ARTICLE

Securing APIs: 10 Best Practices for Keeping Your Data and Infrastructure Safe

Learn the potential risks of APIs and how straightforward it is to secure them.

In this article, we look at how APIs can pose risks to your data and infrastructure—and what you can do to secure them.

Web APIs (application programming interfaces) provide a way for app developers to “call” information from outside sources into the applications they build. A travel app for example, uses web API calls to pull in availability and pricing information from various hotel, airline, cruise line, tour, car rental, and other companies. APIs benefits app developers by simplifying the coding process and granting them access to a wealth of data and resources they would not otherwise be able to access. APIs also benefit providers, who are able to create new revenue streams by making valuable data and services available to developers, usually for a fee. And ultimately, APIs benefit consumers, who appreciate (and drive demand for) innovative, feature-rich, interactive apps that provide many services all in one app.

GARTNER ESTIMATES THAT “BY 2022, API ABUSES WILL MOVE FROM INFREQUENT TO THE MOST FREQUENT ATTACK VECTOR, RESULTING IN DATA BREACHES FOR ENTERPRISE WEB APPLICATIONS.”[1]

Understanding the Potential Risks of APIs

The downside of publicly available web APIs is that they can potentially pose great risk to API providers. By design, APIs give outsiders access to your data: behind every API, there is an endpoint—the server (and its supporting databases) that responds to API requests (see Figure 1). In terms of potential vulnerability, an API endpoint is similar to any Internet-facing web server; the more free and open access the public has to a resource, the greater the potential threat from malicious actors. The difference is that many websites at least employ some type of access control, requiring authorized users to log in. One problem with some APIs, as we’ll see shortly, is that they provide weak access control and, in some cases, none at all. With APIs becoming foundational to modern app development, the attack surface is continually increasing. Gartner estimates that “by 2022, API abuses will move from infrequent to the most frequent attack vector, resulting in data breaches for enterprise web applications.”[1]

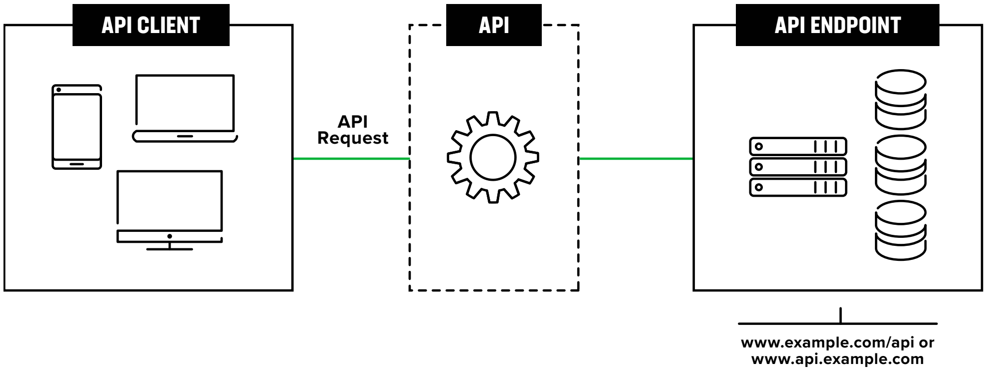

Figure 1: Web APIs connect to an endpoint: the location of the web server and supporting databases

IN WORST CASE, IT'S NOT JUST YOUR DATA THAT IS POTENTIALLY AT RISK BUT ALSO YOUR INFRASTRUCTURE.

In worst case, it's not just your data that is potentially at risk but also your infrastructure. By exploiting a vulnerable API, attackers can gain access to your network using one kind of attack. If they're able to escalate privileges, they can then pivot to other types of attacks and gain a foothold in the network. The right attack—often a multi-level attack—could potentially lead to your organization's most sensitive data being compromised, whether it's personally identifiable information (PII) or intellectual property (IP).

No matter what the attack vector, a data breach is a data breach: it can damage your company's brand and reputation and could result in significant fines and lost revenue. No organization is immune; some of the largest and well-known companies—Facebook,[2, 3] Google,[4] Equifax,[5] Instagram,[6, 7] T-Mobile,[8] Panera Bread,[9] Uber,[10] Verizon,[11] and others—have suffered significant data breaches as a result of API attacks. It's imperative for all companies, not just large ones, to secure all APIs, particularly those that are publicly available.

Common Attacks Against Web APIs

APIs are susceptible to many of the same kinds of attacks defenders have been fighting in their networks and web-based apps for years. None of the following attacks are new but can easily be used against APIs.

- Injection occurs when an attacker is able to insert malicious code or commands into a program, usually where ordinary user input (such as a username or password) is expected. SQL injection is a specific type of injection attack, enabling an attacker to gain control of an SQL database.
- Cross-site scripting (XSS) is a type of injection attack that occurs when a vulnerability enables an attacker to insert a malicious script (often JavaScript) into the code of a web app or webpage.
- Distributed denial-of-service (DDoS) attacks make a network, system, or website unavailable to intended users, typically by flooding it with more traffic than it can handle. API endpoints are among the growing list of DDoS targets.
- Man-in-the-middle (MitM) attacks occur when an attacker intercepts traffic between two communicating systems and impersonates each to the other, acting as an invisible proxy between the two. With APIs, MitM attacks can occur between the client (app) and the API, or between the API and its endpoint.
- Credential stuffing is the use of stolen credentials on API authentication endpoints to gain unauthorized access.

Briefly, Table 1 matches attack types to traditional mitigations:

API Attack Types and Mitigations	
Attack Type	**Mitigations**
Injection	Validate and sanitize all data in API requests; limit response data to avoid unintentionally leaking sensitive data
Cross-Site Scripting (XSS)	Validate input; use character escaping and filtering
Distributed Denial-of-Service (DDoS)	Use rate limiting and limit payload size
Man-in-the-Middle (MitM)	Encrypt traffic in transit
Credential Stuffing	Use an intelligence feed to identify credential stuffing and implement rate limits to control brute force attacks

Table 1: Common attack types that can be used against APIs matched to corresponding mitigations

Best Practices for Securing APIs

IT'S IMPERATIVE FOR ALL COMPANIES, NOT JUST LARGE ONES, TO SECURE ALL APIS, PARTICULARLY THOSE THAT ARE PUBLICLY AVAILABLE.

In addition to employing the mitigations outlined in Table 1, it's critical that organizations adhere to some basic security best practices and employ well-established security controls if they intend to share their APIs publicly.

- **Prioritize security**. API security shouldn't be an afterthought or considered "someone else's problem." Organizations have a lot to lose with unsecured APIs, so make security a priority and build it into your APIs as they're being developed.
- **Inventory and manage your APIs**. Whether an organization has a dozen or hundreds of publicly available APIs, it must first be aware of them in order to secure and manage them. Surprisingly, many are not. Conduct perimeter scans to discover and inventory your APIs, and then work with DevOps teams to manage them.
- **Use a strong authentication and authorization solution**. Poor or non-existent authentication and authorization are major issues with many publicly available APIs. Broken authentication occurs when APIs do not enforce authentication (as is often the case with private APIs, which are meant for internal use only) or when an authentication factor (something the client knows, has, or is) can be broken into easily. Since APIs provide an entry point to an organization's databases, it's critical that the organization strictly controls access to them. When feasible, use solutions based on solid, proven authentication and authorization mechanisms such as OAuth2.0 and OpenID Connect.

NO MATTER HOW MANY APIS YOUR ORGANIZATION CHOOSES TO SHARE PUBLICLY, YOUR ULTIMATE GOAL SHOULD BE TO ESTABLISH SOLID API SECURITY POLICIES AND MANAGE THEM PROACTIVELY OVER TIME.

- **Practice the principle of least privilege**. This foundational security principle holds that subjects (users, processes, programs, systems, devices) be granted only the minimum necessary access to complete a stated function. It should be applied equally to APIs.
- **Encrypt traffic using TLS**. Some organizations may choose not to encrypt API payload data that is considered non-sensitive (for example, weather service data), but for organizations whose APIs routinely exchange sensitive data (such as login credentials, credit card, social security, banking information, health information), TLS encryption should be considered essential.
- **Remove information that's not meant to be shared**. Because APIs are essentially a developer's tool, they often contain keys, passwords, and other information that should be removed before they're made publicly available. But sometimes this step is overlooked. Organizations should incorporate scanning tools into their DevSecOps processes to limit accidental exposure of secret information.
- **Don't expose more data than necessary**. Some APIs reveal far too much information, whether it's the volume of extraneous data that's returned through the API or information that reveals too much about the API endpoint. This typically occurs when an API leaves the task of filtering data to the user interface instead of the endpoint. Ensure that APIs only return as much information as is necessary to fulfill their function. In addition, enforce data access controls at the API level, monitor data, and obfuscate if the response contains confidential data.
- **Validate input**. Never pass input from an API through to the endpoint without validating it first.
- **Use rate limiting**. Setting a threshold above which subsequent requests will be rejected (for example, 10,000 requests per day per account) can prevent denial-of-service attacks.
- **Use a web application firewall**. Ensure that it is able to understand API payloads.

Conclusion

APIs have arguably become the preferred method for building modern applications, especially for mobile and Internet of Things (IoT) devices. And while the concept of pulling information into a program from an outside source is not a new one, constantly evolving app development methods and the pressure to innovate means some organizations may not yet have grasped the potential risks involved in making their APIs publicly available. The good news is that there's no great mystery involved in securing them. Most organizations already have measures in place to combat well-known attacks like cross-site scripting, injection, distributed denial-of-service, and others that can target APIs. And many of the best practices mentioned above are likely quite familiar to seasoned security professionals. If you're not sure where to begin, start at the top of the list and work your way down. No matter how many APIs your organization chooses to share publicly, your ultimate goal should be to establish solid API security policies and manage them proactively over time.

To learn more, visit f5.com/solutions/manage-and-secure-apis

[1] "API Security: What You Need to Do to Protect Your APIs," Gartner, August 28, 2019, found at https://www.gartner.com/en/documents/3956746/api-security-what-you-need-to-do-to-protect-your-apis

[2] "FTC hits Facebook with record $5 billion fine for user privacy violations," ZDNet, July 24, 2019, found at https://www.zdnet.com/article/ftc-hits-facebook-with-record-5-billion-fine-for-user-privacy-violations/

[3] "Facebook reveals another privacy breach, this time involving developers," ZDNet, November 6, 2019, found at https://www.zdnet.com/article/facebook-reveals-another-data-breach-this-time-involving-developers/

[4] "Google Plus shutdown date gets moved up after new security breach," CBS News, December 10, 2018, found at https://www.cbsnews.com/news/google-plus-shutdown-date-moves-up-after-another-security-breach/

[5] "Issue 41: Tinder and Axway API Vulnerability, Equifax fined," APIsecurity.io, July 25, 2019, found at https://apisecurity.io/issue-41-tinder-and-axway-breached-equifax-fined/

[6] "Issue 41: Tinder and Axway API Vulnerability, Equifax fined," TechCrunch, May 20, 2019, found at https://techcrunch.com/2019/05/20/instagram-influencer-celebrity-accounts-scraped/

[7] "Instagram Says Bug Gave Hackers Data on 'High-Profile' Users," TIME, August 30, 2017, found at https://time.com/4922700/instagram-security-breach-verified-users/

[8] "T-Mobile Alerts 2.3 Million Customers of Data Breach Tied to Leaky," Threatpost, August 24,2018, https://threatpost.com/t-mobile-alerts-2-3-million-customers-of-data-breach-tied-to-leaky-api/136896/

[9] "Panera Bread Website Leaking Customer Data," eWeek, April 3, 2018, found at https://www.eweek.com/security/panera-bread-website-leaking-customer-data

[10] "Uber Confirms Account Takeover Vulnerability Found By Forbes 30 Under 30 Honoree," Forbes, September 12, 2019, found at https://www.forbes.com/sites/daveywinder/2019/09/12/uber-confirms-account-takeover-vulnerability-found-by-forbes-30-under-30-honoree/#43291e2d9b87

[11] "Verizon Router Command Injection Flaw Impacts Millions," Threatpost, April 9, 2019, found at https://threatpost.com/verizon-quantum-gateway-command-injection-flaw-impacts-millions/143606/