



Application Programming Interfaces

Security Risks and Controls

Introduction

Application Programming Interfaces (APIs) provide a way for users and other applications to interact with software and data. It may be helpful to think of an API as a sort of broker; just like stock brokers receive buy and sell orders, software APIs use specific communication jargon — or protocols — to transact information requests. This analogy is useful to understand how a website API handles requests for a web application, by receiving instructions from people using web browsers (e.g., “show me the news front page in English, and only for my local region”), and then serve the information to the browser that corresponds with the specific request.

This request and response process is how APIs are able to present dynamic, data-driven web content as web pages in browsers. It allows for scalable application design, so that web pages can handle millions of requests and still provide a high-quality experience to users. However, API use presents a set of specific cybersecurity risk challenges that must be considered in the development, deployment, and maintenance of APIs. This whitepaper will focus on the best practices for securing APIs and the technologies and services that can be used to protect them.

API Risks and Controls

This whitepaper will discuss risks and controls associated with various API formats divided into three broad categories of risk, 1) Discoverability, 2) Input Validation, and 3) Access Control.

Discoverability Risks

Discoverability risks are not themselves a type of attack, but by publishing an API on the Internet and allowing connections from public internet systems, organizations allow users and threat actors to enumerate, or discover the various functions of an API. In short, if you don't understand exactly how your API works, but publish it, threat actors may discover things that your API will do that you are not even aware of!

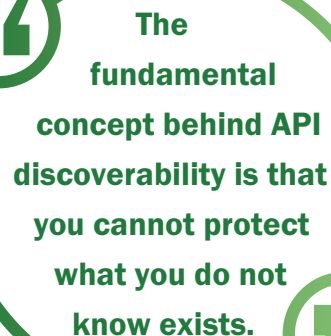
Accordingly, this whitepaper will further categorize discoverability risks as 1) left-of-breach, 2) mid-breach, and 3) right-of-breach or post-breach. The fundamental concept behind API discoverability is that you cannot protect what you do not know exists.

Left-of-Breach

Left-of-Breach refers to the timeline prior to a breach occurring. Left-of-Breach discoverability refers to having adequate visibility into your API environment to take the steps necessary to minimize the likelihood or risk of a breach occurring and includes ensuring that your organization has a complete inventory and comprehensive documentation of all APIs that exist within your organization.

Risks

According to the Salt Labs 2023 Q3 State of API Security Report, 86% of surveyed organizations stated that they “lack confidence that their API inventory is complete.”¹ API inventory challenges are often a result of “API Sprawl” and a general lack of API governance. The risks of undocumented APIs include difficulties with future development efforts, insecure APIs being used in production environments, and lack of monitoring or response capabilities if a breach does occur.



**The
fundamental
concept behind API
discoverability is that
you cannot protect
what you do not
know exists.**

Controlling Risks Prior to Breach

At a high level, the Left-of-Breach discoverability controls are all components of implementing Development, Security, and Operations (DevSecOps) practices and implementing API Governance processes. At a more granular level, key controls for ensuring that organizations have the required visibility into their API environment Left-of-Breach include:

- Creating and maintaining an API inventory;
- API Documentation - APIs should be documented so that appropriate monitoring and alerting can be configured;
- API Discovery - Organizations should invest in tools which automate the discovery of APIs within the environment.

Mid-Breach

Mid-Breach refers to the timeline while a breach is actively occurring. Mid-Breach discoverability includes having sufficient logging within your API environment as well as a sufficient understanding (and documentation) of your APIs to implement and automate actionable alerts and the staffing and processes necessary to quickly detect and respond to a breach.

Risks

Web applications, especially those dedicated to processing sensitive data, are continually under attack by cybercriminals. Organizations must take measures to actively monitor for breaches and perform a coordinated response. Organizations should monitor for unauthorized access (from unauthorized users, compromised users, or even from unapproved IP addresses) to API endpoints which can access sensitive data. However, without sufficient understanding and documentation of their API environment, organizations face several significant risks, including:

- APIs without sufficient logging cannot be monitored for abuse or compromise;

¹“State of API Security Q3 2022” (Salt Labs, 2022), <https://salt.security/api-security-trends>, 13.

- APIs that are not well-documented may not be properly understood in order to accurately interpret what logs do exist;
- APIs that are not known to exist cannot be monitored for abuse or compromise.

Controls That Can Reduce Impact During a Breach

The amount of time that passes from the initial discovery of a breach until the breach has been remediated and the resumption of normal business activity is the strongest indicator of how much impact the organization will take as a result of the breach. Getting back to “the day before the breach,” from an operational perspective, is the primary goal of any response activity. While discussion of specific controls described below is outside of the scope of this whitepaper, organizations can rely on the following controls to detect and respond to an intrusion that leverages API vulnerabilities and minimize the time it takes to recover from an intrusion:

- Managed Extended Detection and Response (MXDR)
- Specific, application-based alerting related to API use via MXDR SIEM
- Next Generation Antivirus and Endpoint Detection & Response
- Network Segmentation and Data Segregation

Right-of-Breach

Right-of-Breach API Security involves ensuring that response activities are well-informed, remediation activity is effective, and the application of lessons learned from an incident is effective.

Risks

During an Incident Response (IR), incident responders may not have an adequate understanding or sufficient logs to reach a determination on root-cause or the impact of any breach which may have occurred. IR engagements are often expensive and time consuming, and lack of API documentation may result in extended IR engagements or incomplete conclusions. Insufficient (or non-existent) logging may prevent incident responders from having the visibility necessary to determine what or how the breach occurred.

Controls

The following controls are critical in successfully resolving an intrusion that has leveraged web APIs:

- API Documentation
- Log Retention
- Incident Response Plan
- Tabletop Exercises

Input Validation Risks

APIs exist to facilitate interaction with data between two or more applications. Regardless of API format, almost every type of API attack is essentially the result of some type of input

Almost every type of API attack is essentially the result of some type of input handling vulnerability, and the controls to mitigate those attacks are to validate the way APIs handle and respond to that input.

handling vulnerability, and the controls to mitigate those attacks are to validate the way APIs handle and respond to that input. The result of various Input Validation exploits can generally be divided into two categories, 1) Denial of Service Queries and 2) Injection and Extraction Queries.

Denial of Service Queries

API Denial-of-Service attacks are any type of API attack which has a goal of preventing authorized users access to the application or the data processed by the application.

Risks

API Denial-of-Service attacks are the result of excessive use of application resources. Web Applications process API requests on hardware that have finite resources (CPU, memory, storage), and if applications are not configured to limit requests to those resources, a Denial-of-Service may occur. The OWASP API Top 10 categorizes these types of attacks as “Lack of Resources or Rate Limiting” vulnerabilities.² Inadequate system resources and ineffective API rate limiting introduce the risk of inaccessible applications.

Controls

In order to minimize the risk of API Denial-of-Service attacks applications should implement:

- Static Application Security Testing (SAST)
- Rate limiting to prevent system or network resource exhaustion
- Application delivery platforms, such as Docker, for simple resource management
- Web Application Firewalls (WAFs)
- Enforced server-side validation of requests and record response count limitations
- Enforced maximum data sizes and string lengths limitations

Injection and Extraction Queries

There are many different types of API attacks which can be categorized as “malicious queries”, but at a high-level the goal of nearly all Malicious API Queries is to either inject data or to extract data from the application or the systems where the application resides.

Injection Attack Risks

Injection attacks are any type of attack where the purpose is to add data to the application or the systems where the application resides. This includes attacks where the API is abused to inject a shell that could provide threat actors with remote access to the underlying system. Injection attacks also include inserting or modifying data to introduce incorrect, fraudulent, or invalid data. Organizations must consider the type of data their application processes and the potential for abuse if cybercriminals are able to insert or modify that data.

The goal of nearly all Malicious API Queries is to either inject data or to extract data from the application or the systems where the application resides.

²<https://github.com/OWASP/API-Security/blob/master/2019/en/src/Oxa4-lack-of-resources-and-rate-limiting.md>

Extraction Attack Risks

Extraction attacks are any type of attack where the purpose is to exfiltrate data from the application or the systems where the application resides. Extraction attacks can include API abuse that reveals vulnerabilities, and privilege escalation through use of these vulnerabilities. Extraction attacks also include those which allow unauthorized individuals to access data from the application itself.

Controls

There are many different types of API malicious query attacks, and each of these has corresponding preventative controls. Implementing the preventative measures for the OWASP API Top 10 Vulnerabilities will significantly improve an organization's API risk posture surrounding malicious queries. However, organizations should also consider the following controls to limit their risk to Malicious API Queries:

- Static Application Security Testing (SAST)
- Dynamic Application Security Testing (DAST)
- Penetration Testing

Access Control

API Access Control Risks can broadly be categorized into 1) Authorization Risks and 2) Authentication Risks. Vulnerabilities such as Insecure Direct Object Reference (IDOR) are made possible via parameter tampering and can allow attackers unauthorized access to resources.

User validation plays another critical part in API security. The level of exploitation risk differs depending on the API format. Stateless APIs such as RPC, REST, and GraphQL pose a greater risk as sessions are stored client-side. Client-side tokens, such as JSON Web Tokens (when not properly signed), can be decoded to reveal potentially sensitive information and edited before being passed to the server. Unauthenticated API calls and data exposure can be the result of such errors and misconfigurations.

Controls

- Static Application Security Testing (SAST)
- Dynamic Application Security Testing (DAST)
- Web Application Penetration Testing
- Multi-Factor Authentication (MFA)
- API Gateway Systems

Contributors

Evans Foster, Jason Ingalls, Chase Theodos, Cyrus Robinson, Kim Buckley, and John Frasier



Cyber Innovation Center (CIC)
6300 Texas Street, Ste. 240, Bossier City, LA 71111

WWW.IINFOSEC.COM
(888) 860-0452