WHY YOUR APIS ARE NOT AS SECURE AS YOU THINK

API Security Done Right

graylog

Today's organization relies on Software-as-a-Service (SaaS) applications as a modern "office" building. Everything from workforce member communications to daily activities occur in the cloud. Application programming interfaces (APIs) are the digital bricks with which organizations build these new offices. In a traditional office, workforce members can share information with peers by walking to each others' desks. In cloud-native "offices," APIs enable applications to share information, enhancing collaboration across organizations.

However, while APIs provide the core infrastructure for these business activities, they also create new security challenges and risks. An organization adopting more cloud-based technologies increases the number of APIs, expanding the attack surface. Organizations need vast amounts of ever-changing APIs while struggling to assign responsibility for them to the right people.

Simultaneously, organizations believe they are securing their APIs with their existing technology. But, tools like Web Application Firewalls and API Gateways focus on performance monitoring and controlling HTTP traffic. They do not provide threat detection and response for APIs and cannot protect against authenticated attacks that bypass perimeter defenses.

The Business Case

In a 2023 report, Enterprise Strategy Group (ESG) found that 35% of respondents reported more than 30% of their production workload run on public cloud infrastructure. They estimated that the number of organizations would increase to 62% over the next two years. The data only reinforces the value that organizations gain by incorporating APIs into the environments. Some examples include:

- **Generating sales and revenue:** Connect and exchange data with customers and partners or expand services through APIs
- **Automating tasks:** Reduce time and administrative costs arising from manual processes
- **Stimulating innovative business solutions:** Enable workforce members to reduce mundane, repetitive, time-consuming tasks so they can focus on higher-value activities.
- **Reducing software development costs:** Reduce the need for custom code, hosting, and maintenance costs

- **Supporting sales and marketing strategies:** Connect technologies, tools, and data to drive leads and increase customer lifetime value

- **Enhancing security on other applications:** Integrate cybersecurity technologies with cloud-based infrastructures, environments, and applications to aggregate security monitoring.

- **Enhancing customization to improve services:** Tailor user experiences to respond to customer demands and expectations with a more efficient online experience.

Pervasive Security Incidents

Simultaneously, the surge in API use combined with the dearth of API-focused security technologies leaves many organizations struggling as threat actors target this new attack vector. The ESG research provides additional insights, noting that in the last twelve months:

- 57% of respondents experienced multiple security incidents related to insecure APIs

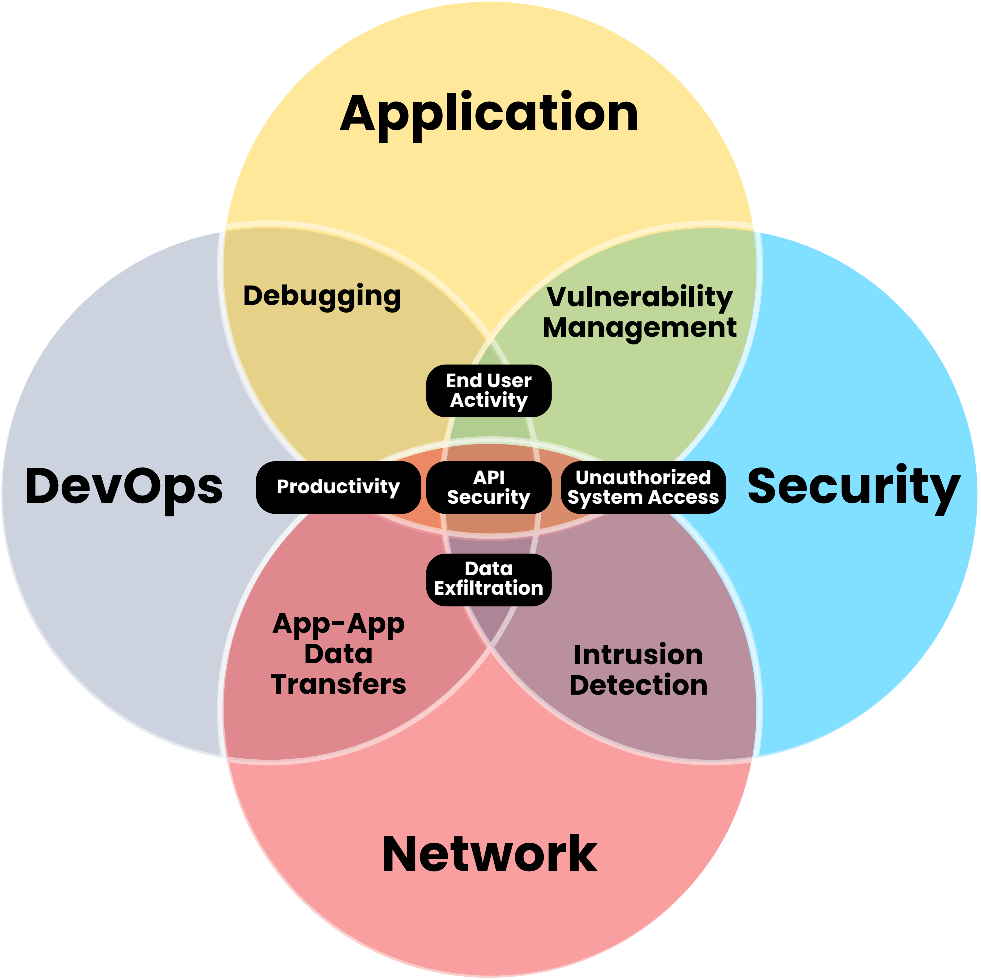
- 35% of respondents experienced one security incident related to insecure APIs

A small number, 1% of respondents, had so little visibility into their API's security that they thought they might have experienced a security incident but were unsure.

A Gray Area of Responsibility

How APIs integrate into environments adds another layer of confusion around security and responsibility. At a basic level, they are attack vectors at the network and application layer. Attackers can use APIs to gain access to networks and then move laterally within the system. Additionally, malicious actors can use APIs to gain unauthorized access to applications managing sensitive data, like payment card information or personally identifiable information (PII). While APIs can undermine network and application security, the tools that manage those security functions fail to provide comprehensive insight.

Meanwhile, they also sit in a responsibility gray area. DevOps may implement them, but security teams need to monitor them. Building an API from scratch typically falls within the software engineer's domain, but cloud security engineers implement the policies and procedures for monitoring, occasionally building APIs to cover edge cases.



Since API sits at the intersection of different domains and business needs, identifying the responsible parties and finding the right solutions becomes overwhelming.

Application Problem Interface *or* Application Programming Interface?

As organizations digitally transform their business operations, they add more applications and integrate more APIs into their environments. Unfortunately, the intersection of API security and DevOps means that most organizations need a clear line of responsibility, creating extra work and miscommunication that can lead to errors. Even more frustrating for most organizations, their traditional security monitoring solutions provide some visibility but still have blindspots that leave them struggling to gain the holistic visibility needed for a robust security posture.

TOO MANY PLACES

Everyone says that "modern IT environments are complex." However, the statement can seem vague or trite without statistics underpinning it. According to Abnormal Security, the average organization integrates 379 third-party applications. Furthermore, 75% of ESG's research respondents report that they have an average of 26 APIs per application deployed. If each application has 26 APIs, the average organization needs to monitor 9,854 APIs.

75% of ESG's research respondents report that they have an average of 26 APIs per application deployed.

A Variety of APIs

APIs come in different flavors, each responding to a level of public internet access risk. Across an organization's 9,854 APIs, it can have any number of the following types:

- **Open/Public:** intended to be used by any third-party developer with minimal restrictions, often accessible using an API key or fully open
- **Partner:** provided with specialized authentication requirements for access but not entirely private
- **Private/Internal:** used by in-house development teams to connect internal systems for productivity, service reuse, sharing, and platform integration and not publicly accessible
- **Hybrid/Composite:** built with API creation tools by developers to combine multiple services or data into a single API

According to ESG's research, organizations use various types of APIs with:

- 67% using open APIs for public consumption
- 64% using APIs to connect applications with partners
- 51% using APIs to connect microservices

Organizations with this variety struggle because each type poses a risk that needs mitigation. For example, a public API with fewer access restrictions creates a potential risk when connected to the public internet. Still, if it is from a trusted party, it may have fewer vulnerability issues. Meanwhile, an internal API may lower access risk if configured properly, but a developer error can create a different security risk.

DIFFERENT FORMATS AND PROTOCOLS

Although all API communications start with requests from the application, their connections to the backend server and security capabilities differ widely. Typically, an organization can have an API landscape consisting of one or more of the following:

- **Representational State Transfer (REST):** A layered, stateless, client-server architecture that communicates using application-layer HyperText Transfer Protocol Secure (HTTPS) or Secure Socket Layer (SSL) for security
- **Simple Object Access Protocol (SOAP):** A protocol with rules and standards around sending and securing messages that supports WS-Security, including encryption, digital signatures, and authentication
- **Remote Procedure Call (RPC):** A specification where the client invokes a remote procedure to send messages to a server often used for internal services from an application on another server or on a network without knowing the network's details. An RPC calls processes on a remote system like it were local to its own system.
- **GraphQL:** A query language that pulls data from various sources to return precise results that rely on query parameters for security

Organizations need help monitoring and securing all APIs because they may have APIs with varying security maturity capabilities.

INABILITY TO DISCOVER

The various types of APIs and their configurations make discovery even more challenging. Many organizations rely on API gateways and web application firewalls (WAF) to identify active APIs. However, not all API calls go through the gateway, creating blind spots.

Without the ability to discover all APIs, organizations may have:

- **Zombie API:** An API that remains open as an access point without being actively used, maintained, updated, or monitored
- **Shadow/Rogue API:** An API that was installed without the appropriate authorization or approval, leaving it as an open but ungoverned access point

ESG's research found that 26% of respondents said that one of their biggest challenges was discovering and remediating misconfigured APIs, specifically discovering shadow and zombie APIs. When explicitly asked about their level of concern with shadow/undiscovered APIs, the results again reinforced the challenge organizations face:

- 38% said these are a significant concern
- 49% said these are a moderate concern
- 13% said these are not a concern

Interestingly, when asked about a critical capability for an API security tool, the respondents distinguished "identifying APIs with sensitive data" and "Inventory of APIs":

- 50% felt that identifying APIs with sensitive data was very important
- 39% felt that offering an inventory of all APIs was very important

Realistically, these two offerings overlap. Without the ability to discover and inventory all APIs, organizations will have no visibility into API risk, as even the ones without sensitive data become an attack vector that threat actors can use to gain a foothold before moving laterally across networks and within systems.

LACK OF VISIBILITY

Discovery overlaps with the visibility issues that many organizations face. Without the ability to discover all APIs, they lack a comprehensive inventory, leading to issues with tracking all of them. Visibility requires additional information about the API's interactions to assess risk appropriately. Often, they lack insight into:

- What information passes through the API,
- How the API typically behaves,
- What risk the API creates.

Often, APIs exchange sensitive or regulated data without the controls and protections implemented in other channels, like databases or applications. For example, data communicated between a payment processing application and a corporate procurement system would include sensitive account information that a lack of API security could undermine.

CONTINUOUSLY UPDATING

APIs are a dynamic technology. Many organizations employ a DevOps methodology that automates code and application structure's continuous integration and delivery (CI/CD). For example, ESG's research found that

- 22% of respondents currently deliver new builds to production **multiple times per day**
- 9% of respondents currently deliver new builds to production **once per day**
- 23% of respondents currently deliver new builds to production **once per week**

With each application maintaining an average of 26 APIs, every newly delivered build impacts every application and its associated APIs. Again, the ESG research supports this, noting that

- 35% of respondents typically change or update their APIs **daily**
- 40% usually change or update their APIs **weekly**,
- 22% of respondents typically change or update their APIs **monthly**.

These dynamic updates create documentation and governance problems that undermine security. Not all developers provide API documentation, meaning organizations have a governance problem from the beginning. However, this problem grows since expecting developers to provide updated documentation for every change becomes untenable. These challenges are compounded by the inability to address older or different undocumented APIs.

Ultimately, most organizations have yet to determine whether they have the most recent version of an API, which may remediate a security vulnerability. With this insight, they can include API security in their overarching risk analysis and program.

TOO MANY PEOPLE

As a technology, APIs do not fit into any specific box. Although they enable connections to networks, they are not network devices. Meanwhile, although applications use them, they are not part of an application's functionality. These challenges around defining API technology equally apply to assigning responsibilities for managing their security.

Security

Logically, the security team would be involved in defending APIs from attacks. Typically, the security team's daily activities overlap with API security, including:

- **Network monitoring:** detecting and responding to anomalous network traffic that indicates potential data exfiltration
- **User access monitoring:** detecting and responding to anomalous access, like an abnormal location or time of day, that indicates an unauthorized user
- **Red teaming:** thinking like attackers to identify security weaknesses

DevOps

As security shifts left, the DevOps team becomes increasingly responsible for securing their code, including the APIs that they create and manage. With this in mind, developers have become far more knowledgeable about API security risks. When describing the development team's collective understanding of API security risks, ESG's research found:

- 71% responded with a high level of knowledge
- 22% responded with a good level of knowledge
- 8% responded with a limited level of knowledge

As security shifts left, many developers support the security team's imperatives by being responsible for:

- Selecting, implementing, and maintaining testing tools
- Selecting, implementing, and maintaining security tools
- Incorporating security checks at each stage of the pipeline
- Treating security configurations, policies, and controls as code artifacts

AppSec

Unlike DevSecOps, the application security (AppSec) team enforces secure coding practices across the application stack. AppSec capabilities that overlap with API security include:

- Identifying and tracing all third-party APIs used, including any dependencies
- Maintaining the Software Bill of Materials (SBOM) using Software Composition Analysis (SCA) tools
- Scanning for vulnerabilities in code with Dynamic Application Security Testing (DAST) and Static Application Security Testing (SAST) tools

Third-Party Risk Management (TPRM)

When integrating partner APIs into an environment, organizations must incorporate them into their third-party vendor risk management strategy. Organizations must consider third-party API risks in their procurement process and TPRM programs. For example, they should review the vendor's secure software development life cycle (SSDLC) practices. Additionally, an organization should test third-party APIs to identify known vulnerabilities or outdated components to mitigate risks.

Leadership

Even within the leadership team, identifying the responsible party can be challenging. Depending on the organization's structure, any of the following roles can be responsible for governing API security:

- Chief Information Security Officer (CISO)
- Chief Information Officer (CIO)
- Chief Technology Officer (CTO)
- Business unit leadership
- Quality assurance
- Software development leadership

The confusion over the role governing API security highlights a critical problem. Organizations may only incorporate API security as part of their overarching risk management programs with a direct line of reporting through the senior leadership team, like the CISO.

TOO MANY — *Yet Not Quite Right* — TOOLS

As API integration has exploded over the last few years, organizations have sought to wrap security around their use. However, traditional security tools focus on issues related to APIs rather than API security directly.

Based on research, organizations use different tools to manage various security functions across their API landscape. For example, when doing its research, ESG asked three questions about managing API security:

- How would you rate each of the following tools or processes your organization uses to discover and track your organization's APIs?
- How would you rate each of the following tools your organization uses to discover and remediate API coding errors?
- How would you rate each of the following tools your organization uses to stop or block attacks on APIs?

Across these three questions, respondents identified 17 different security tools, with only the following specifically focused on APIs:

- API gateways
- API security tools
- API specification conformance tools
- Web application and API protection (WAAP).

Common is an inability to discover APIs and only secure what they know. Additionally, many secure coding tools fail to need to be configured on a per API basis, which becomes overwhelming and cost-ineffectively time-consuming.

Next-Generation Firewalls (NGFW)

NGFWs filter packets based on applications and inspect that data to provide visibility into application and transport layer activity. From the application security perspective, they filter traffic and apply rules to block traffic from certain applications. From the network security perspective, they monitor for malicious activity and log it when detected.

However, at these layers, the data collected and analyzed fail to link back to the APIs themselves, creating a monitoring gap.

Web Application Firewalls (WAF)

WAFs are critical for establishing a strong and robust API security boundary. The WAF monitors and filters traffic between the application and the internet, protecting from typical attack types like:

- Cross-site forgery,
- Cross-site scripting (XSS),
- File inclusion,
- SQL injection
- The WAF works at the application layer, identifying well-known attacks based on signatures and patterns.

Despite their application security value, they were never built for or intended to monitor API security risks. Robust application security incorporating APIs requires augmenting these critical capabilities with technologies that can aggregate and correlate API activity to detect attacks at this vector over time.

API Gateway

The API gateway accepts a client's API request and then directs them to the appropriate service based on defined policies. They typically include the following capabilities:

- **Security policies:** authentication, authorization, access control, encryption
- **Routing policies:** Routing, rate limiting, request/response manipulation, load balancing, health checks, error handling
- **Observability policies:** metrics, logging

API gateways were built to manage traffic and maintain service uptime, meaning their metrics and tracing capabilities were never built for security. While they can provide alerts about activities that could indicate an attack, they lack the comprehensive detection and response capabilities that a robust API-security focused solution would provide.

Static Application Security Testing (SAST)

SAST reviews code for known vulnerabilities as part of SSDLC. They scan source code without executing it, looking for security weaknesses across the source code, third-party libraries, and dependencies.

While some new SAST tools may identify API ingress points, they often fail to identify and respond to unique authorization and authentication issues related to APIs.

Dynamic Application Security Testing (DAST)

DAST tests applications during runtime, seeking to identify vulnerabilities that attackers could exploit. Since they execute the code, they can provide visibility into whether threat actors can use an API security weakness during an attack.

Although they can provide some insights, these tests are limited to known application vulnerabilities, like the OWASP Top 10 Application Threats. Since they only test input/output points, HTTP response, and session management, they fail to provide holistic insight into API security risks.

Graylog API Security:

CONTINUOUS API THREAT DETECTION & INCIDENT RESPONSE

To mitigate risk and reduce the security problems arising from APIs, organizations need the right solutions to help them reap the rewards of these new technologies. Even organizations with a robust cybersecurity stack may find that they still have blindspots, so they need security solutions that enable them to aggregate all the log data necessary for comprehensive API security and monitoring. For comprehensive visibility into their security posture, companies need solutions that correlate data from across their complex environments so they can detect and respond to incidents faster.

Graylog API Security is continuous API security, scanning all API traffic at runtime for active attacks and threats. Mapped to security and quality rules, Graylog API Security captures complete request and response detail, creating a readily accessible datastore for attack detection, fast triage, and threat intelligence. With visibility inside the perimeter, organizations can detect attack traffic from valid users before it reaches their applications.

Graylog API Security captures details to immediately identify valid traffic from malicious actions, adding active API intelligence to your security stack. Think of it as a "security analyst in-a-box," automating API security by detecting and alerting on zero-day attacks and threats. Our pre-configured signatures identify common threats and API failures and integrate with communication tools like Slack, Teams, Gchat, JIRA or via webhooks.

ABOUT GRAYLOG

Graylog is a game-changing cybersecurity firm, revolutionizing the way organizations protect against cyber threats. Our solutions are crafted with the latest advancements in AI/ML, security analytics, and intelligent alerting, offering unparalleled threat detection and incident response capabilities. Graylog stands out by making advanced cybersecurity accessible and affordable, ensuring businesses can easily implement robust defenses against the evolving landscape of cyber threats, including the critical vulnerabilities associated with APIs. Our commitment to innovation and simplicity positions Graylog as the go-to partner for businesses seeking to enhance their cybersecurity posture without the complexity and high costs of traditional solutions. For more information on how Graylog can fortify your cybersecurity, visit our website at graylog.org.

www.graylog.org
info@graylog.com | 1301 Fannin Street, Suite 2000, Houston, TX 77002

graylog