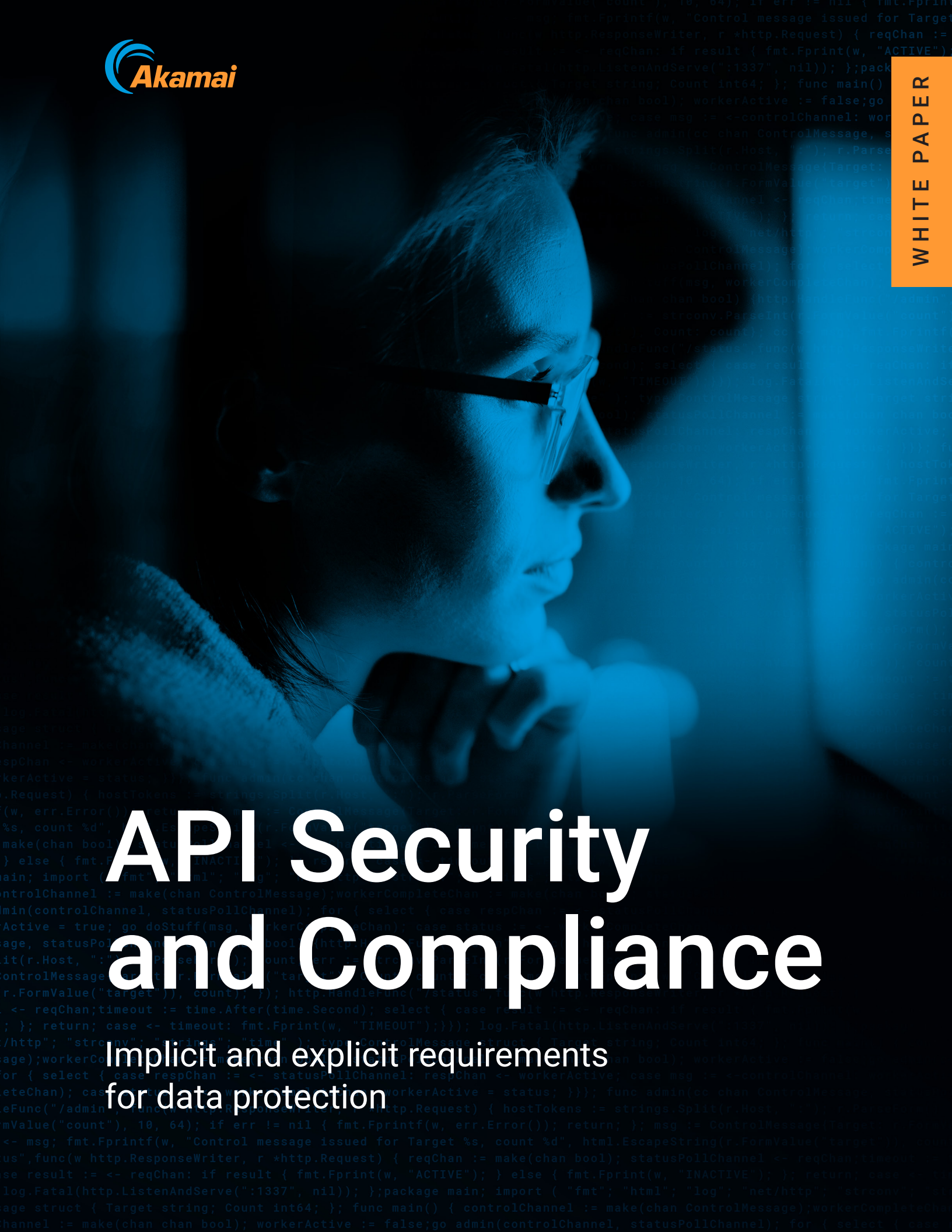
Akamai

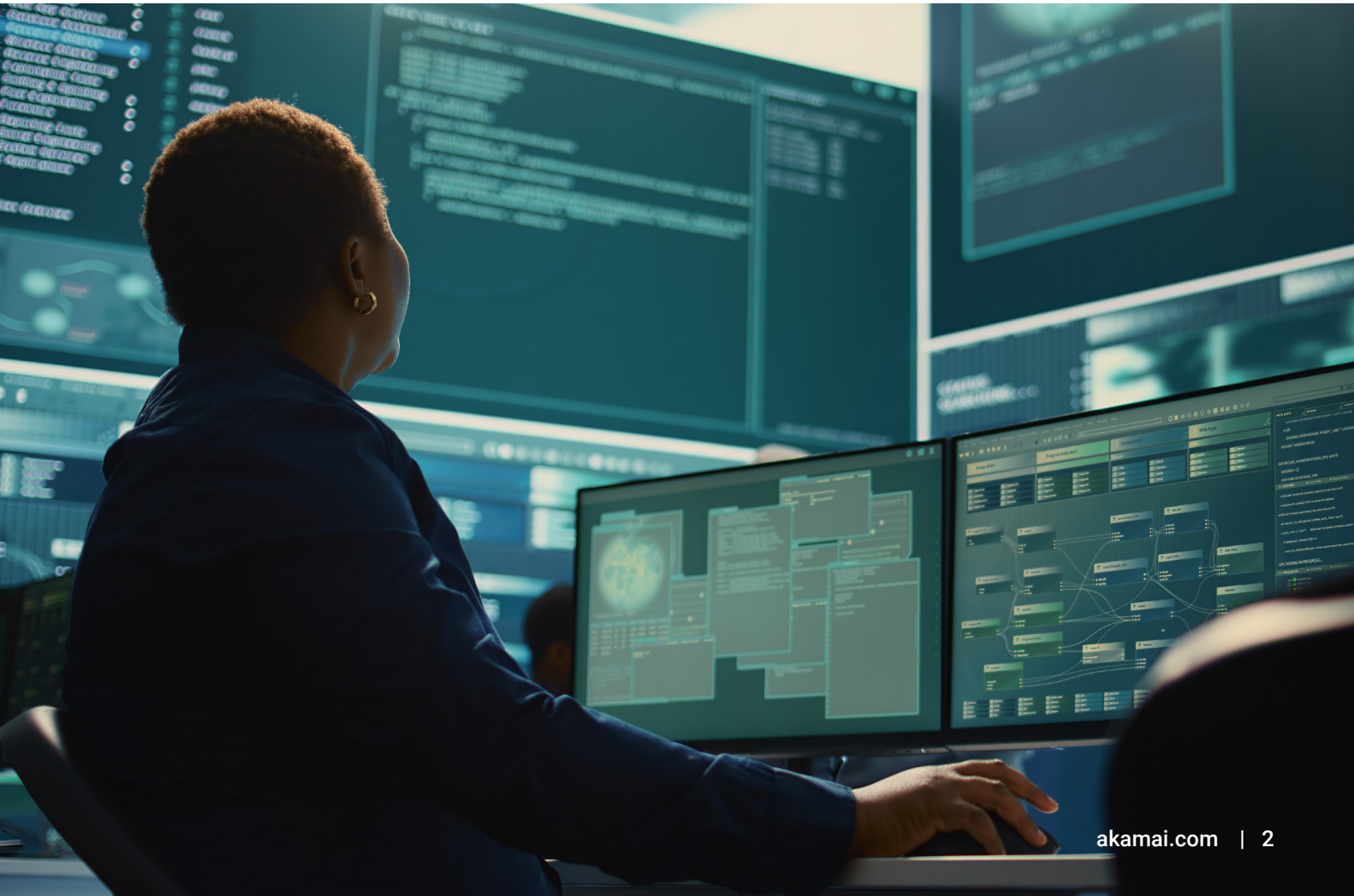
WHITE PAPER



API Security and Compliance

Implicit and explicit requirements for data protection

In this report



Introduction

Demonstrating compliance with data protection regulations has traditionally meant expending large amounts of energy and resources to keep up with mostly familiar risks. But that's changing. Today's attack surface is evolving fast to include threats that most enterprise compliance programs aren't fully accounting for. That's partly because the regulatory bodies themselves can't always keep pace and be explicit about every facet of coverage needed to prevent breaches.

This is the case with API protection. Every time a customer, partner, or vendor engages with your business digitally, there's an API behind the scenes facilitating a rapid exchange of information that often includes sensitive data. Attackers now know that they can simplify their strategy to steal that data by directly targeting APIs.

You may have already seen new language in regulations indicating the need to inventory, assess, or secure APIs. But even when specific language about APIs isn't included, the fact that they have become a clear attack vector *implies* that their adequate protection is required.

The emergence of APIs as a major compliance issue is not surprising. Exposed or misconfigured APIs are prevalent, easy to compromise, and often unprotected. And just one breached API can result in millions of records being stolen. The numbers speak for themselves:

- Seventy-eight percent of organizations have experienced an API security incident.[1]
- Forty-four percent have been fined by regulators for API security incidents.[2]

How does this affect your compliance program? Regulators need to see that your organization is taking measures to protect all access points to sensitive data. This means you need to demonstrate your organization can:

- Account for every API, including elusive shadow APIs
- Uncover and fix any API vulnerabilities
- Apply controls tailor-made to prevent API-centric data breaches

This white paper explores the nature of growing API risks, highlights six examples of regulations and frameworks that require API protections (either explicitly or implicitly), and offers advice on how to meet compliance requirements through API security best practices.

1., 2. Akamai Technologies, "The API Security Disconnect," 2023

Understanding API risks

APIs live at the core of your enterprise's digital products, services, and cloud environments. Their constant access to data makes them both a revenue driver and an operational risk. The trouble is, most enterprises — even those with mature security programs — are not prioritizing API-related threats to the degree they focus on other threats, such as phishing or ransomware.

Some organizations rely on API gateways and web application firewalls (WAFs) for baseline API protection, but these tools aren't designed to provide the degree of visibility, real-time protection, and continuous testing that specialized API security solutions can provide. Here's why these tools aren't enough:

- API gateways and WAFs can only observe *managed* API traffic that is routed through them.
- They can't protect unmanaged APIs, which analysts predict will make up nearly half of a typical enterprise's API ecosystem by 2025.
- As a result, security teams are not fully prepared to protect the fastest-expanding portion of their attack surface, knowing little about where APIs are routed, how they're configured, what kinds of sensitive data they exchange, and the risks they pose.

Protecting user information is a priority for regulators, and they levy severe fines for companies that fail to reasonably secure their customers' data from unauthorized access. Considering that only 4 in 10 security professionals with full API inventories know which of their APIs return sensitive data[3] and that many API calls come from attackers testing for vulnerabilities, data breaches via APIs will only increase — especially because API attacks are currently quite easy to conduct.

3. Akamai Technologies, "The API Security Disconnect," 2023

Four API attacks with compliance implications

How can an API breach affect a company's compliance posture? Here are a few examples:

- A popular project management application was compromised by an attacker who exploited an API endpoint lacking authentication controls. The attacker breached the API, gained unauthorized access to information on millions of users, and months later, leaked over 21 GB of data — including email addresses and board memberships — on the internet.
- More than 11 million customer records of a large telecom firm were exposed, reportedly because of an API that was unknowingly exposed to the internet and didn't require authentication. Attackers breached the API, saw it lacked a unique identifier, guessed its ID number, and easily requested sensitive data.
- A social media company was reportedly hit twice in recent years by a scraping tactic made possible through improper API use. In the first instance, private data was scraped from 500 million user profiles and then sold. In the second instance, an attacker created a database including phone numbers and salary data scraped from 700 million users.
- This same technique was used against another social media company to exfiltrate data on millions of users. The business received a $5 billion fine because a third-party vendor used the company's API to gather sensitive data. It didn't matter that the vendor abused the API; the company *itself* was fined because it failed to monitor its application.

Six examples of regulations and frameworks entailing API security

In many regulations and frameworks, APIs aren't necessarily mentioned by name, but the requirements clearly focus on securing the applications and infrastructure within which APIs operate. For example:

- Payment Card Industry Data Security Standard (PCI DSS) v4.0 offers guidance to confirm that an organization's software securely uses functions of external components. This includes APIs transmitting payment data from a mobile app to a bank's system.
- The NIST Secure Software Development Framework provides guidance on producing well-protected software, securing it continuously, and responding to vulnerabilities. APIs are at the core of software development.

In many cases, regulations suggest loosely defined objectives for securing data, such as the General Data Protection Regulation (GDPR)'s requirement for "appropriate security measures." Your APIs might receive millions of calls per day to serve up that data, from customers *and* attackers. It's on you to determine what security controls are required — and then demonstrate how they'll work.

Let's take a closer look at regulations and frameworks with direct implications for your API ecosystem.

1. PCI DSS v4.0

Created by the Payment Card Industry Data Security Council, PCI DSS has become a global standard for protecting payment data. If your business accepts major credit cards and processes, stores, or transmits cardholder data electronically, you must comply.

The original version's requirements cover security mainstays that are as important now as they were when PCI DSS was published in 2006, such as assigning access to system and cardholder data on a need-to-know basis and defining access requirements by role.

However, with PCI DSS v4.0 in effect, enterprises need to adapt their compliance programs to account for threat actors who frequently target the thousands of APIs living within payment technologies. Overall, PCI DSS v4.0 is centered on four key objectives:

1. Continuing to meet the security needs of the payment industry
2. Advocating for security as a continuous process
3. Giving enterprises flexibility (e.g., new tools, new controls) in how they meet requirements
4. Enhancing validation methods and processes

PCI DSS v4.0 requirement 6.2.3 centers on the need for organizations to review their bespoke custom application code (i.e., code developed by a third-party vendor but not standard commercial off-the-shelf applications) to ensure no vulnerabilities are released into production. Specific to APIs, this requirement offers guidance to confirm that an organization's software securely uses external components' functions (libraries, frameworks, APIs, etc.). Requirements like these underscore the key role APIs play in the broader software supply chain — and what it takes to protect it.

APIs have become the default method for connectivity and data exchange in modern application environments. With that in mind, securing APIs from both a pre-production (shift-left) and post-production (shield-right) perspective is essential to making your digital business resilient against attacks. Here are some API security best practices to follow for compliance with requirement 6.2.3:

- Confirm usage of API-based components and their security posture (e.g., find misconfigurations leading to vulnerabilities, including use of weak encryption ciphers).
- Validate normal and expected behavior of API usage and implement controls to block suspicious actors from abusing your systems (e.g., check the application's behavior to detect logical vulnerabilities).
- Detect third-party frameworks used to power your APIs, determining any that may be outdated and vulnerable.
- Build a full inventory of all APIs, including the different versions you are running; this provides insight into backdoors and potential undocumented features you need to manage.
- Validate the security of your API code and avoid putting any API-related vulnerabilities into production.
- Implement secure coding best practices for APIs, allowing you to adopt a programmatic approach to securely deliver code on a continuous basis.

2. General Data Protection Regulation (GDPR)

The GDPR is a piece of European Union (EU) legislation that aims to strengthen and unify data protection for individuals within the EU. However, the GDPR is not limited to EU-based companies; any organization offering consumer goods or services in the EU must comply.

The regulation stipulates that personal data is information that can be linked or connected to an individual. Data regulated under the GDPR may include an individual's name, contact information, banking and financial data, and medical information. On the more technical side, covered data also includes geolocation data such as IP addresses and web cookies.

What does this mean for API security? Whether you're developing applications, microservices, or Internet of Things (IoT) devices, the APIs living at the heart of these technologies are likely exchanging GDPR-regulated data. Therefore, organizations developing internet-accessible APIs must factor data protection into API design from the start — not after the fact.

Consider the principle of least privilege, which requires ensuring users have only the minimum permissions necessary to perform their jobs.

GDPR Article 25 is *rooted* in least privilege, requiring companies to implement "technical and organizational measures for ensuring that, by default, only personal data which are necessary for each specific purpose ... are processed." In turn, API developers should implement user authentication and authorization controls to safeguard the sensitive data flowing through their APIs. API development teams must also ensure that data stays confidential in transit by using secure communication protocols to encrypt the exchange of information between client and server.

However, what about the existing ecosystem of APIs that organizations have built over the past years or even decades? A substantial portion of enterprise APIs are unmanaged, forgotten about, or running perpetually without checks and balances. In these cases, GDPR compliance requires:

- Discovering every API in your IT environment
- Assessing their risk factors (e.g., the types of data they've been exchanging and who or what can access that data)
- Remediating any vulnerabilities such as misconfigurations or weak authentication mechanisms
- Continuously testing APIs for resilience against both traditional and emerging breach and attack methods

3. Digital Operational Resiliency Act (DORA)

Given the EU financial sector's role as a critical infrastructure operator, DORA's requirements are meant to help organizations in EU member states withstand and recover from cyberattacks. With DORA, the sector will have a binding, comprehensive risk management framework for information and communications technology (ICT). The act aims to harmonize and tighten requirements for EU financial firms, as the current landscape entails myriad regulations and standards.

In total, more than 22,000 financial institutions and IT service providers in the EU are affected by DORA. Of note, this includes third parties that provide EU financial firms with ICT systems and services, including cloud service providers. The act calls for financial institutions to develop ICT third-party risk strategies and conduct due diligence to vet providers' suitability.

DORA sets out several requirements with API security implications, including digital operational stability, which requires organizations to implement regular testing programs that identify potential gaps, vulnerabilities, and/or deficiencies in digital operational stability. Think network security tests, penetration tests, web app tests, and more. It's important to conduct mandatory reviews based on threat-led penetration testing (TLPT), depending on the size, risk, and business profile of the financial enterprise. Equally important is regularly testing your APIs for vulnerabilities.

DORA outlines examples of security testing that include web-based application and API tests. This includes utilizing public-facing resources such as the Open Worldwide Application Security Project (OWASP). The OWASP Top 10 API Security Risks, in particular, help organizations identify configuration errors, weaknesses, logic flaws, and code issues that allow attackers to gain access, manipulate, or otherwise control organizational resources.

4. Health Insurance and Portability and Accountability Act (HIPAA)

HIPAA focuses on data privacy and security rules to safeguard protected health information (PHI) in electronic health records (EHRs), computerized physician order entry platforms, and other healthcare IT systems. Any U.S. healthcare provider, plan administrator, or clearinghouse that electronically stores or transmits PHI must comply with HIPAA. This involves ensuring the confidentiality, integrity, and availability of PHI and protecting it against unauthorized disclosure and improper use.

HIPAA is an example of a regulation that has significant implications for APIs, even if it doesn't explicitly call out APIs in its requirements.

Consider a technology vendor that builds patient portals for 24/7 healthcare clinics. An underlying function of these portals is the ability to give patients efficient and secure access to data on their doctor visits, test results, payments, and more. APIs are the facilitators of that exchange. Both the clinic and the vendor are bound to adhere to HIPAA requirements.

HIPAA's Privacy Rule specifies covered entities "must develop and implement policies and procedures that restrict access and uses of protected health information based on the specific roles of the members of their workforce." Therefore, an organization's API developers must embed technical safeguards such as authentication, unique user IDs, and role-based access controls to ensure least privilege is in place.

Visibility is also essential for HIPAA-covered organizations, be it a provider whose IT team creates bespoke APIs or a vendor developing APIs for the provider. Organizations need real-time assessment and reporting on each API's risk posture, including the types of PHI they transmit. This is relevant for compliance and for fulfilling HIPAA's requirement to respond to individuals requesting information on when, where, why, and to whom their PHI has been disclosed.

5. Network and Information Security Directive (NIS2)

The EU adopted version 2.0 of the NIS directive in January 2023, building upon the original version's guidelines for securing IT infrastructure and reporting incidents. While v2.0 does not specifically mention APIs, its requirements have significant implications for the protection and management of APIs as they are integral to the functioning of many digital services in organizations subject to the directive. Of note, NIS2 includes:

- A broader range of sectors — for example, cloud service providers and social media companies join the existing list, which includes critical infrastructure operators. For these sectors, where APIs are extensively used for integration and service delivery, ensuring API security becomes a priority.
- A new emphasis on securing supply chains — enterprises must assess risk and secure their IT supply chains and third-party supplier relationships. Because APIs are often used to integrate external services, ensuring their security is key to compliance.
- A requirement to build an information security management system that assesses people, policies, and technology to protect sensitive resources and ensure operational resiliency. Since APIs are fast-growing attack vectors, they must be included in risk management strategies.
- Reporting of significant cybersecurity incidents, including API breaches. Therefore, organizations need to put mechanisms in place to monitor, detect, and report API-related incidents.

6. Guidance for U.S. financial services regulators

The Federal Financial Institutions Examination Council (FFIEC) creates the guidance and standards for federal regulators to oversee the U.S. financial industry. This includes the Federal Reserve, FDIC, OCC, and NCUA. The council's mission is to protect consumers and investors from fraud, abuse, and misconduct. While not a regulation, the FFIEC's guidance is key to ensuring financial firms know how to align with its recommended security measures.

This is a key example of a document that includes specific guidance on how to secure APIs and, in turn, protect consumers from fraud and identity theft. Here's an overview:

- **Inventory:** The FFIEC recommends building an inventory of all information systems — including APIs — requiring authentication and access controls. This applies not only to financial institutions but also to their third parties, such as cloud service providers.
- **Authentication:** The API should only allow access to authorized users. It's critical to identify all users (e.g., customers) for which access controls are needed. Also important is identifying users who warrant enhanced controls, such as multi-factor authentication.
- **Authorization:** The API should only allow access to specific resources for authorized users. With that said, the FFIEC recommends implementing layered security — for example, monitoring, logging, and reporting activities to identify and track unauthorized access.
- **Risk management:** There are a number of effective risk management practices the FFIEC identifies in their latest guidance. However, they explicitly call out APIs under the Inventory of Information Systems category, which means you need an accurate inventory of your APIs.

An organization may be up to speed on well-known threats such as phishing or ransomware, but the FFIEC calls for identifying *any* cyberthreat with "reasonable probability of impacting financial institution information systems" and their data. As cited in the introduction, 78% of organizations have faced API security incidents, so you can count on API protection being a compliance imperative as financial regulators' requirements continue evolving.

Meet compliance challenges with best-practices API protection

Today's threat landscape calls for a complete API security solution that provides API discovery, posture management, runtime protection, and API security testing. This comprehensive approach works as a complement to any WAF or API gateway that is already in place.

1. API discovery

It's not uncommon to have APIs that no one knows about. Most organizations have little to no visibility into a large percentage of their API traffic, often because they assume all of their APIs are routed through an API gateway. But that is not the case. Your enterprise is exposed to a range of risks without a complete and accurate inventory. Core capabilities needed:

- Locating and inventorying all of your APIs, regardless of configuration or type
- Detecting dormant, legacy, and zombie APIs
- Identifying forgotten, neglected, or otherwise unknown shadow domains
- Eliminating blind spots and uncovering potential attack paths

2. API posture management

With a complete API inventory in place, it's critical to understand what types of data flow through your APIs and how that affects your ability to comply with regulatory requirements. API posture management provides a comprehensive view of traffic, code, and configurations to assess your organization's API security posture. Core capabilities needed:

- Automatically scanning infrastructure to uncover misconfigurations and hidden risks
- Creating custom workflows to notify key stakeholders of vulnerabilities
- Identifying which APIs and internal users are able to access sensitive data
- Assigning severity rankings to detected issues to prioritize remediation

3. API runtime security

You're no doubt familiar with the concept of "assume a breach." API-specific breaches and attacks are reaching that same degree of inevitability. For all of your APIs that are live in production, you need to be able to detect and block attacks in real time. Core capabilities needed:

- Monitoring for data tampering and leakage, policy violations, suspicious behavior, and API attacks
- Analyzing API traffic without additional network changes or difficult-to-install agents
- Integrating with existing workflows (ticketing, SIEMs, etc.) to alert security/operations teams
- Preventing attacks and misuse in real time with partially or fully automated remediation

4. API security testing

API development teams are under pressure to work as quickly as possible. Speed is essential for every application developed, making it easier for a vulnerability or design flaw to happen and subsequently go undetected. Testing APIs in development before they are released into production greatly reduces both risk and the cost of fixing an API that is vulnerable. Core capabilities needed:

- Running a wide range of automated tests that simulate malicious traffic
- Discovering vulnerabilities before APIs enter production, reducing the risk of successful attacks
- Inspecting your API specifications against established governance policies and rules
- Running API-focused security tests that run on demand or as part of a CI/CD pipeline

How Akamai API Security can streamline API compliance complexities

APIs are a leading cause of the breaches that today's regulations are designed to prevent. What does it take to secure your enterprise as APIs — and their risks — multiply? The existing tools many organizations use for baseline API protection provide some protection, but it's not nearly enough. If you're seeking a better way to secure your organization's APIs and demonstrate compliance, we'd like to help you.

For every requirement and guidance covered in this white paper, Akamai API Security strengthens the protection that enterprises need — not only to comply with regulations but also to secure your customers' data and trust.

Akamai's comprehensive solution protects APIs in their initial stages of development all the way to post-production, giving you the ability to adhere to core best practices:

- API discovery
- Posture management
- Runtime protection
- Security testing

Learn more about APIs and how to protect them from attacks.

Learn how Akamai API Security can help your organization.

Akamai Security protects the applications that drive your business at every point of interaction, without compromising performance or customer experience. By leveraging the scale of our global platform and its visibility to threats, we partner with you to prevent, detect, and mitigate threats, so you can build brand trust and deliver on your vision. Learn more about Akamai's cloud computing, security, and content delivery solutions at akamai.com and akamai.com/blog, or follow Akamai Technologies on X, formerly known as Twitter, and LinkedIn. Published 09/24.