TAG CYBER

REAL-TIME MITIGATION OF CYBERTHREATS TO APIs: AN OVERVIEW OF THE SALT SECURITY PLATFORM

EDWARD AMOROSO, TAG CYBER

TAG CYBER

REAL-TIME MITIGATION OF CYBERTHREATS TO APIs: AN OVERVIEW OF THE SALT SECURITY PLATFORM

EDWARD AMOROSO

The protection of APIs during the entire lifecycle for software applications has emerged as an essential security requirement. This is best done through discovery of APIs, mitigation of attacks, improvement of APIs and response to incidents. The Salt Security platform is used to illustrate this critical API security control.

INTRODUCTION

The need to protect application programming interfaces (APIs) has emerged as one of the most prominent aspects of application security. In fact, the task has become so central to the protection strategies of so many organizations that it is not hyperbole to refer to API security as one of the most accelerating aspects of enterprise security, in terms of its overall impact on cyber risk.

Many strategies exist to protect APIs, ranging from MITRE ATT&CK-based testing platforms for programmers to skills development courses on how to develop and use APIs securely. Given that existing technologies, such as API gateways and WAFs, cannot identify API attacks, the holy grail for API security, involves use of a platform that will discover APIs and then provide runtime security through prevention, detection and response to live threats. Such capability requires innovative technology to keep up with modern attack tactics.

In this report, we outline how this method of discovery, mitigation and response might be integrated into modern API architectures. We also show how the commercial Salt Security platform provides this type of protection for API deployments. The goal is to provide insight to enterprise security teams on how they can address this vital aspect of their overall cyber risk management program.

API SECURITY ISSUES

Cybersecurity issues with modern APIs have grown considerably, as applications continue to expand in scope and usage for the typical enterprise. APIs exist for internal and external applications supporting a variety of architectures; multiple deployments, including premise, cloud and SaaS; and across many phases of lifecycle development, including the continuous steps of DevOps and CI/CD.

As a result, cybersecurity experts have had to address a range of vulnerabilities that emerge in the context of API design, deployment and use. To help, many frameworks have emerged to support the identification and mitigation of API security threats. The OWASP API Security Project, for example, maintains a top 10 list of risks to APIs that can help developers and administrators to better identify their risk.

OWASP API Security Issue	Brief Explanation
API1 Broken Object Level Authorization	Need to check authorization for every function accessing data sources.
API2 Broken Authentication	Must ensure proper authentication token usage to avoid identity threats.
API3 Excessive Data Exposure	Should always minimize views of data to only what is required for the access.
API4 Lack of Resources & Rate Limiting	Need to restrict size and number of resources being requested.
API5 Broken Function Level Authorization	Must separate authorizations for administrative and normal functions.
API6 Mass Assignment	Care is required, such as whitelisting, when binding client-provided data to models.
API7 Security Misconfiguration	Need to avoid insecure configurations such as defaults or ad hoc setups.
API8 Injection	Common issues emerge when unstructured data can be passed along to commands.
API9 Improper Assets Management	Need to track API versions and maintain accurate documentation.
API10 Insufficient Logging & Monitoring	Should provide sufficient logging to address attack persistence.

Figure 1. OWASP API Security Issues

The advantage of the OWASP API top 10 list is that it includes sensible advice from practitioners based on experiences dealing with actual vulnerabilities that have been cited in APIs. It offers a useful framework for guiding best practices in API design and implementation. The challenge, however, is that it requires diligence from developers and administrators, which can be uneven, depending on their skills and management incentives in the local environment.

Of course, it must also be acknowledged that even with diligent developers and administrators, malicious actors can abuse an API based entirely on how that interface was intended to be used. This is a general conundrum in cybersecurity, and is often best resolved through runtime detection and analysis to determine if the behavior observed matches up with what is considered an acceptable or typical use.

REAL-TIME THREAT MITIGATION FOR APIS

A powerful means for dealing with attacks to APIs involves the collection of live data for the purpose of detecting and mitigating actions that are indicative of malicious intent. The OWASP API list offers useful insight into the types of behaviors that might be detected in such real-time mitigation, but other actions might also be identified, using profile analysis or even machine learning models.

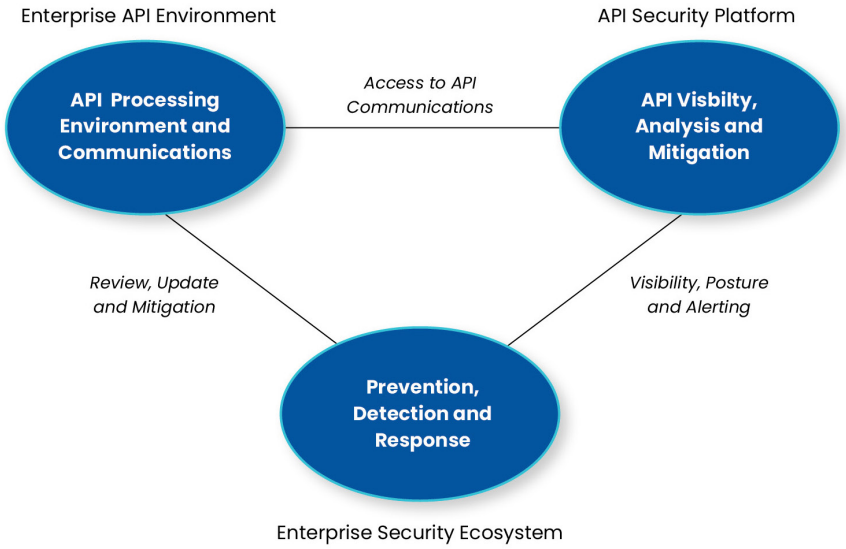

Figure 2. API Security Mitigation Approach

The mitigation of live attacks on APIs requires two preliminary steps. First, the organization must establish a means of detecting and cataloging the traffic of all applicable APIs for analysis. This collection provides the data for subsequent analysis. Second, the organization must identify any vulnerabilities that might exist in the API, through static review, API testing or other means. These steps reduce risk in advance of the dynamic coverage.

Algorithms for mitigating attacks to APIs include the familiar types of algorithmic options known to all cybersecurity experts. Certainly, signature-based solutions are possible, generally based on the OWASP list or other best practices. But since most API attacks are rooted in business logic flaws, signatures are far from sufficient—behavioral analytics must be performed to detect anomalous or unusual activity. In addition, machine learning models should be developed from training sets to detect attacks.

In the next section, we outline how one popular commercial platform from Salt Security implements these security concepts for practical enterprise API deployments. The company's commercial solution is part of a significant push in enterprise and government across all sectors to reduce API cyber risk, as more emphasis has shifted toward cooperation between applications and workloads across networks.

OVERVIEW OF THE SALT SECURITY PLATFORM

Founded in 2016 by former Israeli Defense Force members, Silicon Valley-based Salt Security provides an API protection platform that uses advanced technology to automatically and continuously identify and secure APIs. The solution is designed to complement web application firewalls (WAFs) or API gateways through use of a big data engine that utilizes machine learning to protect APIs.

Salt Security Platform Features

Several of the major functional areas supported by the Salt Security platform for enterprise security teams are outlined below.

- *API Discovery*—The Salt platform is designed to automatically discover and continuously monitor APIs of interest. The Salt system collects a copy of all API traffic, including granular data and tracking changes as the API environment evolves according to business needs. The inventory that results from this step allows for identification of shadow APIs, the APIs not released through gateways or properly documented.
- *API Data Exposure Protection*—The Salt platform reviews API data to understand the type of information being used (e.g., Social Security numbers) to flag exposure issues. This protection process also helps to identify proprietary or other sensitive data that might be shared via an API. The platform generates alerts when it detects instances of sensitive data exposure.
- *Mitigation of Live API Attacks*—The Salt platform utilizes collected API data to analyze (using AI tools), correlate and draw conclusions about possible attack conditions. The goal is to provide a means of alerting teams to live conditions or automatically mitigating live attacks as they are occurring. The platform is able to detect the reconnaissance behavior of bad actors learning the APIs, and leverages web application firewalls (WAFs) or other inline devices in the customer's environment to block the attacks before attackers can reach their objective.
- *Sharing of Remediation Insights*—The Salt platform provides insights from pre-prod and runtime environments to harden APIs. The platform performs API design analysis, comparing static OAS/Swagger files to API security best practices, looking for gaps. It also provides attack simulation to detect business logic flaws in pre-prod. And it uses attackers as pen testers, so runtime learnings, with bad actors' minor successes, become another input for sharing remediation insights with dev teams so they write improved API design security.

In addition, the platform integrates with existing security tools and workflow systems, including Apigee, Slack, Jira, NGINX, Cloudflare, MuleSoft, Kong, F5 and AWS. It can also generate complete documentation based on the APIs and sensitive data found in runtime. These testing, integration and documentation capabilities provide useful assistance for both cybersecurity objectives based on threat and compliance obligations based on security or privacy frameworks.

Salt Security Architecture

The Salt Security architecture includes layered collection, processing, discovery and reporting. Each layer interacts with the goal of reducing API security risk based on ingested data. Positioning of the Salt platform for most environments involves a mirroring of the live API traffic via agentless collection in the API gateways, microservices, load balancers, edge processing, server and cloud infrastructure that interact via APIs with clients. The Salt platform does not deploy inline, to avoid any API performance degradation.

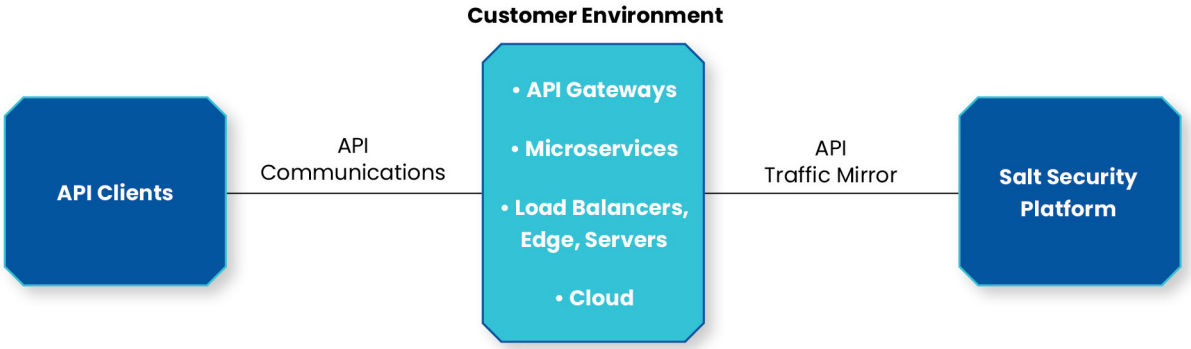

Figure 3. Salt Security Mirrored API Data Collection

Salt Security Contextual Analysis

The contextual analysis is supported by a patented API Context Engine (ACE), which takes ingested API communications traffic in mirrored form into a processing engine that utilizes machine learning to build models that can predict or detect evidence of attacks. This ACE operates at cloud scale using big data analytics, runs continuously, and can be set up to enable automated mitigation of attacks. Note that the Salt team contends that API attack detection requires cloud-scale big data—the amount of context available in an on-prem solution is insufficient to detect many of today's sophisticated API attacks.

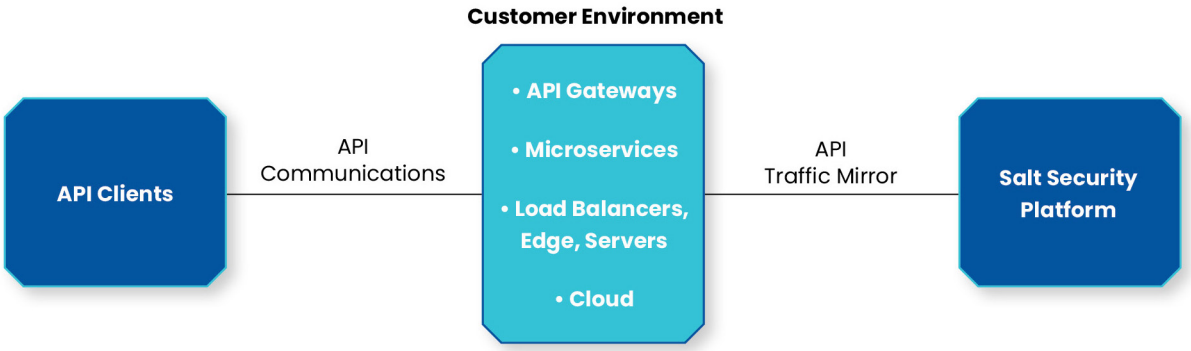

Figure 4. Processing Inbound API Traffic

The Salt Security platform can also be engaged during API development to detect vulnerabilities and gaps during build and staging and to remediate any detected issues, either in OWASP-type flaws or business logic. Runtime protections keep APIs protected regardless of when dev teams can integrate the remediation insights. The combination of shift-left and runtime protections provides the kind of full-lifecycle protections needed to provide API security for enterprise teams.

ABOUT TAG CYBER

TAG Cyber is a trusted cyber security research analyst firm, providing unbiased industry insights and recommendations to security solution providers and Fortune 100 enterprises. Founded in 2016 by Dr. Edward Amoroso, former SVP/CSO of AT&T, the company bucks the trend of pay-for-play research by offering in-depth research, market analysis, consulting, and personalized content based on hundreds of engagements with clients and nonclients alike—all from a former practitioner perspective.