SANS

Analyst Program

Survey

2023 SANS Survey on API Security

Written by **John Pescatore**

July 2023

Akamai

Introduction

From its beginning, computing has involved continual movement from monolithic to distributed and layered systems. Computers went from mainframes to departmental to client/server to virtual machines to cloud computing. Networks went from point-to-point connections to layered physical networks to internet communications. Applications went from monolithic blocks of code to layered to distributed applications. See Figure 1.

This migration led to increases in performance and flexibility, but as the old saying goes, "There is no such thing as a free lunch." Those advantages came at the expense of additional complexity and, as the other old saying goes, "Complexity is the enemy of security." Distributed applications invariably increase both the attack surface available to malicious actors and the likelihood of vulnerabilities being built into production code.

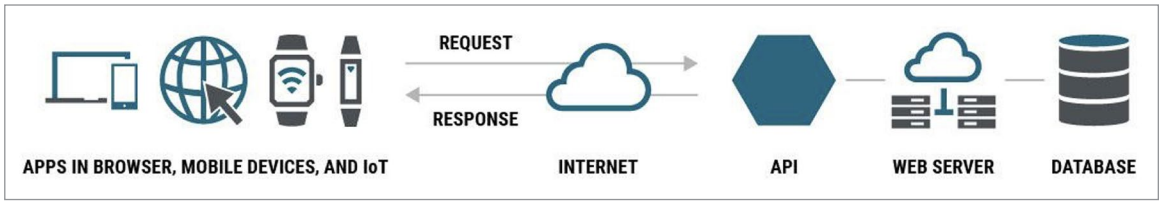

Figure 1. Evolution of Computing Migration (Source: Axway)

Modern applications use application programming interfaces (APIs) to define rules for how different elements should communicate with each other. In a distributor's catalog, for example, rather than having to continually modify one gigantic application every time a supplier is added or deleted, or their listing is changed, the distributor publishes APIs that define data flows for vendors to join, leave, update, and so on. These APIs essentially capture the business processes and break them into the lower-level communications required to efficiently enable business partners and customers to work with the business. A 2022 survey by 451 Group Research reported the average enterprise has more than 15,000 APIs in use.[1]

Like software developers, API writers are highly skilled at capturing legitimate business requirements and defining how legitimate business needs can be met efficiently. Modern APIs also must support a variety of computing platforms and user devices, which means that APIs are a threat surface that malicious actors may try to subvert, corrupt, or disrupt in unexpected ways. Most APIs get updated many times as attackers find vulnerabilities that will then need to be mitigated.

The most used standards for implementing APIs are Simple Object Access Protocol (SOAP) and Representational State Transfer (REST). SOAP is XML-based and incorporates WS-Security for encryption, digital signing, and authentication services. REST is HTML-based and uses HTTPS and JSON standards.

[1] S&P Global Market Intelligence, "The 2022 API Security Trends Report," https://nonamesecurity.com/resources/api-security-trends-report/

The bottom line is that API security, like application security, starts with:

- Inventory of APIs in use and processes that use those APIs
- Vulnerability assessment of APIs in use
- Threat assessment of active attacks exploiting those vulnerabilities
- Risk-based mitigation of critical API vulnerabilities

Although those security activities are well known, there are often gaps in knowledge, skills, and management prioritization in applying them to API security issues. The SANS API security survey was conducted to determine enterprise awareness, readiness, and future plans for dealing with API security risks.

Survey Results

In most publicly reported security incidents, the top three exploited vulnerabilities are generally:

1. Reusable privileged credentials obtained via phishing
2. Attackers exploiting misconfigured servers or cloud services
3. Exploitation of missing patches on servers and PCs

These same issues (weak authentication, misconfigured settings/positions, and failure to use latest versions) are vulnerabilities that are also exploited in attacks focusing on APIs.

Perceived Risks

Survey respondents ranked phishing and missing patches as the top two API security risks. See Figure 2. Of note, misconfigured servers/services were rated last in the weighted rankings, below exploiting vulnerable apps/APIs with zero-day (no patch available) attacks.

The weighted results from this same question show the following ranking:

1. Phishing to obtain reusable credentials
2. Attackers exploiting missing patches
3. Attackers exploiting vulnerable applications/APIs
4. Accidental disclosure of sensitive/covered information by users
5. Denial of service
6. Misconfiguration of servers/services by system administrators

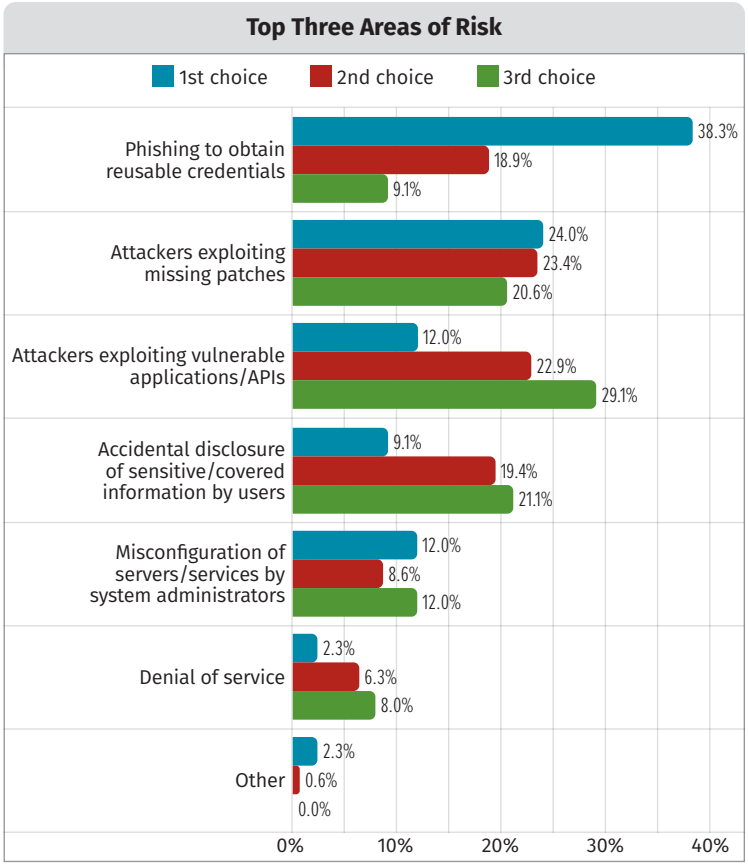

Figure 2. Top Three Areas of Risk

Takeaways

In the ranked weightings, when thinking about API security risks, respondents seemed to be underweighting the risk of misconfigured applications and overestimating zero-day risks. However, the same number of respondents chose misconfigured applications as their top risk as chose zero-day risks, indicating awareness of misconfiguration risks. Security managers should prioritize assuring that an accurate inventory of APIs is maintained, that updated versions of APIs are in use, and that configurations and options emphasize security.

Frameworks in Use

Cybersecurity frameworks provide a common language and reference model for determining the completeness of a security program, exposing gaps, and assessing risks. Mature security programs generally use full-coverage frameworks such as the Center for Internet Security Critical Security Controls or the NIST Cybersecurity framework.

More than half of respondents cited the Open Worldwide Application Security Project (OWASP)[2] Application Security and API Top Ten lists (Figure 3), and the MITRE ATT&CK Framework[3] as the basis for defining application and API risk. See Figure 4.

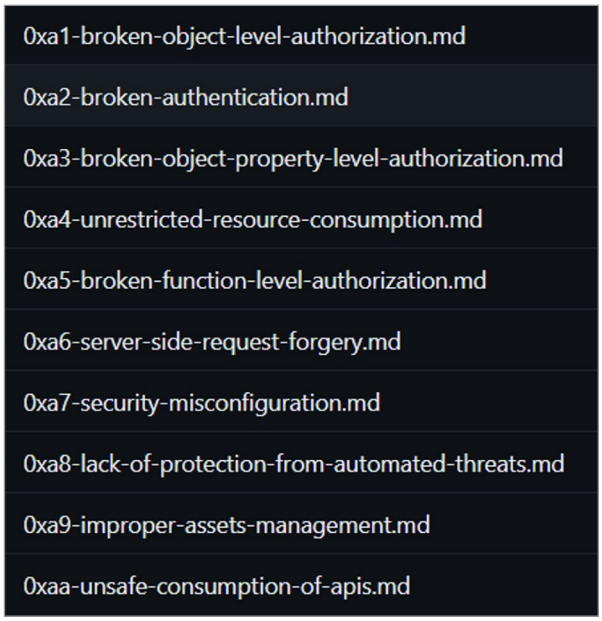

Figure 3. OWASP API Security Top 10 Vulnerabilities

Takeaway

The OWASP API Top 10 vulnerabilities and the MITRE ATT&CK model are powerful community-driven starting points for vulnerability assessment of APIs in use, assessing protection gaps and prioritizing action steps to mitigate API risks.

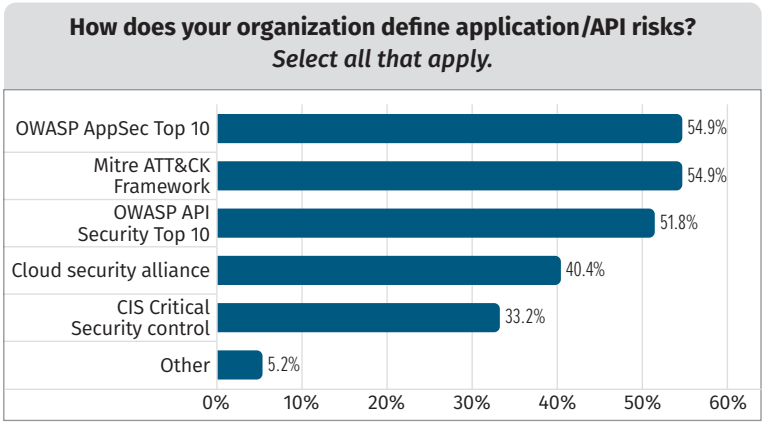

Figure 4. Frameworks Used to Define Application and API Risk

Tools/Controls in Use

Vulnerability assessment and management is a core component of every successful cybersecurity program. It requires a well-defined set of processes, including:

- **Discovery/inventory—**Knowing what systems, networks, resources, and applications are relied on for business operation
- **Vulnerability assessment and prioritization—**Determining if assets have vulnerabilities and their level of exposure and criticality
- **Remediation/mitigation—**Applying patches to or replacing vulnerable assets or shielding those that cannot be remediated

[2] OWASP is a nonprofit organization that has been leading community efforts to improve the security of applications and the accuracy and effectiveness of application security tools since 2001.

[3] MITRE, a nonprofit company that operates US federally funded research labs, started ATT&CK in 2013 to document the tactics, techniques, and procedures (TTPs) actively being used to compromise enterprise networks, systems, applications, and data. The MITRE ATT&CK framework is a widely used model for defining API threat models and assessing current and needed security posture against API threats.