International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048
Volume 6, Issue 6, pp 108-117, June-2022
https://doi.org/10.47001/IRJIET/2022.606014

Application Programming Interface (API) Security: Cybersecurity Vulnerabilities Due to the Growing Use of APIs in Digital Communications

[1]Alaa Abdul Al Muhsen Hussain Al Zubaidi, [2]Dr. Pro. Florentin Ipate

[1,2]Department of Computer Science, Faculty of Mathematics and Computer Science and ICUB,
[1,2]University of Bucharest, Str. Acadimiei 14, sector 1, 010014, Buchareast, Romania
[1]Department of Computer Information Systems, Faculty of Computer science and Information Technology, University of Al-Qadisiyah, Iraq
[1]E-mail: alaa-abdulalmuhsen.alzubaidi@s.unibuc.ro, alaa.abd@qu.edu.iq, alzalaa94@gmail.com
[2]E-mail: florentin.ipate@ifsoft.ro

***Abstract* - Organizations and consumers must have robust API management and security programmes to ensure they employ the most up-to-date policies to verify that these interactions are sufficiently safe. Technology providers provide API Management solutions to their customers, and API security standards have been established to ensure the security of API transactions. As part of the endeavor to make APIs open and straightforward to deploy for both Business to Business (B2B) and Business to Consumer communications, security requirements must be considered as part of the API management process.**

This paper gathered information in order to determine why APIs are susceptible. It investigated the various perspectives among Customers regarding their own professional experiences with developing private APIs for their organizations. It compared this to the Cyber Security Vendor/Supplier segment, which provides products and services to assist their Customers with API development and security and management. The findings were compared to the findings of the previous research. According to the study's findings, API exploits are typically not identified when they occur, and attitudes on security readiness fluctuate depending on the IT job. During the course of this study, several essential blocking and tackling concepts were discovered that might be used by any business to improve API security management.

Keywords: API, B2B, Cyber Security, Vendor/Supplier segment, IT job.

I. INTRODUCTION

Since they were initially documented in 2005, the rise of publicly available APIs (Application Programming Interfaces) has been exponential (Santos, 2017). API economy refers to the new and rapidly growing area of IT that focuses on APIs. How enterprise business is conducted has primarily been influenced by API web services and increased functionality (Rajaram et al., 2013). An improved definition of APIs is required to keep pace with the rapid advancement of the cloud operating system (Chen et al., 2017). APIs are at the heart of these connections as the Internet of Things progresses from a concept to managing consumer vehicles and kitchen appliances (Siriwardena, 2014). With 29 billion IP-connected devices by 2023, Cisco expects APIs to be the primary means of communication. When mobile apps communicate with web services via APIs, input validation discrepancies are at risk, leading to significant security vulnerabilities (Mendoza, Gu, 2018).

The ProgrammableWeb is the world's most comprehensive resource for publicly available API information. Since its inception, The ProgrammableWeb's Research Center has documented and categorized more than 22,000 public APIs. Companies like Google, Sales force, eBay, and Amazon are among the API providers. Since 2005, the ProgrammableWeb has kept track of API growth, starting with 105. Since then, APIs have evolved from a novelty to a hot topic, with many enterprises now relying on them for critical service functions. APIs have had a significant impact on a wide range of businesses. One hundred five were recorded in 2005, with a minor rise to 2000 in January of the following year. In 2014, there were 12,000 people, while in 2017, there were 17,000 people (Santos, 2017). There are 22,000 people on the list (Berlind et al., 2019). Since private/managed APIs cannot be accurately assessed, the spike in the use of public APIs is an indication that APIs, whether public or private, constitute the backbone of system communications with a strong growth trend. As a result of this study, we can better understand why APIs are susceptible from the standpoints of both customers of cyber security services and the vendors and suppliers who provide them.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 6, Issue 6, pp 108-117, June-2022

https://doi.org/10.47001/IRJIET/2022.606014

Virtualization products have made it possible for many companies to set up server farms in their own data centers or in Private Clouds, which are more secure. There has been an increase in the number of Cyber Security Customers using services from multiple Public Cloud Providers as the technology has matured. A new term, "Hybrid Cloud Environment," refers to a situation in which an organization's IT infrastructure utilizes both private and public cloud services (Edwards et al., 2017). As a result of the hybrid cloud's interplay between private and public cloud services, API communications have become more complex. For this reason and others, it is becoming more common for organizations that rely on internal data centers to move their IT services to Public Cloud Providers, where they can be accessed by multiple entities and across both Private and Public Clouds. For example, if institutional data is stored in a SaaS offering from a Public Cloud Provider, third-party API communications must be secure. Every time a new service is added, the cyber security challenges increase. This is because these connections are between servers, or between services, or even between services and servers (McGrath, Brenner, 2017). Thus, the development of both private and public APIs (managed and open source) to correspond with the REST software architecture design standard is required. RESTful APIs provide specific purposes. DELETE can be used to remove data from a data source using the GET function, the PUT function, the POST function, and the DELETE function. The GET, PUT, POST, and DELETE operations in the RESTful API (Representational State Transfer) standard are incredibly powerful. Aside from this, companies that offer APIs make a concerted effort to maintain APIs that are open and permissive for B2-B and B2C communication (Monahan, 2017). As a result, security standards must be carefully studied and applied thoroughly. API security is in direct conflict with its stated goal of facilitating B2B and B2C communication by making integrations available to everyone, which is the paradox of the Application Programming Interface (API). Customers of B2B and B2C companies' cyber security services suffer due to the APIs' excessive openness, which makes them susceptible (Karhu et al., 2018). There is a visual representation of this relationship in Figure 1.

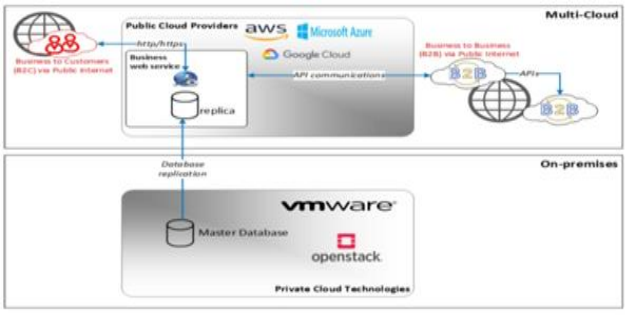

Figure 1

Statement of the Problem and Research Questions

The Open Web Application Security Project is a well-known online community on the subject of web application security. OWASP publishes papers, techniques, documentation, and tools on various security-related topics. The Open Web Application Security Project provides these services to software developers for free (Wichers, Williams, 2018). The Open Web Application Security Project published a report titled "The Top Ten Most Critical Web Application Security Risks," which was last updated in March 2018. Over 100,000 applications and APIs were analyzed to acquire data. While all ten vulnerabilities are indirectly tied to APIs, two of them are directly related. Specifically,

#1; (A3:2017) – Sensitive Data Exposure

Data such as credit card numbers, medical records, and personally identifiable information (PII) is frequently not adequately protected in web apps and APIs (PII). Credit card fraud, identity theft, and other crimes can be perpetrated by stealing or modifying weakly secured data. Without additional protection, such as encryption at rest or in transit, and special safeguards while exchanging sensitive data with the browser," sensitive data can be compromised.

#2: (A9:2017) – Using Components with Known Vulnerabilities

This means that the same privileges are granted to the application's components as they are granted to the application. Data loss or server takeover can occur if a vulnerable component is exploited. Applications and APIs that rely on components known to have security flaws expose themselves to various threats and consequences.

In this study, we collect data from the security community (Cyber Security Customers and Vendors/Suppliers) to understand why APIs are insecure. Furthermore, we study whether there is a difference in the views of Cyber Security Vendors/Suppliers and Cyber Security Customers about API threats and vulnerabilities.

Embedded vulnerabilities can result from today's API architectural standards, or they can result from specific implementation choices made by individual organizations. Organizations need to know if recent API security issues could have been prevented if improved protection standards in authentication, authorization and encryption had been in place at the time.

"*It is straightforward to create a negative impression and bad API and rather difficult to create a good one. Even minor and quite innocent design flaws have a tendency to get magnified*

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 6, Issue 6, pp 108-117, June-2022

https://doi.org/10.47001/IRJIET/2022.606014

out of all proportion because APIs are provided once but are called many times." (Henning, 2009).

As such, the study's research questions are as follows; the fundamental research questions are as follows:

1. Customers and vendors in the security industry and those that provide and manage APIs agree that the security standards are adequate to address any emerging vulnerabilities or threats in APIs used by public and private/managed services?
2. Are customers and vendors/suppliers in the security sector, particularly those in the cyber security industry, convinced that new and better security requirements are needed in public and private/managed API domains?
3. Do Cyber Security Vendors/Suppliers have different views on API threats and vulnerabilities than their customers?
4. Will you be utilizing Micro services or Server less Compute?

II. THE STUDY METHODOLOGY

Using semi-structured interviews: qualitative research may be conducted

Qualitative research techniques vary from quantitative research methods in that they use a different inquiry approach. Qualitative approaches attempt to comprehensively explain a phenomenon and understand how others perceive their own experiences (Creswell, 2009; Merriam, 2009). On the other hand, quantitative approaches are better suited to reducing data to quantifiable variables that may be extrapolated to larger populations or the statistical analysis of cause and effect relationships. Semi-structured interviews are one approach to acquiring qualitative data; they are often used in research. With this approach, the researcher is able to obtain adequate data related to the study's research question while also providing participants with an opportunity to depict their lived experiences adequately. The semi-structured interview approach is an open-ended format in which questions are used as a guide with two intentions: 1) the ability for the researcher to obtain adequate data related to the study's research question and 2) an opportunity for participants to sufficiently depict their lived experiences (Kvale, Brinkmann, 2009). According to Kvale and Brinkmann (2009), semi-structured interviews allow for the participant to relate data spontaneously and richly, where the participant engages in a back and forth conversation, allowing for not only the answering of questions but also the telling of one's own story to take place.

As a result, six one-on-one in-depth interviews were performed with various respondents who qualified as Cyber Security Customers and Vendors/Suppliers based on their position in the information technology business. They were mapped onto the study topics and focused on the areas of interest as listed below:

- According to your assessment, do you believe that the security standards are sufficiently strong to address emerging security risks in public and private/managed API domains and cross-vendor API communications?
- Customers in the cyber security industry only: Do you intend to employ Micro services or Server less Compute in your application?

Cyber Security Customers and Vendors/Suppliers were both represented among the respondents to the in-depth interviews. They were chosen via the writers' own networks based on an availability sample. In Table 1 provides a breakdown of the respondents' demographic characteristics.

Table 1

Title	Industry	Role
Systems Team Leader	Higher Education	Customer: Applications Leader, Database and Integrations
Director of IT Security,	Higher Education	Customer: CISSP, CISM, C CISO, Security Plus
Chief Technology Officer	Cyber Security	Cyber Security Vendor/Supplier
Director, Cloud Enablement	Insurance	Customer: software development leader with cloud and security expertise.
Account Executive	Cyber Security	Cyber Security Vendor/Supplier
Chief Information Security Officer	Healthcare	Customer: leads and implements progressive IT security practices within Healthcare.

As a result of in-depth interviews, the survey instrument for the descriptive research design was developed to collect data to answer the quantitative research questions.

III. RESEARCH DESIGN: DESCRIPTIVE STRUCTURE

An online survey as a descriptive study methodology was used to collect data from 50 eligible respondents. Descriptive research seeks to identify differences in the features of a population sample (Siedlecki, Sandra, 2020).

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 6, Issue 6, pp 108-117, June-2022

https://doi.org/10.47001/IRJIET/2022.606014

Customers and vendors/suppliers who have used Cyber Security goods and services in their capacity as a customer or vendor/supplier were eligible to participate in the survey based on screening criteria for survey participants. The survey data was collected using a sample of available people to participate. Responses were obtained from the authors' networks due to an availability sampling, which included different interactions through social media and telephone recruitment. Qualtrics XM was the online survey solution that was used.

Based on these demographic screening criteria, a sample of three responder profiles was generated,

- Vendors/suppliers in the field of cyber security; 47%
- Customers who purchase cyber security products account for 47% of all customers.
- The remaining 6% of the workforce is comprised of instructors, DevOps, and indirect security tasks.

Figure 2, Depicts a representation of these responder parts as a visual representation.

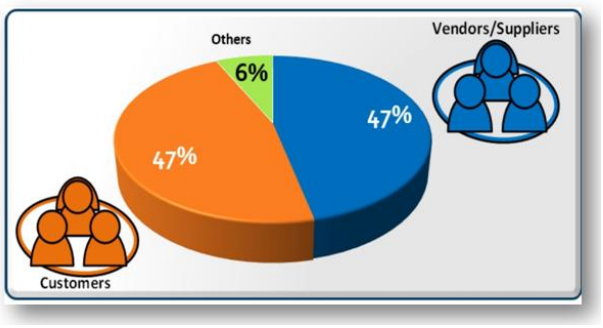

Figure 2

Customers in the cyber security industry and cyber security vendors/suppliers were asked questions corresponding to the study's research questions.

The following questions were put to cyber security vendors and suppliers:

- Do you feel that the security technologies you provide now can handle API connections between vendors?
- What new products or services do you plan to launch that will be ready to support multi-vendor microservices and serverless communications?

Whereas, Customers in the cyber security industry were asked the following questions:

- What are the issues and problems you are now dealing with regarding API security?
- What measures are you taking to protect the API transaction between vendors?

Customers in the Cyber Security industry, as well as vendors and suppliers, were given the following question: -

- In your perspective, are the API security standards currently in place strong enough to remediate security vulnerabilities in the current environment?

Following that, the data were subjected to frequency analysis. In this study, the differences between groups (Cyber Security Customers and Cyber Security Vendors/Suppliers) were investigated utilizing a multivariate approach that included cross-tabulations and the Chi-Square hypothesis test (where the significance level was set at.05.).

IV. FINDINGS

Regarding the present API Standards, the poll's findings revealed that more than two-thirds of all respondents believed that the existing security standards for APIs were insufficiently strong to address the current security risks that API implementations were facing, as seen in Figure 3.

The following hypothesis is based on this finding:

- Ho There is no difference in opinion among Cyber Security Customers vs. Vendors/Suppliers on whether existing API security standards are robust enough to manage current API implementation risks.
- HA Cyber Security Customers vs. Vendors/Suppliers have opposing viewpoints on whether existing API security standards are strong enough to address the current risks affecting API implementations.
- When comparing the findings by IT security job, a Chi-Square Test was used to establish statistical significance. The study discovered evidence of a change in attitude based on IT security jobs. All Cyber Security Customers polled (security professionals who employ security solutions to secure their companies' and customers' data) said "no" to the assertion that existing API security standards are insufficient to address current security threats. Over two-thirds of Vendors/Suppliers (IT executives in organizations that provide security products and services) said the existing API security requirements are sufficient to address current security risks. This suggests that Cyber Security Vendors/Suppliers have a higher level of confidence than Customers. The

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 6, Issue 6, pp 108-117, June-2022

https://doi.org/10.47001/IRJIET/2022.606014

hypothesis test yielded a P-value of—003, indicating that the null hypothesis was rejected with a confidence level of 99.7%. As a result, this data demonstrates that the IT security function has a different mindset viewpoint. Table 2 shows the percentage breakdown.

- As shown in Table 2, "Are established API security standards strong enough to remedy current security threats?"

Table 2

Responses	Cyber Security Vendors/Suppliers	Cyber Security Customers
Yes	67.0 %	0.0 %
No	33.0 %	100.0 %

Pearson Chi-Square Value = 9.000
P value = .003

People who replied "yes" when asked whether the current API requirements are adequate were asked to explain why. Cyber Security Vendors/Suppliers accounted for 100 percent of those who said "yes" when asked whether the existing API security requirements are strong enough to combat current security threats. Vendor/Supplier replies tended to emphasize the need to focus on the application's security architecture so that the API may benefit from and use those capabilities. An example from the Vendor/Supplier group illustrates the point:

"*An API is only the interaction with the application. The focus should be on developing the application properly. If we took the proper time to develop software with a security focus from step one, we would not need to strengthen API. OWASP would not exist."*

All customers and 33% of vendors/suppliers that replied "no" when asked whether API standards are adequate commented on their reasoning. As a result of customer feedback, it became clear that existing API security requirements are woefully inadequate to protect against today's security threats. This subject is shown with a statement from the Customer sector, which reads as follows:

"*API-based attacks exploit API design flaws that are specific to each API and are therefore unique in nature. Other attacks involve brute force attacks on the login or the theft of tokens or credentials which give access to the API service and data as a normal user."*

Other questions included, "As a Cyber Security Vendor or Supplier, will your organization introduce any new products or services in your roadmap to addressing emerging API security vulnerabilities?" Many Cyber Security Vendors/Suppliers believe that new solutions and services will solve API vulnerabilities. More than six in seven respondents (67 percent) identified themselves as Cyber Security Vendors/Suppliers (particularly organizations that develop security software and provide security services). However, just 33 percent of the Vendors/Suppliers said no. Figure 4 depicts the breakdown in percentage terms.

- Are new API vulnerabilities being addressed by security vendors and suppliers' product and service roadmaps?

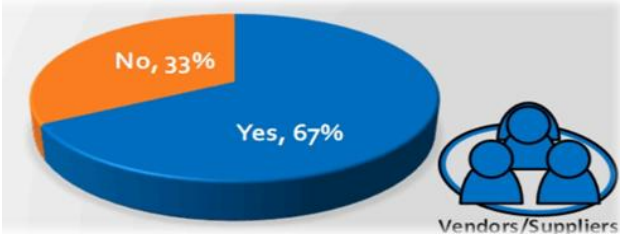

Figure 4

Additionally, individuals who described themselves as Cyber Security Customers were asked, "Are you going to employ Microservices or Serverless Compute?" Microservices and Serverless computing were on the minds of almost two-thirds of those surveyed as potential implementations in their organizations. Security professionals who utilize security solutions to secure their customers' data and their entities replied "yes" at 65%, while the remainder of the Cyber Security Customers responded "no" at 35 percent. Figure 5 shows the percentage breakdown.

Figure 5. Will you be using Microservices or Serverless Computing?

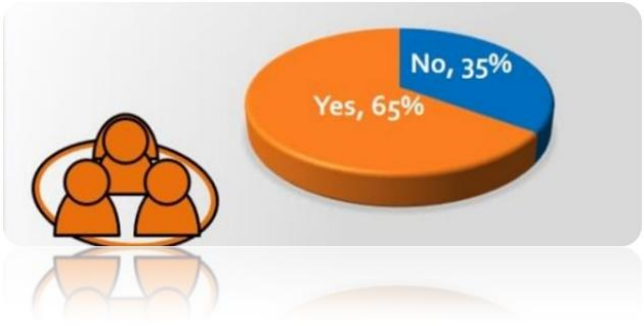

Figure 5

V. CONCLUSIONS AND IMPLICATIONS

The major problem with API vulnerabilities emerges when the underlying application's security architecture is poor. API security is only as secure as it is intended to be. There are several real-world instances of "Works as Designed" (WAD), in which firms were exposed to data breaches due to inadequate implementations of API security architecture. Hackers did not create these vulnerabilities since

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 6, Issue 6, pp 108-117, June-2022

https://doi.org/10.47001/IRJIET/2022.606014

they did not require breaching firewalls or deciphering sophisticated encryption techniques. Due to the inadequate implementation of API security, anybody may walk in and harvest data that they should not have had access to in the first place. As a result, the newly coined term "leaky API" is correctly called (Spring, 2018).

According to the most current revisions, it is now believed that over 87 million Facebook Cyber Security customers had their private information exposed due to an API initially built for a mobile application (Romano, 2018). Cambridge Analytica (CA), a data analytics business that collaborated with political election campaigns, discovered the "leaky API" and collected allegedly private data from 87 million Facebook user profiles.

While Cambridge Analytica was operating without authorization between 2013 and 2015, they stole Facebook users' profile data, which they then used to fill their own marketing database with information about each user's specific likes and interests. They then developed a personality profile for each user, allowing them to be more successfully targeted for certain political campaigns. According to Feiner and Rodriguez (2019), the Federal Trade Commission penalised Facebook $5 billion for misusing data. Additionally, Cambridge Analytica suspended operations and filed for bankruptcy (Confessore, Rosenberg, 2018).

Cambridge Analytica was able to get this information in the first place due to a flaw in Facebook's private API, which enabled third-party developers to acquire data not just from users of their applications but also from everyone in those users' Facebook friends' network. Access to this information was granted with the condition that the information not be shared, promoted, or sold, a condition that CA swiftly violated (Romano, 2018).

It is wrong to refer to CA's data collection as a "hack" or to say that it constitutes a significant breach of Facebook rules.

This is due to the fact that the information gathered by the corporation was information that Facebook had freely provided and that was initially meant to be accessed solely by mobile developers. Technically speaking, anybody who utilised third-party Facebook applications might have discovered that they were enabling such applications to access information from their friends' accounts by checking their Facebook settings. An official Facebook spokesman confirmed to the New York Times that "no systems were compromised, and no passwords or critical bits of information were taken or hacked," according to the spokesperson. (Rosenberg and colleagues, 2018).

As a result, the API-level Security Certification of Android Applications (ASCAA) group discovered that, out of 200 evaluated API applications, 12.5 percent failed their sample criteria (Pei et al., 2017). The ASCAA discovered evidence that the failing apps were either over privileged or did not disclose permissions.

Another example of the "Works as Designed" paradigm in relation to API security was the T-Mobile data incident (Spring, 2018). In August of 2018, T-Mobile placed an unsecured, unprotected API on their website, exposing the personal information of 2.3 million Cyber Security Customers.

A hacker was able to test for genuine customer phone numbers by directly changing the end of the URL (Uniform Resource Locator) string with a new phone number in the web browser, and the website replied with personal information as a result. Figure 6 shows an example of the URL modification that was employed in this instance.

Figure 6. T-Mobile API – URL string;

The API retrieved the following sensitive customer data for the number 123-456-7890:

- Email address
- Name
- Billing Account Number
- International Mobile Subscriber Identity Number (IMSI)
- Other phone numbers associated with the account
- Other phone numbers associated with the account (e.g., friends and family).

Another example given by Netflix staff indicated that some API-based interactions increased the attack surface for their microservices, which was previously unknown. A Netflix security engineer conducted a test on the company's streaming infrastructure in front of a crowd of hundreds of employees at the 2017 DefCon Security Conference.In the end, he managed to reduce the site down to nothing. Because Netflix was susceptible to a strange sort of Distributed Denial of Service (DDOS) attack, he and a Netflix cloud security engineer demonstrated that the streaming service was vulnerable.

Realizing this new vulnerability prompted Netflix to take steps to defend the service and the rest of the Internet against this new danger. The assumption was that a few basic queries may produce a large number of backend requests, similar to how a badly written structured query language (SQL) script on a database could generate a large number of backend requests. This query systematically examines the table list inefficiently,

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 6, Issue 6, pp 108-117, June-2022

https://doi.org/10.47001/IRJIET/2022.606014

clogging up all database connections available for other traffic. Since the incoming client activity was below the API gateway's rate restrictions, the crucial protective mechanism for API traffic in the design allowed the bad request to pass (Newman, 2017). Rate restrictions, in which the API gateway specifies the maximum number of times an API may be used, may be an effective technique to secure an API. However, in this scenario, the requests were sent at a rate that exceeded the rate restriction setting.

After analyzing the survey results and segmenting them by Cyber Security Customers against Vendors/Suppliers, it is clear that there is a considerable gap in preparedness to handle current security threats between Vendors/Suppliers and Cyber Security Customers of security products and services. Vendors/Suppliers should pay more attention to Cyber Security Customers' opinions in order to infuse suggestions in the field with "customer voice."Customers perceive that existing API security standards are insufficiently strong to prevent current security risks, even though they are heavily investigating emerging technologies such as Microservices or Serverless Compute.

A web application's security is not the same as an API's security. JSON Web Token (JWT) and Oauth2 are two examples of stateless API authentication (Stannard, 2015). It is simple for a hacker to keep attempting various URL string variants to attack an unprotected, unsecured API vulnerability since the websites that host the APIs do not maintain session data. Using stateful transactions, which create a session cookie (a tracking key that is valid just once for that particular session), is a more secure transaction practiced by web applications. In order to keep a particular customer's communication with the website private, the session cookie is used. Because session cookies can't be reused, a new one is produced each time a consumer logs in. Cross-site scripting (XSS) and SQL injection are two of the most common threats to traditional online security. API security necessitates additional safeguards because of the ease with which hackers may get access to data through stateless transactions, which is inherent with APIs.

API security is more difficult since it occurs at layer 7 of the OSI model (Mitchell, 2019). Detection of fraudulent usage through API gateways is only beginning to develop at layer 7. Due to the fact that an API gateway (Netflix example) has a lower rate limit than a DDOS (Advanced Distributed Denial of Service), a hacker can access data that is not secured. When a DDOS assault is launched, the gateway and, by extension, the website it protects are inundated with a massive influx of connections. Protecting a website with a properly configured Intrusion Prevention System (IPS) is typically as simple as looking for suspicious source IP addresses and then only allowing traffic from those IP addresses through. Problems arise because IPS systems function on layers four and seven, where API traffic happens, rather than layer seven.

As a result, API security cannot be overstated in today's IT world. Currently, everything has a digital counterpart (Harguindeguy, 2017). Using a bank as an example, the vast majority of everyday transactions are now carried out by smartphones. Facebook, Instagram, and iCloud are all places where you may save your photos. The following examples highlight the significance of API security:

- Growth in Public APIs is increasing exponentially, as stated by The Programmable Web (Santos, 2017).
- Hackers are constantly looking for the quickest and easiest route in. Unprotected API services are potential targets for hackers and other criminals (Wheeler, 2018).
- Several of the participants in this survey mentioned new compute services like Microservices and Serverless computation. AWS Lambda and other serverless compute services, such as the Internet of Things (IoT), are examples of the growing number of attack surfaces that hackers may exploit.
- By 2021, Interconnection bandwidth is expected to increase ten times the current Internet traffic, according to the Global Equinix Interconnection Index (Equinix, 2019). A majority of API vulnerabilities go undetected while they are taking place. An organization's security information and event management (SIEM) approach must be powerful enough to discover API vulnerability in order to effectively repair the issue (Harguindeguy, 2017).

For security to work at its best for any company, it must be properly nurtured. Based on this study, the following are some basic blocking and tackling principles that may assist any firm in enhancing API security management.

1) Start by creating an API inventory, then use a SIEM (Security Information and Event Management) system to record API traffic (Harguindeguy, 2017).

- Check out the APIs that are available in your company. When businesses install new software, specific APIs are automatically installed.
- Don't put your internal API names in public DNS, either. Avoid disclosing confidential information to the outside world.
- In the case that your company uses an API Gateway, ensure that all events are appropriately logged.

2) Always maintain security in mind when developing a system. If security is not integrated into the design process from the start, your security strategy will perform as an

ISSN (online): 2581-3048
Volume 6, Issue 6, pp 108-117, June-2022
https://doi.org/10.47001/IRJIET/2022.606014

afterthought in how it was designed. Security is frequently considered during integration or deployment, which is far too late in the development process (Siriwardena, 2014).

3) Make use of API management techniques. Beyond just providing business functionality, manage the establishment, publishing, deprecation, and retirement of your APIs. It is critical to maintaining proper documentation (Siriwardena, 2014).

4) If your business is ready to embrace Agile operations, go beyond improving Development and Operations (DevOps) procedures and include Development, Security, and Operations (DevSecOps) processes (George, 2018). Several instances of DevSecOps include the following:

- Continuous Integration/Continuous Delivery (CICD) and code repository/code review - Antivirus scanning
- Automated code distribution with rollback.
- Automated Continuous Configuration (CCA).
- Automate each repeatable and scriptable procedure you come across.

5) Use standard authentication rather than basic authentication (Salem, Mazalevskis, 2017), such as the following:

- JWT (JSON Web Token)
- Oauth2
- Username /password is not enough.
- When possible, end-user authentication rather than API keys or Client ID/Client secret.

6) Limit access requests (throttling) and use Hypertext Transport Protocol Secure (HTTPS) server-side and HTTP Strict Transport Security (HSTS) headers with Secure Sockets Layer (SSL) (Salem, Mazalevskis, 2017).

7) When it comes to input, utilize the appropriate HTTP methods such as GET, POST, PUT, and DELETE and validate the content (Salem, Mazalevskis, 2017).

8) Send X-Content and X-Frame options for output and do not return sensitive data (Salem, Mazalevskis, 2017).

9) Processing (Salem and Mazalevskis, 2017):

- I avoid user-owned resource IDs and auto-incremented IDs for endpoint protection; instead, I use Universally Unique Identifiers (UUID).
- Use End-To-End TLS (version 1.3).

The descriptive research design used in this study was the Qualtrics XM Online survey tool. It was done among qualified respondents in order to have a better understanding of the security concerns surrounding APIs. The data collected and statistical analysis performed aided in determining how respondents, as CyberSecurity Customers and Vendors/Suppliers, see API vulnerabilities differently.

The sample size provided a statistically significant result when comparing Cyber Security Customers and Vendors/Suppliers. Future studies will also seek views from a larger audience in order to develop a more global viewpoint. Additionally, future research may include a qualitative component by conducting additional one-on-one in-depth interviews (IDIs) to go deeper into the insights uncovered in this study by applying extensive probing approaches to understand managerial better practices.

REFERENCES

[1] Berlind, D., Santos, W., Sundstrom, K. (2019, June). The ProgrammableWeb Research Center. Retrieved from https://www.programmableweb.com/api-research.

[2] Chen, Z., Chen, K., Jiang, J., Zhang, L., Wu, S. (2017). Evolution of Cloud Operating System: From Technology to Ecosystem. Journal of Computer Science and Technology; Beijing Vol. 32, Iss. 2, 224-241. DOI:10.1007/s11390-017-1717-z

[3] Confessore, N., Rosenberg, M. (2018, May). Cambridge Analytica to File for Bankruptcy After Misuse of Facebook Data. Retrieved from https://www.nytimes.com/2018/05/02/us/politics/cambridge-analytica-shutdown.Html?searchResultPosition=2

[4] Creswell, J. W. (2009). Research design: Qualitative, quantitative, and mixed methods approach (3rd ed.). Thousand Oaks, CA: Sage.

[5] Edwards, M., Gawade, P., Leung, J., McDonald, B., Schalk, K., Scott, K., Van Order, B., Woodward, S. (2017, July). Practical Guide to Cloud Management Platforms. Cloud Standards Customer Council. Retrieved from https://www.omg.org/cloud/deliverables/CSCC-Practical-Guide-to-Cloud-Management-Platforms.pdf.

[6] Equinix (2019, October). Global Interconnection Index, Volume 3. Equinix, Inc. Retrieved from https://www.equinix.com/global-interconnection-index-gxi-report

[7] Feiner, L., Rodriguez, S. (2019, July). FTC slaps Facebook with a record $5 billion exemplaries, and orders privacy oversight. Retrieved from https://www.cnbc.com/2019/07/24/facebook-to-pay-5-billion-for-privacy-lapses-ftc-announces.html

[8] George, T. (2018, June). The Next Big Cyber-Attack Vector: APIs. SecurityWeek. Retrieved from https://www.securityweek.com/next-big-cyber-attack-vector-apis

ISSN (online): 2581-3048
Volume 6, Issue 6, pp 108-117, June-2022
https://doi.org/10.47001/IRJIET/2022.606014

[9] Harguindeguy, B. (2017, Mar). AI-powered API security with Bernard Harguindeguy of Elastic Beam. Pentester Academy TV. Retrieved from https://www.youtube.com/watch?v=R9QAJri8jAU&t=42s

[10] Henning. M. (2009, May). API design matters. Commun. ACM 52, 5, 46–56. Retrieved from https://doi-org.avoserv2.library.fordham.edu/10.1145/1506409.1506424

[11] Kvale, S., Brinkmann, S. (2009). Interviews: Learning the Craft of Qualitative Research Interviewing. Second Edition; Sage.

[12] Karhu, K., Gustafsson, R., Lyytinenc, K. (2018). Exploiting and Defending Open Digital Platforms with Boundary Resources: Android's Five Platform Forks. Information Systems Research SYSTEMS RESEARCH, Vol. 29, No. 2. ISSN 1047-7047 (print), ISSN 1526-5536 (online).

[13] Malinverno, P., O'Neill, M. (2016). Magic Quadrant for Full Life Cycle API Management. The Gartner Group. Document ID: G00277632.

[14] McGrath, G, Brenner, P. (2017). Serverless Computing: Design, Implementation, and Performance. 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW), Atlanta, GA, 2017, pp. 405-410.

[15] Mendoza, A., Gu, G., (2018). Mobile Application Web API Reconnaissance Web-to-Mobile Inconsistencies and Vulnerabilities. IEEE Symposium on Security and Privacy.

[16] Merriam, S. B. (2009) Qualitative research: A guide to design and implementation. San Francisco, CA: Jossey-Bass.

[17] Mitchell, B (2019, August). The Layers of the OSI Model Illustrated. Retrieved from https://www.lifewire.com/layers-of-the-osi-model-illustrated-818017

[18] Monahan, D., (2017, April). Why There Is No API Security. Radware Blog. Retrieved from https://blog.radware.com/security/2017/04/no-api-security/

[19] Newman, L.H. (2017 July). How Netflix Ddos'd Itself to Help Protect The Entire Internet. Wired. Retrieved from https://www.wired.com/story/netflix-ddos-attack

[20] Niinioja, M., Moilanen, J. (2018, May). Do you Categorize your APIs? Osaango. Retrieved from https://www.osaango.com/blog/why-should-you-categorize-your-apis

[21] Rajaram, B., Babu, C., Kishore, C., Kumar R, (2013). API based security solutions for communication among web services, 2013 Fifth International Conference on Advanced Computing (ICoAC), Chennai, pp. 571-575.

[22] Romano, A., (2018, March). The Facebook data breach wasn't a hack. It was a wake-up call. Vox. Retrieved from https://www.vox.com/2018/3/20/17138756/facebook-data-breach-cambridge-analytica-explained

[23] Rosenberg, M., Confessore, N., Cadwalladr, C. (2018, March). How Trump Consultants Exploited the Facebook Data of Millions. Retrieved from https://www.nytimes.com/2018/03/17/us/politics/cambridge-analytica-trump-campaign.html

[24] Salem, E., Mazalevskis, C., (2017, July). API-Security-Checklist. Shieldfy. Retrieved from https://github.com/shieldfy/API-Security-Checklist

[25] Santos, W. (2017, March). API Directory Eclipses 17,000 as API Economy Continues Surge. The ProgrammableWeb. Retrieved from https://www.programmableweb.com/news/programmableweb-api-directory-eclipses-17000-api-economy-continues-surge/research/2017/03/13

[26] Shoemaker, A., Lambert, K. (2018, January). API Endpoints: The New DDoS Attack Vector for Cybercriminals. BrightTALK. Retrieved from https://www.brighttalk.com/webcast/14611/296621/api-endpoints-the-new-ddos-attack-vector-for-cybercriminals

[27] Siedlecki, Sandra L. (2020, January/February). Understanding DescriptiveResearch Designs and Methods. Clinical Nurse Specialist. Retrieved from https://journals.lww.com/cns-journal/Fulltext/2020/01000/Understanding_Descriptive_Research_Designs_and.4.aspx

[28] Siriwardena, P (2014). Advanced API Security – Securing APIs with OAuth 2,0, OpenID Connect, JWS, and JWE. Apress ISBN 978-1-4302-6818-5e-ISBN 978-1-4302-6817-8.

[29] Spring, T (2018, August). T-Mobile Alerts 2.3 Million Cyber Security Customers of Data Breach Tied to Leaky API. ThreatPost. Retrieved from https://threatpost.com/t-mobile-alerts-2-3-million-Cyber Security Customers-of-data-breach-tied-to-leaky-API/136896

[30] Stannard, A. (2015, August). The Inevitable Rise of the Stateful Web Application. Petabridge. Retrieved from https://petabridge.com/blog/stateful-web-applications

[31] W. Pei, J. Li, H. Li, H. Gao and P. Wang (2017). ASCAA: API-level security certification of android applications, in IET Software, vol. 11, no. 2, pp. 55-63.

[32] Wheeler, C., (2018, February). Three New Attack Vectors That Will Be Born Out of IoT. Liquid Web.

ISSN (online): 2581-3048
Volume 6, Issue 6, pp 108-117, June-2022
https://doi.org/10.47001/IRJIET/2022.606014

Retrieved from https://www.liquidweb.com/blog/three-new-attack-vectors-will-born-iot/

[33] Wichers, D., Williams, J. (2018, March). Top Ten Most Critical Web Application Security Risks. The OWASP Foundation. Retrieved from https://owasp.org/www-project-top-ten/

Citation of this Article:

Alaa Abdul Al Muhsen Hussain Al Zubaidi, Dr. Pro. Florentin Ipate, "Application Programming Interface (API) Security: Cybersecurity Vulnerabilities Due to the Growing Use of APIs in Digital Communications" Published in *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 6, Issue 6, pp 108-117, June 2022. Article DOI https://doi.org/10.47001/IRJIET/2022.606014

****

