Journal of International Technology and Information Management

Volume 29 | Issue 3 Article 2

2021

The Future of API (Application Programming Interface) Security: The Adoption of APIs for Digital Communications and the Implications for Cyber Security Vulnerabilities

Alison Munsch PhD
Iona College, amunsch1@earthlink.net

Peter Munsch MBA
Fordham University, munschp@gmail.com

Recommended Citation

Munsch, Alison PhD and Munsch, Peter MBA (2021) "The Future of API (Application Programming Interface) Security: The Adoption of APIs for Digital Communications and the Implications for Cyber Security Vulnerabilities," *Journal of International Technology and Information Management*: Vol. 29: Iss. 3, Article 2.
DOI: https://doi.org/10.58729/1941-6679.1454
Available at: https://scholarworks.lib.csusb.edu/jitim/vol29/iss3/2

The Future of API Security: The Adoption of APIs for Digital Communications and the Implications for Cyber Security Vulnerabilities

Alison Munsch PhD *(Iona College)*
Peter Munsch MBA (*Fordham University)*

ABSTRACT

Businesses and consumers need to have a robust Application Programming Interface (API) management and security program in place to ensure they are using the most updated policies to certify that these transactions are adequately secure. Technology vendors do provide API Management tools for Customers, and there are established API security standards for securing API transactions. Given the effort to keep APIs open and easy to implement for Business to Business (B2B) and Business to Consumer (B2C) communications, security standards must be part of API management.
This research gathered data to investigate why APIs are vulnerable. The research explored the different perspectives among Customers with regards to their own professional experiences with developing private APIs for their organizations and compared it to the Cyber Security Vendor/Supplier segment that offer products and services to assist their Customers with API development, security, and management. The research found that API exploits are usually not detected while they are occurring and perspectives about security readiness are different by IT role. Some basic blocking and tackling fundamentals that can help any organization improve API security management are identified by this research.

Keywords: application programming interface; api; security; software as a service; saas; hybrid cloud; open web application security project; owasp

INTRODUCTION

The growth of publicly available Application Programming Interfaces (APIs) have been growing exponentially since they were first chronicled in 2005
(Santos, 2017). This new and flourishing domain of Information Technology (IT) is referred to as the "API Economy". The popularity of API web services and the

ISSN: 1941-6679-On-line Copy

additional facility they provide have primarily influenced how enterprise business is conducted (Rajaram, et. al., 2013). As the Cloud Operating System evolves, APIs must be better defined (Chen, et. al., 2017). As the Internet of Things evolves from a concept to literally controlling consumer vehicles and kitchen appliances, APIs are at the heart of these communications (Siriwardena, 2014). Cisco estimates that by the year 2023, there will be 29 billion devices connected via IP networks, primarily communicating via APIs. Furthermore, the diverse nature of mobile applications communicating with Web applications via APIs can cause input validation inconsistencies, thus leading to serious security issues (Mendoza, Gu, 2018).

A community organization called The ProgrammableWeb is the world's leading source of information regarding publicly available APIs. With the largest API directory on the Web, The ProgrammableWeb's Research Center has documented and categorized over 22,000 public APIs to date. These API providers come from companies such as Google, Salesforce, eBay and Amazon. The ProgrammableWeb has tracked API growth since 2005, starting with a count of 105. They have notated the growth from a curiosity to a trend, to where APIs are providing core service functionality for many businesses. The value that APIs have contributed to countless organizations is undeniable. They have shown a starting count of 105 in 2005 with a slight slope to a count of 2000 in January 2010. The numbers immediately spike to 12,000 in 2014 and surge past 17,000 in 2017 (Santos, 2017). The count as of June 2019 is 22,000 (Berlind, et. al., 2019). Since, private/managed APIs cannot be adequately measured (Niinioja, Moilanen, 2018), the surge in the use of public APIs is a signal that APIs, whether they be public or private, are the backbone of systems communications with a strong growth trend. In this research, we gather data to investigate why APIs are vulnerable based on the security community perspectives of Cyber Security Customers and Vendors/Suppliers.

With the adoption of virtualization products, many organizations have established server farms in their own data centers on-premises, or Private Clouds. As the services provided by Public Cloud Providers have matured, more Cyber Security Customers have also adopted services in this space, and sometimes from multiple Cloud Providers. The term "Hybrid Cloud Environment" has recently emerged, where organizations have both private and various public cloud services in their IT portfolio (Edwards, et. al., 2017). The interaction between the components of the Hybrid cloud, specifically Private Cloud and Public Cloud services, further complicates the transference of data via API communications. Furthermore, as more companies move their IT services out of internal data centers to Public Cloud Providers, the potential requirements for essential institutional data to be accessible from multiple entities and across Private Clouds and Public Clouds become more

ISSN: 1941-6679-On-line Copy

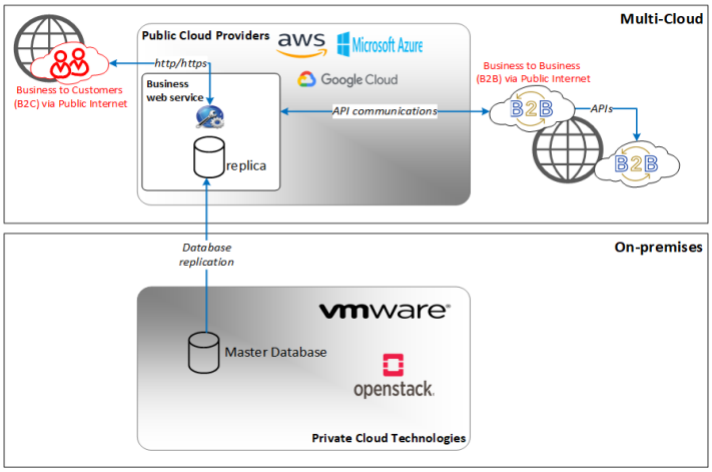
.

prevalent. If that institutional data resides at a Public Cloud Provider Software as a Service (SaaS) offering as an example, there is a need to establish secure API communications between third parties. Since these connections are server-to-server, or service-to-service and very soon serverless-to-serverless (McGrath, Brenner, 2017), the cyber-security challenges increase with every new service offering.

As such, the development of APIs (both private/managed and publicly accessible/open source) to conform to the software architectural design standard called Representational State Transfer (REST) is needed. RESTful APIs have particular functions. The GET function can retrieve data, the PUT function can modify existing data, the POST function can create new data, and DELETE can remove data from the data source. The RESTful API (Representational State Transfer) standard encompasses a lot of power within the GET, PUT, POST, and DELETE functions. Also, entities that provide APIs as part of their service make a conscious effort to keep APIs open and relaxed for Business to Business (B2B) and Business to Consumer (B2C) communications (Monahan, 2017). As a result, security standards should be strongly considered and implemented correctly. There lies the paradox of the Application Programming Interface (API); the essence of the API is to further communications between B2B and B2C by making integrations open and accessible, and security runs directly counter to that effort. B2B and B2C efforts to keep their APIs open to provide value to their Cyber Security Customers frequently open them too wide, leaving them vulnerable (Karhu, et. al., 2018). A visualization of this relationship is illustrated in Figure 1.

Figure 1.

Hybrid Cloud Environment; Multi-cloud Providers and Private Cloud Technologies



.

Source: Survey Results: The Future of API (Application Programming Interface) Security: The Adoption of APIs for Digital Communications and the Implications for Cyber Security Vulnerabilities

PROBLEM STATEMENT AND RESEARCH QUESTIONS

The Open Web Application Security Project (OWASP) is an online community that has become prominent in the field of Web application security. OWASP produces related articles, methodologies, documentation, and tools. OWASP provides these services free to software developers (Wichers, Williams, 2018). The Open Web Application Security Project (OWASP) creates a Top Ten Most Critical Web Application Security Risks report that was most recently updated in March 2018. The data was collected from over 100,000 applications and APIs. Even though all ten noted vulnerabilities relate to APIs indirectly, there were two that are directly related to APIs. Specifically,

> **#1**; (A3:2017) – Sensitive Data Exposure
>
> "*Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and Personally Identifiable Information (PII). Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser."*
>
> **#2**: (A9:2017) – Using Components with Known Vulnerabilities
>
> "*Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts."*

In this research, we gather data to investigate why APIs are vulnerable based on the security community perspectives (Cyber Security Customer and Vendors/Suppliers). Furthermore, we investigate if there is a difference in attitudes in terms of the API threats and vulnerabilities between Cyber Security Vendors/Suppliers and Cyber Security Customers.

.

Inherent vulnerabilities could be part of the design of API architectural standards that are in place today, or it could be more related to how organizations implement APIs within their environments. With the proliferation of APIs in the IT industry today, organizations need to understand if recent API security incidents could have been prevented with new protection standards in authentication, authorization, and encryption.

"*It is very easy to create a bad API and rather difficult to create a good one. Even minor and quite innocent design flaws have a tendency to get magnified out of all proportion because APIs are provided once but are called many times.*" (Henning, 2009).

As such the research questions to be addressed in this study are as follows,

The primary research question is:

Is the security community including Cyber Security Customers, and Vendors/Suppliers of the opinion that security standards currently in place are robust enough to remediate new security threats in public and private/managed API domains and cross-vendor API communications?

The second research question is:

Does the security community including Cyber Security Customers, and Vendors/Suppliers of the opinion that there is a need to develop new and improved security standards in public and private/managed API domains and to secure cross-vendor API communications?

The third research question is:

Is there is a difference in attitudes in terms of the API threats and vulnerabilities between Cyber Security Vendors/Suppliers and Cyber Security Customers?

The fourth research question was directed to Cyber Security Customers only:

Are you planning to use Microservices or Serverless Compute?

METHODOLOGY

Qualitative Research, Semi-Structured Interviews

Qualitative methods of research differ from quantitative methods in their means of inquiry. Qualitative methods seek to describe a phenomenon in a rich and holistic manner and to understand how people interpret their experiences (Creswell, 2009; Merriam, 2009). Whereas, quantitative methods are more suited to reducing data to measurable variables that can be generalized to larger populations or statistically measuring cause and effect. One method of obtaining qualitative data is through the use of semi-structured interviews. 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 adequately obtain data related to the study's research question and 2) an opportunity for the participants to sufficiently depict their lived experiences (Kvale, Brinkmann, 2009). Kvale and Brinkmann (2009) noted that semi-structured interviews allow for the participant to relate data in a spontaneous and rich manner where the participant engages in a back and forth conversation allowing for not merely the answering of questions, but the telling of one's story.

As such, six one-on-one in-depth interviews were conducted with various respondents that qualified as Cyber Security Customers and Vendors/Suppliers in terms of their role in the IT industry. The unstructured questions were mapped to the research questions and examined the following areas:

In your opinion, do you think that security standards currently in place are robust enough to remediate new security threats in public and private/managed API domains and cross-vendor API communications?

In your opinion do you feel there is a need to develop new and improved security standards in public and private/managed API domains and to secure cross-vendor API communications?

For Cyber Security Customers only: Are you planning to use Microservices or Serverless Compute?

The respondents for the in-depth interviews included both Cyber Security Customers and Vendors/Suppliers. They were selected through an availability sample through the authors' networks. The respondent's demographics are detailed in the following Table 1.

.

Table 1. In-depth Interview Respondent Role

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

Source: Survey Results: The Future of API (Application Programming Interface) Security: The Adoption of APIs for Digital Communications and the Implications for Cyber Security Vulnerabilities

The qualitative insights garnered from the in-depth interviews informed the development of the survey tool for the descriptive research design used to capture the data to address the research questions quantitatively.

Descriptive Research Design

A descriptive research design in the form of an online survey was conducted among 50 qualified respondents. A descriptive study intends to look for variations in characteristics within a sample for a given population (Siedlecki, Sandra, 2020).

The respondents were qualified with screening criteria for survey participants to be information systems business professionals with experience using Cyber Security products and services in their role as a Customer or Vendor/Supplier. An availability sample was used for the survey data collection. As such, respondents were recruited from the authors' networks to participate in the survey through an availability sample using various communications via social media, along with telephone recruiting. The online survey system used was Qualtrics XM.

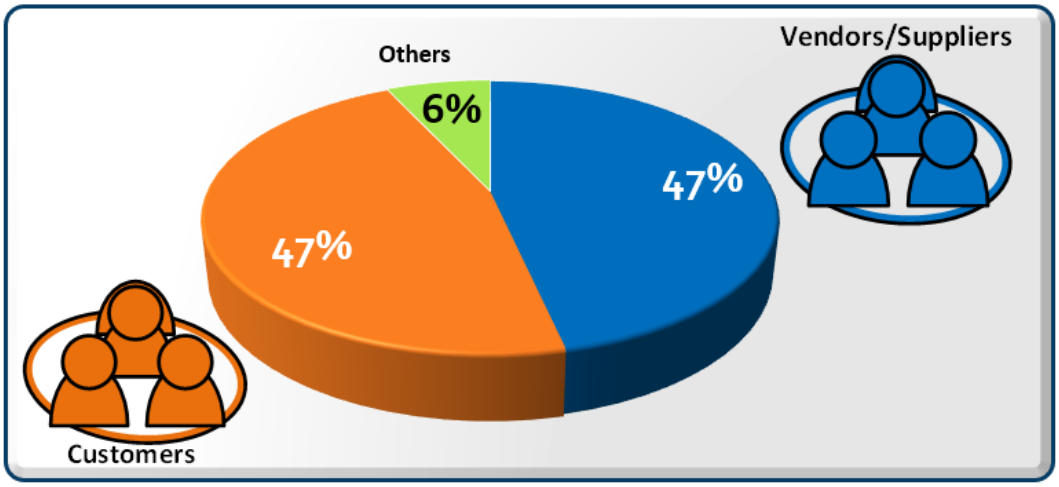
This screening criteria of the population produced a sample of three respondent profiles;

ISSN: 1941-6679-On-line Copy

.

- Cyber Security Vendors/Suppliers; 47%,
- Cyber Security Customers; 47%,
- Others; Instructors, DevOps, and indirect security responsibilities; 6%.
  A visualization of these respondent segments is indicated in Figure 2.

Figure 2.



Source: Survey Results: The Future of API (Application Programming Interface) Security: The Adoption of APIs for Digital Communications and the Implications for Cyber Security Vulnerabilities

Questions were directed to Cyber Security Customers and Cyber Security Vendors/Suppliers mapped to the study's research questions.

Cyber Security Vendors/Suppliers were asked:

- Do you believe the security products you provide today can address cross-vendor API communications?
- Do you have new products or services that will be ready for multi-vendor microservices and serverless communications?

Cyber Security Customers were asked:

-What are the threats/challenges you face with API security? How are you securing the cross-vendor API transaction?

.

Both Cyber Security Customers and Vendors/suppliers were asked:

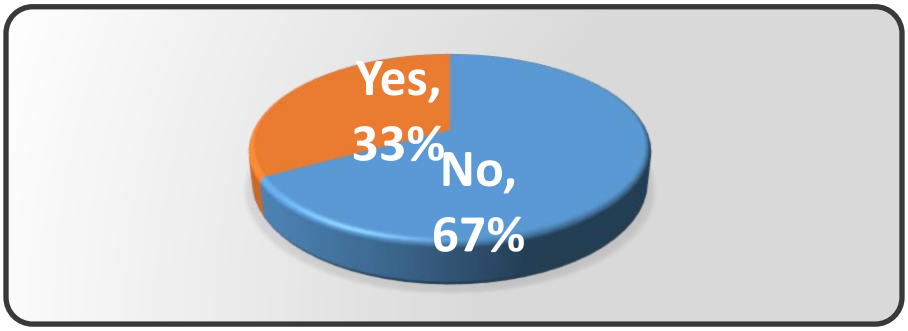
-In your opinion, are the API security standards in place robust enough to remediate security threats in the current environment?

The data was then analyzed using a frequency analysis. The differences between groups (Cyber Security Customers and Vendors/Suppliers) were explored using a multivariate technique using cross-tabulations and the Chi-Square hypothesis test (where α was set to .05).

FINDINGS

Regarding the current API Standards, the analysis of the survey results found that over two thirds of all respondents felt that that the existing security standards for APIs are not robust enough to remediate the current security threats facing API implementations. as illustrated in Figure 3.

Figure 3. Are the current API security standards robust enough to remediate current security threats?



Source: Survey Results: The Future of API (Application Programming Interface) Security: The Adoption of APIs for Digital Communications and the Implications for Cyber Security Vulnerabilities

This finding leads to the following hypothesis:

H_o There is no difference in perspective as to whether the current security standards for APIs are robust enough to handle the current threats facing API implementations, Cyber Security Customers versus Vendors/Suppliers.

.

H_A There is a difference in perspective as to whether the current security standards for APIs are robust enough to handle the current threats facing API implementations, Cyber Security Customers versus Vendors/Suppliers.

A Chi-Square Test was performed to confirm statistical significance comparing the results by IT security role. The test found evidence that there is a difference in attitude by IT security role. Specifically, all Cyber Security Customers surveyed; (security professionals that use security products to protect the data of their entities and clients) responded "no" that the current API security standards are not robust enough to remediate current security threats. Over two thirds of Vendors/Suppliers; (IT executives in companies that produce security software and security services) responded "yes" that the current API security standards are robust enough to remediate current security threats. This indicates that Cyber Security Vendors/Suppliers are more optimistic than Customers. The hypothesis test generated a P-value of .003, which led to the rejection of the null hypothesis with a 99.7% level of confidence. As such, this finding provides evidence that there is a difference in attitude perspectives by the IT security role. The percentage split can be seen in Table 2.

Table 2: "Are the current API security standards are robust enough to remediate current security threats?"

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

Source: Survey Results: The Future of API (Application Programming Interface) Security: The Adoption of APIs for Digital Communications and the Implications for Cyber Security Vulnerabilities

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

Respondents who answered "yes" when asked if the API standards in place today are sufficient were asked to elaborate on their response. All of the respondents who answered "yes" (that the current API security standards are not robust enough to remediate current security threats) were Cyber Security Vendors/Suppliers. The general theme that emerged from the Vendor/Supplier responses was that the focus should be on the security design of the application so

.

the API can inherit and leverage those features. An illustrative comment among the Vendor/Supplier segment demonstrates this theme:

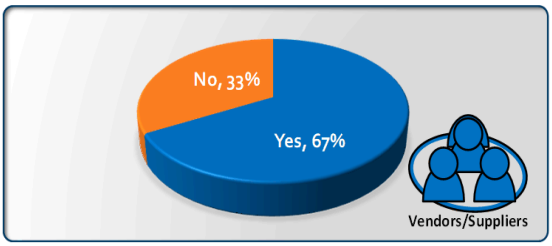
> "*An API is only the interaction with the application. The focus should be on developing the application properly. If we took the proper time to develop software with a security focus from step one, we wouldn't need to strengthen API. OWASP wouldn't exist."*

Respondents who answered "no", (who were all Customers and 33% of the Vendors/Suppliers) when asked if the API standards in place today are sufficient elaborated on their response. The general theme that emerged from the Customer responses was that the current API security standards are not robust enough to remediate current security threats. An illustrative comment among the Customer segment demonstrates this theme:

> "*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."*

In addition, those who identified themselves as Cyber Security Vendors/Suppliers were asked, "As a Cyber Security Vendor/Supplier, is your company planning to release any new products or services in your roadmap to address new API security vulnerabilities?" Over two thirds of the Cyber Security Vendors/Suppliers felt that new security products and services would address API vulnerabilities. Cyber Security Vendors/Suppliers; (specifically companies that produce security software and security services) responded "yes" at 67%. However, the rest of the Vendors/Suppliers responded "no" at 33%. The percentage split can be seen in Figure 4.

Figure 4. Are the upcoming Security Vendors/Suppliers Product and Service Roadmaps addressing new API vulnerabilities?

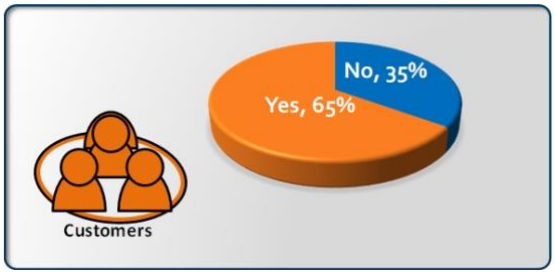


.

Source: Survey Results: The Future of API (Application Programming Interface) Security: The Adoption of APIs for Digital Communications and the Implications for Cyber Security Vulnerabilities

Furthermore, those who identified themselves as Cyber Security Customers were asked, “Are you planning to use Microservices or Serverless Compute?”. Approximately two thirds of the respondents were planning to implement new compute services, such as microservices and Serverless computing. Cyber Security Customers; (specifically security professionals that use security products to protect the data of their entities and clients) responded “yes” at 65% and the rest of the Cyber Security Customers responded “no” at 35%. The percentage split can be seen in Figure 5.

Figure 5. Are you planning to use Microservices or Serverless Compute?



Source: Survey Results: The Future of API (Application Programming Interface) Security: The Adoption of APIs for Digital Communications and the Implications for Cyber Security Vulnerabilities

CONCLUSIONS AND IMPLICATIONS

The primary issue with API vulnerabilities occurs when the security design of the underlying application is insufficient. API security only goes as far as it is designed. There are many real-world examples of "Works as Designed" (WAD), where the poor implementations for API security design left companies exposed to data breaches. These vulnerabilities were not the result of hacking, per se, since the hackers did not have to penetrate firewalls or decipher complicated encryption algorithms. The poor implementation of API security left the door wide open for anyone to walk in and harvest the data that they should not have had access to in the first place. Therefore, the expression that has recently developed, the "leaky API" is named appropriately (Spring, 2018).

.

Most recently revised, *it is now estimated that over 87 million Facebook Cyber Security Customers had their private information exposed by an API that was originally installed for a mobile application* (Romano, 2018). *The Facebook user data that was harvested by Cambridge Analytica (CA),* a data analytics firm that worked with political election campaigns, found the "leaky API" and extracted the supposedly private data from 87 million user accounts.

Between 2013 and 2015, Cambridge Analytica harvested profile data from Facebook users without permission and used that data to populate their own marketing database based on each user's individual likes and interests. They then created a personality profile for each user so they can be targeted for specific political campaigns more effectively. The Federal Trade Commission fined Facebook 5 billion dollars for mishandling data (Feiner, Rodriguez, 2019), and Cambridge Analytica ceased operations and filed for bankruptcy (Confessore, Rosenberg, 2018).

Cambridge Analytica was able to procure this data in the first place thanks to a loophole in Facebook's private API that allowed third-party developers to collect data not only from users of their apps but from all of the people in those users' friends network on Facebook. This access came with the stipulation that such data could not be marketed or sold; a rule CA promptly violated (Romano, 2018).

Reports calling CA's data harvesting a "hack," or a serious violation of Facebook policy are all incorrect.
This is because the information collected by the company was information that Facebook had freely allowed and originally intended only mobile developers to access. Technically, anyone who used third-party
Facebook apps also could have found out that they were allowing those apps to see data from their friends' profiles. As a Facebook spokesperson reiterated to the New York Times, "*No systems were infiltrated, and no passwords or sensitive pieces of information were stolen or hacked*." (Rosenberg, et al., 2018).

To this effect, the API-level Security Certification of Android Applications (ASCAA) organization found that out of 200 tested API applications, 12.5% failed their sample rules (Pei, et al., 2017). The ASCAA found evidence that the failed applications were either over-privileged or did not declare permissions at all.

Another example of the "Works as Designed" paradigm as it pertains to API security was the T-Mobile breach (Spring, 2018). In August of 2018, T-Mobile left an insecure, unprotected API on their website, thus exposing the personal data of 2.3 million Cyber Security Customers.

.

By directly manipulating the end of the URL (Uniform Resource Locator) string with a different phone number in the web browser, a hacker was able to test for actual customer phone numbers, and the web site responded with personal information. An example of the URL manipulation that was used is in Figure 6 below.

Figure 6. T-Mobile API – URL string;

```
.../...?access_token=xx&misisdn=123-456-7890
```

The API returned the following confidential customer data for 123-456-7890;

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

Another example demonstrated by Netflix employees revealed that certain API-based communications extended the attack surfaces for their microservices. In front of an audience of dozens of coworkers at the 2017 DefCon Security Conference, a Netflix security engineer ran a test on their streaming system.

He was able to bring the site down. Instead of admonishment, there was a sense of accomplishment since he, along with a Netflix cloud security engineer, successfully proved that the flagship streaming site was vulnerable to an unconventional type of Distributed Denial of Service (DDOS) attack.

Recognizing this new vulnerability triggered efforts from Netflix to protect the service from this new threat, along with the rest of the Internet. The premise was that a few simple requests could generate many backend requests, similar to a poorly written structured query language (SQL) script on a database. A query like this scans the entire list of tables inefficiently, thus filling up all of the database connections for any other traffic. Since the inbound client activity occurred below the rate limits on the API gateway, the critical protective measure for API traffic in the architecture let the bad request through (Newman, 2017). Rate limits, where the API gateway can set a fixed number of times an API can be evoked, can be an

.

effective way of securing an API. In this use-case, however, the requests occurred at a pace that evaded the rate limit setting.

After reviewing the survey responses and segmenting by Cyber Security Customers versus Vendors/Suppliers, there is a significant difference between Vendors/Suppliers and Cyber Security Customers of security products and services in terms of readiness to address current security threats. Vendors/Suppliers should more closely monitor Cyber Security Customers' perspectives so that indications in the area are infused with the "voice of the customer." Specifically, Cyber Security Customers sense that the current API security standards are not robust enough to mitigate current security threats while they are overwhelmingly considering new technologies such as Microservices or Serverless Compute.

API security is different from Web application security. API Authentication (e.g., JavaScript Object Notation (JSON) Web Token (JWT) and Oauth2) is a stateless transaction (Stannard, 2015). Websites that host the APIs do not track session data, so it is easy for a hacker to keep trying different combinations of URL string variations to exploit an unprotected, insecure API vulnerability. Web applications generally use stateful transactions that track session data by creating a session cookie (a tracking key that is valid only one time for that individual session) is a more secure transaction. The session cookie ensures that the transaction is a single conversation between one specific customer and the website. Session cookies typically cannot be reused, so a new one is created when the customer authenticates on the next visit. Traditional web security protects against structured query language (SQL) injection and cross-site scripting. API security requires more protections since hackers can go straight to the data via a stateless transaction, by nature of the service that APIs intend to provide.

API security is more complex since it happens at layer 7 of the Open Systems Interconnection (OSI) model (Mitchell, 2019). Layer 7 is the application layer, so the detection of malicious use via API gateways is only just starting to mature. An event where a hacker is retrieving data that is unprotected sometimes occurs below the rate limits of an API gateway (Netflix example) since it is not as evident as an Advanced Distributed Denial of Service (DDOS), which occurs at layer 4 of the OSI model; the transport layer. A DDOS attack will usually flood the gateway, and thus the web site behind it with so many connections it will affect service. A well-configured Intrusion Prevention System (IPS) can usually protect the web site by detecting the suspect source IP addresses and then preserve the service by dropping the specific TCP/IP packets only from those suspect source IPs. The issue is that IPS systems operate at layer four and not at layer 7, where the API traffic occurs.

ISSN: 1941-6679-On-line Copy

.

As such, the importance of API security in the current IT environment cannot be understated. Everything is a digital asset now (Harguindeguy, 2017). As an example, banks are now primarily an online presence, where most daily transactions are performed via smartphone. Photos reside on Facebook, Instagram, and iCloud accounts. Other elements that demonstrate the importance of API security are as follows;

- As reported by The ProgrammableWeb (Santos, 2017), Public API growth is exponential.
- Hackers always find the path of least resistance. An unprotected API service is an easy target (Wheeler, 2018).
- Respondents in this study indicated plans to implement new compute services, such as Microservices and Serverless compute. The industry is moving to the Internet of Things (IoT), Microservices, and Serverless Compute services (e.g., Amazon Web Services (AWS) Lambda), which extends the possible attack surfaces to hackers.
- According to the Global Equinix Interconnection Index (Equinix, 2019), by 2021, Interconnection bandwidth is projected to be ten times the volume of the present-day Internet.
- API exploits are usually not detected while they are occurring. Organizations need a robust security information and event management (SIEM) process to detect API exposure in order to remediate properly (Harguindeguy, 2017).

Security needs the proper nurturing to perform optimally for any organization. Some basic blocking and tackling fundamentals that can help any organization improve API security management are listed below, as indicated by the literature and this research.

1) Start with an API inventory, then gain visibility of your API traffic with leveraging a Security Information and Event Management system (SIEM) for logging (Harguindeguy, 2017).
    i) Know about the APIs that are up and running in your organization. Some APIs are installed via default when organizations install new software.
    ii) Do not register your internal API names in public DNS. Keep internal information internal.
    iii) If your organization utilizes an API Gateway, make sure that you are properly logging all events.
2) Always design with security in mind. If security is not a part of your design process right from the very beginning, your security strategy will perform as an afterthought, as it was designed. Most organizations consider security at the

.

time of integration or deployment, which is too late in the development process (Siriwardena, 2014).

3) Use API Management methodology. Beyond just providing the business functionality, send your APIs through the creation, publication, deprecation, and retirement cycles. Proper documentation is essential (Siriwardena, 2014).
4) If your organization is ready to adopt Agile operations, do not just implement better Development and Operations (DevOps) processes, but go the extra mile and implement Development, Security and Operations (DevSecOps) processes (George, 2018). Some DevSecOps examples are:
    i) Continuous Integration/Continuous Delivery (CICD),
    ii) Code repository/code review – Antivirus scan and automate code deployment with rollback.
    iii) Continuous Configuration Automation (CCA).
    iv) If you find any process that is repeatable and scriptable, automate.
5) Don't use basic authentication, use standard authentication (Salem, Mazalevskis, 2017) e.g.:
    i) JWT (JSON Web Token)
    ii) Oauth2
    iii) Username /password is not enough.
    iv) Use end-user authentication rather than API keys or Client ID/Client secret when possible.
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, use the proper HTTP methods for GET, POST, PUT and DELETE and validate the content (Salem, Mazalevskis, 2017).
8) For output, send X-Content and X-Frame options and don't return sensitive data (Salem, Mazalevskis, 2017).
9) Concerning processing (Salem, Mazalevskis, 2017):
    i) for endpoint protection, avoid user-owned resource IDs and auto-incremented IDs, use Universally Unique Identifiers (UUID),
    ii) use End-To-End TLS (version 1.3).

IMPLICATIONS FOR FUTURE RESEARCH

This study utilized a descriptive research design in the form of the Qualtrics XM Online survey tool. It was conducted among qualified respondents to obtain better clarity on security issues related to APIs. The data captured and the statistical analysis generated helped determine how respondents as security community Cyber

.

Security Customers and Vendors/Suppliers have different views on API vulnerabilities.

The sample size yielded a statistically significant result in exploring the hypothesis test of differences between Cyber Security Customers and Vendors/Suppliers. Future research would also attempt to gain insights from a broader audience for a more global perspective. Furthermore, future research can take on a qualitative aspect with additional one-on-one in-depth interviews (IDIs) to further explore the insights surfaced in this research utilizing comprehensive probing techniques to understand management practices in greater detail.

REFERENCES

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

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.

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-shut-down.html?searchResultPosition=2.

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

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.

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

.

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

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

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.

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.

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

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).

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

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.

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

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

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

.

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

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.

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

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.

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.

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.

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

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.

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.

Siedlecki, Sandra L. (2020, January/February). Understanding Descriptive

.

Research Designs and Methods. Clinical Nurse Specialist. Retrieved from https://journals.lww.com/cns-journal/Fulltext/2020/01000/Understanding_Descriptive_Research_Designs_and.4.aspx.

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.

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.

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

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.

Wheeler, C., (2018, February). Three New Attack Vectors That Will Be Born Out of IoT. Liquid Web. Retrieved from https://www.liquidweb.com/blog/three-new-attack-vectors-will-born-iot/.

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/.

.