SALT

# Protecting APIs in Financial Services

SOLUTION BRIEF

SALT

# Overview

**APIs are at the core of every financial services company powering applications, fueling rapid innovation, and enabling connections across the complex ecosystem of services and partners. Arguably, financial services companies could not thrive without APIs today. The same could be said about many businesses, with digital transformation efforts driving organizations to modernize and stay relevant or innovate with new services to enter new markets.**

APIs were around long before digital transformation was even a thing, but thanks to digital transformation, API growth has exploded in recent years. Today, APIs are found globally in organizations of every size and industry, powering customer-facing mobile and web applications, connecting microservices-based development environments, and enabling the exchange of data and services with partners.

There are more APIs in use than ever, and the number of APIs found in an average financial services company can number in the hundreds if not thousands. This number continues to grow exponentially with each new use case or service and explodes with connections across the financial ecosystem with partners and technology suppliers.

The rapid proliferation of APIs has not gone unnoticed by attackers who, in recent years, have made APIs the primary target for their efforts. Attackers know that behind financial services APIs lies a rich target of valuable data and services that, in many cases, lack proper protection.

The analyst firm Gartner has been predicting for years that "By 2022, API abuses will move from an infrequent to the most frequent attack vector, resulting in data breaches for enterprise web applications." Gartner's latest prediction estimates that "By 2024, API abuses and related data breaches will nearly double." The Q3 2021 Salt Security State of API Security report found malicious API traffic grew 348% across our customer base in the first six months of 2021 alone, compared to overall API traffic, which increased 141%.

Protection of APIs requires a new approach that considers the unique nature of each API, unique vulnerabilities, and the unique methods attackers use to target and exploit APIs to perform fraud, access data, and disrupt services.

## Financial Services APIs

Consumer expectations and demands have been rapidly evolving for the better part of a decade. The days of walking into a bank branch to make a transaction are quickly fading as consumers become increasingly comfortable with online services and mobile devices for banking. Consumers today expect more from their financial institutions and demand integrated services and the ability to connect their financial lives as they see fit.

APIs are the intellectual property (IP) that enable every financial service company to innovate, deliver customer-facing services, and connect to an ecosystem of partners. Without APIs, it would be impossible for financial services companies to evolve to meet changing consumer expectations and innovate to remain competitive in an increasingly digital world.

Financial services APIs, by nature, expose sensitive data such as account details and personally identifiable information (PII), and with CI/CD development practices, change is constant as a result of rapid innovation. These characteristics of APIs create new challenges for security teams who must reassess security strategies and the tools used to protect critical services, data, and customers.

## Open Banking Driving API Usage

Open banking originated in Europe, where it is most evolved, but the concepts have reached nearly every corner of the globe, with financial services companies taking notice. While many companies do not yet fall under specific open banking regulations, the desire to reach a global customer base and meet the changing expectations of consumers has driven companies to use its constructs as a framework for modernization.

Open banking aims to give consumers more control over their financial data and, ultimately, their financial lives. Open banking not only creates a push for financial services companies to modernize but also creates new opportunities to acquire new customers, enable new revenue streams, and remain competitive through innovation.

APIs are at the center of open banking, enabling financial services companies to standardize how they create and connect to the financial ecosystem and exchange financial data. To a large degree, open banking has ushered in the democratization of banking across the globe, and it's all been possible thanks to APIs.

SALT

## Just Because it's a "Standard" Doesn't Make it Secure

While open banking defines standards around how APIs should be structured to enable predictable integrations and communications, it provides no standard to meet the majority of security requirements for APIs. Much of the focus for security in open banking centers on applying basic controls such as authentication, authorization, and encryption, to specific, standard financial functions. The overall scope of open banking does not cover the full potential of financial services APIs, and the basic controls fall short of meeting the security challenges that APIs create.

Best practices help guide the implementation of controls like authentication and authorization, but these best practices leave a lot of room for interpretation. Since more than one way exists to implement a service through an API, and each organization will take a different approach, the results are unique APIs with unique business logic for each organization.

With each organization having APIs with unique business logic, it's impossible to standardize how authorization should be applied. Furthermore, considering many services under the open banking umbrella are a combination of multiple created and consumed APIs, the result is unique APIs with unique logic connecting to other unique APIs with unique logic. This combination of APIs results in a multiplicative effect with new logic and ultimately unique vulnerabilities.

## Goals and Impact of an API Attack

Attackers have realized that APIs make an attractive target and have shifted their focus to APIs as their primary attack focus. Behind each financial services API lies valuable data that includes sensitive financial and customer information. APIs also expose the inner workings of services and provide an opportunity for attackers to understand and manipulate the underlying business logic to probe for and exploit vulnerabilities.

The Salt State of API Security Q3 2021 report found that in the first six months of 2021, overall API traffic increased 141% across the Salt customer base, while malicious traffic grew three times that rate at 348%. A clear sign that attackers are increasingly focusing their attack activity on APIs.

The financial services industry has seen a steep rise in the volume and sophistication of API attacks, with big-name companies in the headlines for breaches and vulnerability disclosures. Attackers targeting financial services APIs are typically motivated by these common goals:

- **Account takeover (ATO)** - Authentication APIs are targeted with credential stuffing and brute force attacks allowing attackers to take over customer accounts and drain funds.

- **Fraudulent transactions** - Attackers exploit API vulnerabilities and manipulate API logic to perform fraudulent transactions.

- **Data exfiltration** - APIs vulnerabilities are exploited to gain unauthorized access to sensitive data such as account details and other PII. A sophisticated attacker can automate attacks to slowly scrape data from thousands of users or find a vulnerability that enables access to an entire database and large stores of data.

- **Service disruption** - Unlike Distributed Denial of Service (DDoS) attacks that use volumes of data, a single, subtly crafted API call can slow down or make services unavailable. These types of Denial of Service (DoS) attacks are missed by DDoS protections and rate limits.

  Should an attacker successfully execute any of the above attacks, the impact to your business can include one or more of the following:

- **Impact on revenue** - Depending on the country where a customer account exists, financial services companies may be responsible for the full amount of funds lost due to a fraudulent transaction. In many jurisdictions, the consumer has limited liability and, in some, no liability at all, meaning that the financial services company could be liable for the total amount of loss.

- **Compliance or other regulatory fines** - Organizations can face fines according to compliance mandates if customer data is lost or exposed. Under the GDPR, fines for exposing private data can reach up to $24.1 million or 4% of annual global turnover, whichever is higher. Some mandates will also assess penalties until organizations resolve issues associated with the data loss.

- **Impact on customer confidence and brand reputation with partners** - Reputation is everything in the highly competitive financial services market. The viability of the business depends on customer acquisition, customer retention, and connections across the ecosystem of partners. A breach can significantly impact confidence, and with numerous options, customers may opt for other services, and partners may look to other providers.

## Challenges Protecting APIs

APIs are characteristically different from traditional applications and present unique security challenges. The following factors contribute to the unique challenges of protecting financial services APIs:

- **More APIs used than ever** - Financial services companies use APIs everywhere to enable customer-facing applications, connect to partner services, and power microservices-based development environments. Each new service comes with a new set of APIs, resulting in an attack surface that continues to expand exponentially.

- **More interconnected services** - Modern applications are composed of multiple services and connect to those services using APIs. This interconnectivity creates a complex mesh of application logic and often results in unique vulnerabilities and unrealized risk.

- **More data exposed** - The nature of financial services applications is to exchange sensitive financial and customer data. This factor has made financial services APIs a primary target for attackers and a high-stakes asset for defenders to protect. A challenge for defenders is knowing where APIs expose sensitive data and if that exposure is necessary. Another challenge is knowing when sensitive data exposure changes, which might result from an update to an API. Insight into exposure and changes to exposure is critical to understanding the attack surface, assessing risk, and meeting compliance requirements.

- **Frequent changes** - APIs are a perfect fit for agile development practices that enable rapid innovation. Developers build new applications by combining multiple APIs and update APIs quickly to add new features and functionality. Rapid development creates a moving target for security teams that struggle to understand the API landscape to assess risk and align protections. Traditional security approaches that depend on manual efforts can't keep up to meet the needs of rapidly changing APIs.

## Why Traditional Approaches Don't Protect APIs

Financial services companies employ best practices during development to minimize vulnerabilities in production code. Despite these efforts, it's not realistic to think that DevOps teams can eliminate all vulnerabilities before deployment. Another important consideration is that DevOps teams need to move rapidly, and security cannot inhibit this speed of innovation.

Distributed and remote teams add to the complexity of identifying vulnerabilities in the development phase. Even with mature DevOps practices, gaining access to all code, applications, and systems can be challenging. Working with outsourced teams further complicates efforts since supplier attestation only goes so far, and digital supply chains are complex.

Additionally, the application must be released and under production load to identify some vulnerabilities. With the inevitability of vulnerabilities making their way to production, runtime protection is essential to prevent attackers from exploiting vulnerabilities.

Most financial services companies have sophisticated runtime security stacks with multiple layers of security tools such as bot mitigation, WAFs, and API gateways. These traditional tools can provide some foundational security capabilities and protection for traditional applications; however, they lack the architecture and context needed to identify and stop attacks that target the unique logic of each API.

- **Authentication and authorization** are essential foundational capabilities for security; however, the Salt Security State of API Security Report Q3 2021 found that 95% of API exploits happen against authenticated APIs. Also, many attackers take advantage of commonly found authorization flaws - defined as Broken Object Level Authorization (BOLA), the number one threat on the OWASP API Security Top 10 list.

- **Encryption** such as with TLS is important to protect data in motion against person-in-the-middle attacks; however, attackers have easy access to unencrypted data at a client endpoint using common tools such as OWASP Zed Attack proxy or Portswigger Burp Suite.

- **Rate limiting** is another common security approach, but these controls do nothing to identify or stop attackers who use subtle methods to stay under policy limits.

- **Bot mitigation** offerings commonly rely on a combination of techniques that include client-side code and CAPTCHAs to detect and deter attacks. These approaches can negatively impact the application experience for legitimate users and sophisticated attackers can reverse engineer client-side code and bypass CAPTCHAs rendering these approaches ineffective.

To attack your APIs, attackers reverse engineer them and use reconnaissance techniques to understand the structure and unique logic. Reconnaissance takes time and consists of low and slow activity and subtle methods to probe an API to map the structure, determine the application components in use, understand the API logic, and look for vulnerabilities.

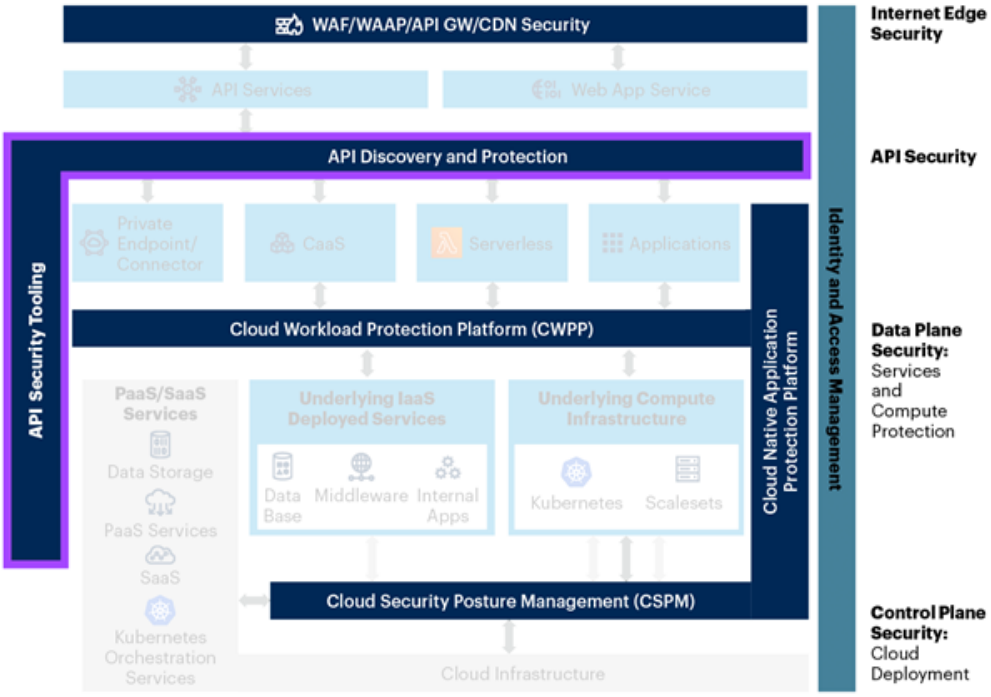
Attacker reconnaissance activity looks like normal API traffic to traditional tools such as WAFs, API gateways, and other proxy-based solutions. The architecture of these tools limits them to inspecting transactions one at a time, in isolation, and beyond rate-limiting; they depend on signatures to detect well-known attack

patterns such as Cross-Site Scripting (XSS) and SQL Injection (SQLi). If the transaction does not match a known attack signature, the WAF will send it through. Since each API is unique with unique vulnerabilities, signatures cannot help prevent API attacks. Tools such as WAFs and API gateways not only miss the attacks that target unique API vulnerabilities, they also miss the hallmarks of an attacker's reconnaissance activities.

## What is Needed to Protect Financial Services APIs

In today's highly competitive market, financial services companies need to move at a rapid pace, and to do so requires security that does not impede innovation. Security can and must enable developers to move at the speed the business demands. APIs require a new approach to security with dedicated solutions leveraging architecture that goes beyond traditional tools that are limited to looking at individual transactions in isolation.

In a recent research note, Gartner has acknowledged the need for dedicated solutions by adding API Security as a distinct tier in their updated Security Reference Architecture. This new tier sits between traditional tools such as WAFs, WAAPs, API gateways, and CDNs that protect the edge and tools that protect the data and control plane. With this new architecture, Gartner affirms that these traditional tools leave gaps and are not enough to protect APIs.



API security requires a new architecture based on big data to capture all API traffic and artificial intelligence (AI) with machine learning (ML) to continuously analyze the large volumes of API traffic. Continuous analysis of API traffic is the only way for a solution to understand normal behavior for each unique API and gain the context required to identify subtle deviations and pinpoint attackers.

APIs also require security at every step of the lifecycle, another limitation of traditional tools. With an architecture of big data, AI/ML, and continuous analysis of API traffic, a solution can provide benefits across the entire API lifecycle. Benefits include up-to-date attack surface visibility, early attack prevention, and insights to enable continuous security improvement. The following are critical capabilities of an API security solution:

## Discovering APIs

An accurate view of the attack surface is essential to inform your security strategy but can be challenging with constantly changing APIs. Consider a fast-moving development team tasked to release a new customer registration API. Focused on project deadlines, the team circumvents the official process, and the API is not added to the organization's API management tool and is not visible to the security team.

A solution must discover all APIs, including unknown shadow APIs, as in the example above, and zombie APIs that should be deprecated. Discovery must be automatic and continuous to keep up with the constant release of new and updated APIs and must cover all customer-facing, partner-facing (provided and consumed), and internal APIs.

Knowing an API exists is not enough. Understanding each API at a granular level is critical to understanding the intended functionality, assessing risk, and determining if the API exposes sensitive data such as personally identifiable information (PII). Automatic and continuous discovery helps ensure that the view of the attack surface and sensitive data exposure is always up to date.

## Stopping API Attacks

Attackers targeting APIs use subtle methods to uncover and exploit vulnerabilities. Consider fraudsters attempting to redirect ACH transfers to alternate, unauthorized account numbers. The fraudsters could exploit a commonly found API vulnerability and manipulate routing numbers by changing only a few API parameters.

A solution with an architecture combining big data, AI, and ML can capture all API traffic to create a baseline of normal activity and look for deviations. In the case

above, a solution would identify the subtle routing number manipulation and flag the activity.

Identifying deviations alone is not enough, so the correlation of activity is required to reduce false positives by separating benign deviations that might result from API changes and deviations that are part of the fraudsters' reconnaissance activity.

During reconnaissance, the fraudsters probe the API to understand the structure and uncover the vulnerability in the example above. This activity creates several deviations from the normal baseline, and by correlating those deviations, a solution can build a profile for the fraudsters and assess the risk of their overall activity.

Unlike traditional tools that can only identify known malicious traffic, a solution using context and correlation can detect the fraudsters' suspicious activity and stop them early in their process before an attack is successful.

### Eliminating Gaps

DevOps teams play an essential role in security, but any software will inevitably release containing gaps despite teams employing development best practices and leveraging scanning tools. APIs are no different. Arguably, financial services APIs are more susceptible to gaps because financial services APIs go hand in hand with agile development practices and frequent release cycles, meaning that dev teams might compromise security to meet tight schedules.

Runtime protection is critical to prevent exploitation of any vulnerability that makes it to production. But relying solely on runtime protection leaves you in a situation of playing a virtual game of whack-a-mole. Gaps must continually be identified and eliminated by dev teams to improve API security.

In the example above, a runtime API security solution will block fraudsters and learn from their activity as they probe and manipulate the API. These learnings provide insights into the vulnerabilities unique to that API and are valuable to help development teams prioritize and eliminate gaps quickly.

In addition to runtime insights, an API security solution must analyze APIs to identify gaps before an attacker finds them and enable developers to proactively eliminate potential vulnerabilities while simultaneously sharpening their API security best practices.

## Protecting Financial Services APIs with Salt

The Salt Security API Protection Platform secures the APIs at the heart of all financial services applications. Our solution is built on an architecture of big data,

AI, and ML to deliver the context you need to protect your APIs across build, deploy, and runtime phases. The platform collects API traffic across your entire application landscape and makes use of big data, AI, and ML to help you see all your APIs, stop attackers during the early stages of an attempted attack, and share insights to improve API security posture. The Salt platform enables organizations to:

- **Discover APIs automatically and continuously** to find and catalog all APIs, including new and unknown (shadow and zombie) APIs. In addition, Salt identifies sensitive data exposure to help teams understand risk and meet compliance requirements.

- **Identify gaps in documentation** that include incomplete and inaccurate details. Salt research has shown a common gap of 40% when comparing manually created documentation with running APIs. Closing these gaps can help with API reuse and, more importantly, enables a complete understanding of risk.

- **Eliminate vulnerabilities early** in the development cycle. By analyzing documentation, Salt also finds potential issues and provides valuable insight to developers to help them eliminate vulnerabilities before production deployment.

- **Protect APIs in runtime** by leveraging big data, AI, and ML to gain the context needed to prevent API attacks. Salt analyzes all API activity to create a baseline of normal behavior and identifies activity that falls outside of the baseline. By correlating activity, Salt can differentiate between normal discrepancies in traffic, resulting from user mistakes or API changes, and malicious attacker activity. This approach enables organizations to identify and prevent attacks early during the reconnaissance period.

- **Achieve continuous improvement** of API security by using attackers like penetration testers. In addition to stopping attackers, Salt uses attacker activity to identify potential vulnerabilities found at runtime. These insights are provided to engineers to help them understand gaps, prioritize remediation efforts, and eliminate vulnerabilities efficiently to support a model of continuous improvement of API security.

Salt has a flexible, easy model for integration. The platform deploys in minutes with no agents, no changes to application code, and no configuration. With the largest number of ecosystem integrations, Salt supports the broadest set of use cases and application environments and spans internal, external, and third-party APIs.

SALT

## Salt Security Financial Services Customers


## Ally Bank protects business-critical APIs with Salt

For Ally Bank, the key challenge was that its business-critical API-based services required protection that already deployed WAFs, bot mitigation tools, and API gateways could not provide. Key Salt use cases:

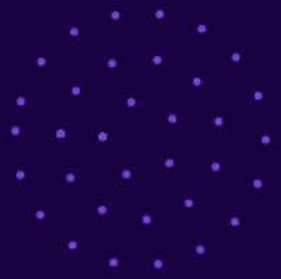
- **Attack prevention** - Salt identifies and stops attacks missed by the company's WAF and API gateway including the threats defined in the OWASP API Security Top 10
- **Remediation insights** – Salt creates remediation insights based on identified attack activity that dev teams use to eliminate API vulnerabilities and improve security
- **Compliance** - Salt identifies Zombie and Shadow APIs as well as exposure of sensitive data such as PII

**Salt Security** – Securing your innovation

Salt Security protects the APIs that form the core of every modern application. Its patented API Protection Platform is the only API security solution that combines the power of cloud-scale big data and time-tested ML/AI to detect and prevent API attacks. By correlating activities across millions of APIs and users over time, Salt delivers deep context with real-time analysis and continuous insights for API discovery, attack prevention, and shift-left practices. Deployed in minutes and seamlessly integrated within existing systems, the Salt platform gives customers immediate value and protection, so they can innovate with confidence and accelerate their digital transformation initiatives.

SB-ProtectingFinServAPIs-10202022

Request a Demo today!
info@salt.security I www.salt.security



SALT

# Securing your Innovation.