine learning algorithms using some of the popularly known programming languages including Python, R and Java. Unlike traditional software, AI-based systems can perform repetitive and detailed oriented tasks better and faster than humans.

The traditional rule-based security is based on the rules of security systems but is limited by the lack of experts defining these rules, leading to API security breaches. In addition, humans can make mistakes and miss important rules that should be defined in these systems. AI engines analyzing APIs need only to crunch numbers but not rules, which is a powerful protection against security gaps. API security can be enhanced by experimenting with AI integration into security and efficiency strategies that are constantly self-improving. AI-based security systems, when trained by using sophisticated algorithms can detect malware, run pattern recognition and detect minutest behavior of malware or ransomware attacks before it enters the system. Some of the AI and ML platforms are given in the diagram below, which also describes the various uses.

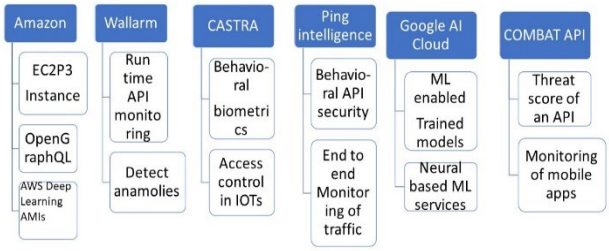

Figure 5. AI and ML Techniques for API security

7.1. Applications of AI in cloud API security

For organizations using and managing APIs, enhanced AI can also protect APIs in the following five security applications.

- Visibility into API traffic to detect and block threats. Existing security solutions using access control policies are a good foundation for securing APIs in the ever-increasing number of diverse clients in the enterprising networks. However, such a solution is not enough to protect against advanced attacks that require visibility into API traffic to detect and block treats. One example of using AI to seek visibility and protection for API infrastructure is Ping Intelligence for APIs [78, 79]. This is a cloud-based service that adds an AI security layer to

the organization's API gateway and monitors all API traffic continuously by tracking activity associated with each user's identity across an entire organization. An in-depth insight is gained into anomalous API activity while deleting and blocking any potential threats. The company services seek to bring visibility to internal- and external-facing APIs, either with gate or no-gate APIs.

- Authentication. Authorization is the process of recognizing a user identity and to validate the individual user of an API to maintain authorized use. Use of AI security solutions can check for authorization through certificates, OAuth 2.0, web tokens, and even Extensible Access Control Markup Language.
- Throttling capabilities. AI-based security tools have the ability of enhancing API security to catch and throttle high-volume cyber-attacks by making use of dynamic checks to data changing rapidly. AI can further pick up on the unauthorized access of information and can prevent the intruder from accessing the information and by blocking the API calls that put information in danger, keeping the system and users safe.
- Static security checks. Static testing is a method of checking the code and designing documents before it is executed to find errors. It does not require the program to be executed because the goal is to find flaws in the early stages of development. AI-based security tools enhance the ability of security programming to detect intrusion through static security checks that examine access patterns and scan payloads. Static checks powered by AI have the potential of catching more than simple rules and gateways by validating requests and scanning for harmful content patterns within calls and messages.
- API dataflow. Dataflow is the movement of data from message source to the destination endpoint through a path consisting of software, hardware or a combination. It often uses a diagram or a model that shows the entire process of data movement. By using AI to analyze the API data flows and the entire customer relationship spectrum early in the sale cycle, customer behavior can be predicted for better marketing campaigns, possible payment delays and other order fulfillment issues before damage occurs. Analyzing customer behavior can predict any possible payment delays and optimize cash recovery. In addition to analyzing supply chain events, stocks can be optimized, shorten delivery delays, and predict any other fulfillment issue before damage occurs.

ChatGPT (Chat Generative Pre-Transformer) created by Open AI can be used to improve API security, for writing APIs, debugging and finding security flaws in existing APIs [80]. ChatGPT was released in November 2022, and is a natural language processing tool that allows human-like conversations and much more with chatbot. However, it needs to be carefully monitored and mitigated to avoid certain risks including data protection and availability and functionality.

8. Conclusion

Our study analyzed the security of APIs in cloud applications, which involves protecting insecure APIs. As the number of APIs increases over the internet, insecure APIs increase attack surfaces. The insecurity of APIs is being overlooked in a zero-security environment giving rise to data breaches and attacks. This paper highlighted the insecurity of APIs, their vulnerabilities, and suggested best practices to mitigate attacks. It is important to follow API security measures, and users should know the presence of APIs and their vulnerabilities. API security must be implemented with state-of-the-art security tools at each step of the design process. Also, training for users should be provided in organizations to make sure employees and other users of APIs use APIs correctly. The author's study showed that most users are unaware of API insecurity, organizations lack resources and training to educate users about APIs, and organizations depend on the overall security of the network instead of the security of standalone APIs. The results of this qualitative study further showed that data vulnerabilities can arise from using insecure APIs in a secure environment and the ways of mitigating attacks from insecure APIs. Current machine learning tools for detecting anomalies and attacks were discussed.

Future research

The purpose of this study was to increase the awareness of insecure APIs by analyzing the respondents' data and being aware that API security should be implemented at all levels of software development. For future research, the author recommended the following:

- Using mixed method studies which includes a comparison of security of APIs using AI algorithms and traditional API security methods.
- Using qualitative or mixed method studies in the future with a larger sample size representing banking, healthcare, and e–commerce industries. This future research can involve users and developers to determine how and when smart API security can be effective so that API security can be relatively easy and fast to implement.
- Use of artificial intelligence tools will be helpful to APIs to detect and stop attacks because they will be enhanced from the large amount of available data.

By not implementing security protocols at the API level, vulnerabilities can arise causing data breaches; however, the technology solutions exist to fix this. Another realization made after this study was the serious lack of representation of engineers, developers and managers of color and female population in the organizations which can be both a limitation as well as a future recommendation.

Acknowledgements.

I am indebted to my young family to whom I could not give enough time to complete this paper. I would like to thank my parents for encouraging and supporting me. Thanks, are also due to Dr. Sondria Miller and Capitol Technical University for helping me to finish my research.

References

[1] Butler, B. (2015). The myth about how Amazon's web service started just won't die: How AWS got started and what its co-founder is doing now that he says could be bigger than cloud. Network world, URL https://www.networkworld.com/article/2891297/the-myth-about-how-amazon-s-web-service-started-just- won-t-die.html

[2] Campbell, S. (2021) Postman's 2021 State of API Report Finds APIs Key to Sparking Innovation During Pandemic, Ushering in API-First World, Business Wire. URL https://www.businesswire.com/news/home/20211028005033/en/Postman%E2%80%99s-2021-State-of-API-Report-Finds-APIs-Key-to-Sparking-Innovation-During-Pandemic-Ushering-in-API-First-World

[3] Market Research Future Cloud (2022) API Market Is Anticipated Grow USD 3.71 Billion at a CAGR of 23.2% by 2030 - Report by Market Research Future (MRFR) URL https://www.globenewswire.com/news-release/2022/09/28/2524089/0/en/Cloud-API-Market-Is-Anticipated-Grow-USD-3-71-Billion-at-a-CAGR-of-23-2-by-2030-Report-by-Market-Research-Future-MRFR.html

[4] Bettendorf, M. (2021) API growth continues to skyrocket in 2020 and into 2021. URL https://blog.postman.com/api-growth-rate/.

[5] Salt Labs (2023) Salt State of API Security Report Q1 2023 https://content.salt.security/state-api-report.html

[6] Salt Security. (2022) Salt Security State of API Security Report Reveals 94% of Companies Experienced Security Incidents in Production APIs in the Past Year. URL https://salt.security/press-releases/salt-security-state-of-api-security-report-reveals-94-of-companies-experienced-security-incidents-in-production-apis-in-the-past-year

[7] Lemos, R. (2022) API Security Losses Total Billions, But It's Complicated," Dark Reading, June 30, 2022. https://www.darkreading.com/application-security/api-security-losses-billions-complicated)

[8] Qazi, F. and S. Miller, A Qualitative Study of Security in Application Programming Interfaces (APIs). In 20th International Conference on Security & Management (SAM'21), July 26-29, 2021, USA.

[9] Bush, T. (2021) What is an API gateway? URL https://nordicapis.com/what-is-an-api gateway/

[10] Sandoval, K. (2015). API Keys ≠ Security: Why API Keys Are Not Enough.
URL https://nordicapis.com/why-api-keys-are-not-enough/.

[11] Berlind, D. (2020) Understanding the realities of API security. URL https://www.programmableweb.com/api-university/understanding-realities-api-security

[12] Deahl, D. (2018) Panera bread leaked customer data on its website for eight months. The verge. URL https://www.theverge.com/2018/4/3/17192348/panera-bread-leaked-customer-data-breach-website,

[13] Chinnasamy V. (2022) Bad bots are coming at APIs! How to beat the API bot attacks?" Help Net Security. URL https://www.helpnetsecurity.com/2022/09/12/api-bot-attacks/.

[14] Psarris, S. (2022) API Security in the Cloud, Reblaze,. URL https://www.reblaze.com/blog/api-security/api-security-in-the-cloud/

[15] Bavati. I (2020) Moving to the Cloud? How to Secure APIs on AWS, Azure, and GCP," Nordic APIs URL https://nordicapis.com/moving-to-the-cloud-how-to-secure-apis-on-aws-azure-and-gcp/

[16] Taylor, D., John Downs, J., Vic Vhorne V; Alex Buck, A. (2020) Azure Web Application Firewall on Azure Application Gateway bot protection overview. URL https://learn.microsoft.com/en-us/azure/web-application-firewall/ag/bot-protection-overview/.

[17] Microsoft, (2022) Protect APIs with Application Gateway and API Management. URL https://learn.microsoft.com/en-us/azure/architecture/reference-architectures/apis/protect-apis

[18] Liu , N. (2022) Google Cloud Combats API Misconfiguration, Bot Attacks. URL https://www.sdxcentral.com/articles/news/google-cloud-combats-api-misconfiguration-bot-attacks/2022/06/

[19] Macy, J. (2018). Public cloud API security: How safe is our data?,
URL https://www.itproportal.com/features/public-cloud-api-security-how-safe-is-our-data/.

[20] Walker, A. (2021) API vs Web Service: What's the Difference?,
URL https://www.guru99.com/comparison-between-web-services.html.

[21] Fitzgerald, A. (2021) SOAP vs REST APIs: The Key Differences Explained for Beginners. URL https://blog.hubspot.com/website/rest-vs-soap.

[22] Sengupta, S. (2021) What is GraphQL Security? Best Practices for GraphQL Security. URL https://crashtest-security.com/graphql-security-vulnerabilities/.

[23] Populi, N. (2018) How to Secure a GraphQL API (The Complete Vulnerability Checklist). URL https://leapgraph.com/graphql-api-security.

[24] Wallarm, (2022) Q2-2022 API Vulnerability & Exploit full report, Wallarm Resource Library. URL https://www.wallarm.com/resources/q2-2022-api-vulnerability-exploit-full-report.

[25] Vizard, M. (2021) Survey Finds API Security Incidents on the Rise URL https://securityboulevard.com/2021/08/survey-finds-api-security-incidents-on-the-rise/.

[26] OWASP. (2021) OWASP Top Ten Web Application Security Risks. URL https://owasp.org/www-project-top-ten/#.

[27] Schmidt, J.) OWASP OWASP Top 10 risks get update, highlighting insecure design — injection No longer on top. URL https://devclass.com/2021/09/28/owasp-top-10-2021/.

[28] Madhani, P. (2021) "OWASP Working Group Releases Draft of Top 10 Web Application Risks for 2021. URL https://www.k2io.com/owasp-working-group-releases-draft-of-top-10-web-application-risks-for-2021/.

[29] Writer, G. (2021) API Security: Protect your APIs from Attacks and Data Breaches. Insight,

URL https://www.itsecurityguru.org/2021/10/21/protecting-your-apis-from-attacks-and-data-breaches/
[30] L7Defence. (2021) API Security Attacks, How API attacks work and How to Identify and Prevent them. URL https://www.l7defense.com/solutions/api-attacks/.
[31] GoldSky Security. (2023) Understanding API Attacks and How to Prevent Them https://www.goldskysecurity.com/understanding-api-attacks-and-how-to-prevent-them/
[32] Akami.com. (2021) Akamai Finds API Vulnerabilities to be a High-Stakes Game for Companies and Individuals Worldwide. URL https://www.akamai.com/newsroom/press-release/akamai-finds-api-vulnerabilities-to-be-a-high-stakes-game-for-companies-and-individuals-worldwide.
[33] Coble, S. (2021) Q1 2021 Sees 2.9 million DDoS Attacks Launched.
URL https://infosecurity-magazine.com/news/q1-2021-sees-millions-ddos
attacks#:~:text=Sarah%20Coble%20News%20Writer%20Approximately%202.9%20million%20Distributed,increase%20compared%20to%20the%20same%20period%20in%202020.
[34] Constantin, L. (2020) APIs are becoming a major target for credential stuffing attacks.
URL https://www.csoonline.com/article/3527858/apis-are-becoming-a-major-target-for-credential-stuffing-attacks.html..
[35] Harguindeguy, B. (2018, January 17). API security: the past, present, and future [Video file]. Retrieved from https://www.brighttalk.com/webcast/288/297033/api-security-the-past-present-and-future
[36] Perry, M. (2019, June 17). The dangerous connections that can damage your business: Why API security is critical to the digital business era [Web log post]. Retrieved fromhttps://www.csoonline.com/article/3502895/the-dangerous-connections-that-can-damage-your-business-why-api-security-is-critical-in-the-digital-.html
[37] Simpson, J (May 2023). 8 Significant API Breaches of Recent Years.
https://nordicapis.com/8-significant-api-breaches-of-recent-years/
[38] Kerner, L. (2020) Critical API security risks: 10 best practices.
URL https://techbeacon.com/security/critical-api- security-risks-10-best-practices.
[39] Juviler, J. (2021) 8 API Security Best Practices to Protect Sensitive Data.
URL https://blog.hubspot.com/website/api- Security.
[40] Backer, S. (2020) Securing APIs:10 Ways to Keep Your Data and Infrastructure Safe.
URL https://www.f5.com/labs/articles/education/securing-apis--10-best-practices-for-keeping-your-data-and-infra.
[41] Chinnasamy, V. (2021) Top 6 API Security Best Practices for 2022.
URL https://www.indusface.com/blog/top-6- api-security-best-practices for-2022.
[42] Sandoval, K. (2020) Introducing the API security maturity model. URL https://nordicapis.com/introducing-the-api-security-maturity-model/.
[43] Farrell, S. (2016, September 25). 28 tips for creating great qualitative surveys [Web log post]. Retrieved from https://www.nngroup.com/articles/qualitative-surveys
[44] Castellani, S., & Dorairajan, A. (2020, April). What are the different types of apis?. *APIfriends.* Retrieved from https://apifriends.com/api-creation/different-types-apis/
[45] Pompon, R. (2018, November 27). Reviewing recent api security incidents. [Web log post]. *f5 labs*. Retrieved from https://www.f5.com/labs/articles/threat-intelligence/reviewing-recent-api-security-incidents
[46] Richer, J., & Sanso, A. (2016). *Understanding API security.* Shelter Island, NY: Manning Publications
[47] Gerring, J. (2007) Case study research: Principles and practices. Cambridge University Press, Cambridge, England, U.K.
[48] Silverman, D. (2000). *Doing qualitative research: A practical handbook*. Thousand Oaks, CA: Sage.
[49] Kabir, S. M. S. (2016). Methods of data collection. In *Basic guidelines for research: An introductory approach for all disciplines* (pp.201–275). Retrieved from https://www.researchgate.net/publication/325846997_METHODS_OF_DATA_COLLECTION
[50] Jansen, H. (2010). The logic of qualitative survey research and its position in the field of social research methods. *Forum: Qualitative Social Research, 11*(2), 1–21. Retrieved from https://www.qualitative
research.net/index.php/fqs/article/view/1450/2947.
[51} Creswell, J. W. (2015). *Educational research: Planning, conducting, and evaluating quantitative.and qualitative research* (5th ed.). Lincoln, Nebraska: Pearson.
[52] Stake, R. E. (1995). *The art of case study research* [DX Reader version]. Retrieved from https://books.google.com/books?id=ApGdBx76b9kC&pg=PA7&lpg=PA7&dq=the+case+study+is+the+study+of+particularity+and+complexity+of+a+case,+coming+to+understand+its+activity+within+important+circumstances&source=bl&ots=KvNMk6Mocu&sig=ACfU3U0621yLWdK_VaU8d446pIIN9ByfXg&hl=en&ppis=_e&sa=X&ved=2ahUKEwjfnqilmPzmAhUDhOAKHS6YBEgQ6AEwC3oECAgQAQ#v=onepage&q=nuance&f=false
[53] Myers, M. D. (1997). Qualitative research in information systems. *MIS quarterly, 21*(2), 1–19.
Retrieved from https://www.researchgate.net/publication/220260372_Qualitative_Research_in_Information_Systems
[54] Braun, V., & Clarke, V. (2013). *Successful qualitative research: A practical guide for beginners*. London, U.K.: Sage
[55] Meng, M., Schubert, A., & Steinhardt, S. (2017). Application programming interface documentation: What do software developers want? *Journal of Technical Writing and Communication, 48*(3), 295–330. Retrieved from https://journals.sagepub.com/doi/abs/10.1177/0047281617721853?journalCode=jtwa
[56] Elliott, V. (2018). Thinking about the coding process in qualitative data analysis. *The qualitative report, 23*(11), 2850–2861. Retrieved from https://nsuworks.nova.edu/tqr/vol23/iss11/14
[57] Creswell, J. W. (2013). *Qualitative inquiry & research design: Choosing among five approaches* (3rd ed.). Los Angeles, CA: SAGE Publications.
Creswell, J. W. (2013, November). Steps in conducting a scholarly mixed methods study. *Discipline-Based Education Research Group*. Retrieved from https://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1047&context=dberspeakers
[58] DeCuir-Gunby, J. T., Marshall, P. L., & McCulloch, A. W. (2011). Developing and using a codebook for the analysis of interview

[59] Saldaña, J. (2013). *The coding manual for qualitative researchers* (2nd ed.). Los Angeles, California: SAGE
[60] Yin, R. K. (2014*). Case study research design and methods* (5th ed.). Thousand Oaks, CA: SAGE.
[61] Dowell, A., Roberts, K., & Nie, J. B. (2019). Attempting rigour and replicability in thematic analysis of qualitative research data; a case study of codebook development. *BMC Medical Research Methodology, 19*(66), 1–8. Retrieved from https://bmcmedresmethodol.biomedcentral.com/articles/10.1186/s12874-019-0707-y#citeas
[62] Bowen, G. A. (2006). Grounded theory and sensitizing concepts. *International Journal of Qualitative Methods, 5*(3), 12–23. Retrieved from http://scholar.google.com/scholar_url?url=https://journals.sagepub.com/doi/pdf/10.1177/160940690600500304&hl=en&sa=X&ei=T-Q6YMKBNMbPmAGYsb6gDA&scisig=AAGBfm37Y2E4rSEPt3olmfOam9E-XH1BcA&nossl=1&oi=scholar
[63] Lewins, A., & Silver, C. (2020). *Using software in qualitative research: A step-by-step guide* (2nd ed.). Thousand Oaks, California: SAGE.
[64] Green, M., & Smith, M. (2016) Developers are not the enemy: The need for usable security APIs.
URL
http://mattsmith.de/pdfs/DevelopersAreNotTheEnemy.pdf..
[65] Yin, R.K. (2014) Case study research design and methods (5th ed.), Thousand Oaks, CA: SAGE.
[66] Patton, M.Q. (2014) Qualitative research & evaluation methods: Integrating theory and practice (4th ed.), Thousand Oaks, CA: Sage Publications.
[67] O'Neill, M., S. Heidbrink, J. Whitehead, T. Perdue, L. Dickinson, T. Collett, N. Bonner, and D. Zappala, "The secure socket API: TLS as an operating system service." 27th USENIX Security Symposium, 799–816, URL https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-o_neill.pdf.
[68] National Institute of Standards and Technology. (2019) "Security strategies for microservices-based application systems," (Special Publication 800-204), Gaithersburg, MD: U.S. Government Printing Office.
[69] Swanner, N. (2017) Build: What 'intelligent cloud, intelligent edge' means.
URL https://insights.dice.com/2017/05/11/build-intelligent-cloud-intelligent-edge/.
[70] Dadhich., P. (2020) Top 10 cybersecurity incidents in 2020. URL https://www.znetlive.com/blog/top-10-cybersecurity-incidents-in-2020/.
[71] Drake, N. & Turner, B. (2021) Best cloud log management services of 2021.
URL https://www.techradar.com/best/best-cloud-logging-services.
[72] Jon. (2021) Cyber-attacks and data breaches list from 2014 to 2021, 2021, URL https://www.51sec.org/2021/02/16/security-events-and-data-breaches-in-2018-2017-2016-2015-2014/,
[73] Gaurav, S. (2017) Machine learning impact on cloud computing.
URL https://www.botmetric.com/blog/machine- learning-impact-on-cloud-computing.
[74] Gander, M., B. Katt, B. M. Felderer, A. Tolbaru, R. Breu, R., & A. Moschitti. (2012). Anomaly detection in the cloud: detecting security incidents via machine learning. URL http://disi.unitn.it/moschitti/articles/2012/JIMSE2012-UIBK.pdf.
[75] Drinkwater, D. (2016) How to get more from your security budget. URL https://www.infoworld.com/article/3152153/how-to-get-more-from-your-security-budget.html.
[76] Stoltzfus, J. (2019) How cloud computing is changing cybersecurity. URL https://www.techopedia.com/how-cloud-computing-is-changing-cybersecurity/2/3394.
[77] Fiala, J. (2015) A Survey of Machine Learning Applications to Cloud Computing. URL https://www.cse.wustl.edu/~jain/cse570-15/ftp/cld_ml/index.html.
[78] Hoadley, D.S. & N.J. Lucas, N.J. AI and national security," 2018, Library of Congress Congressional Research Service, URL https://www.hsdl.org/?abstract&did=810166
[79] Canner, B. (2018) Ping identity releases survey on the perils of enterprise APIs.
URL https://solutionsreview.com/identity-management/ping-identity-releases- survey-on-the-perils-of-enterprise-apis/.
[80] Jason Kent (March 2023) Using ChatGPT to Improve API Security: Open AI & Security , Security Boulevard https://securityboulevard.com/2023/03/using-chatgpt-to-improve-api-security-open-ai-security/)