API SECURITY

A DEFINITIVE REFERENCE FOR BUILDING SECURE APIS AT SCALE

SECURE BY DESIGN
Build security in from the start

STRONGER PROTECTIONS
Authentication, authorization, and cryptography

THREAT-FOCUSED
OWASP API Security Top 10 and threat modeling

PRACTICAL & HANDS-ON
Real-world examples in 8 programming languages

GOVERNANCE & COMPLIANCE
Standards, compliance, and operational excellence

OWASP Top 10 | NIST Cybersecurity Framework | IETF Best Current Practices | ISO Standards | ZERO TRUST Zero Trust by Design | AI & MCP Frontiers

PATRICK HOCHHEIMER

API Security

A Definitive Reference for Building Secure APIs at Scale

Steve T. Publications

This book is available at https://leanpub.com/apisecurity

This version was published on 2026-07-17

Contents

A Definitive Reference for Building Secure APIs at Scale

APIs are the bloodstream of modern software. Every microservice call, every third-party integration, every mobile app request, and every AI agent invocation flows through an API. They are also the fastest-growing attack surface in enterprise IT. This book provides a comprehensive, authoritative guide to securing APIs from design through deployment to production operation. It covers authentication and authorization frameworks, cryptographic protections, threat modeling, the OWASP API Security Top 10, secure coding practices, infrastructure security, testing methodologies, governance, compliance, and emerging frontiers including zero-trust architectures, AI-enabled APIs, and the Model Context Protocol. Every chapter is grounded in established standards from OWASP, NIST, IETF, and ISO, with practical code examples in Java, Go, Python, JavaScript/TypeScript, C#, Rust, and Kotlin. Whether you are a developer building your first REST endpoint or a security architect designing an enterprise API platform, this book gives you the knowledge to build APIs that are secure by design and resilient under attack.

Introduction: The API Security Imperative

In late 2019, the Open Worldwide Application Security Project released the first edition of the OWASP API Security Top 10, a document that would fundamentally reshape how the industry understands API vulnerabilities. The project, co-founded by security researchers Erez Yalon and Inon Shkedy, was launched at the Global AppSec conference in Tel Aviv after years of observing a pattern: APIs were failing in ways that the traditional OWASP Top 10 for web applications did not capture. At the top of that inaugural list sat Broken Object Level Authorization, or BOLA, a vulnerability class that would hold the number-one position through the 2023 edition and appear in roughly 40 percent of all API attacks observed in production [1, 2].

BOLA is not a zero-day exploit or a sophisticated buffer overflow. It is a simple, almost embarrassing oversight: the API accepts an object ID from the client but never verifies that the authenticated user actually owns that object. An attacker needs only to change a number in a URL (`/api/v1/users/12345` becomes `/api/v1/users/12346`) and observe whether the server returns data belonging to someone else. The vulnerability is structurally easy to introduce because it stems from a design assumption rather than a coding error. Developers build endpoints that serve resources by ID, test them with their own accounts, assume the authentication check at the top of the request pipeline is sufficient, and ship. It is only when an attacker looks at the same endpoint from a different angle that the gap becomes visible.

The 2019 OWASP API Security Top 10 was not born in a vacuum. Industry researchers had been documenting similar vulnerabilities for years under the older name Insecure Direct Object Reference, or IDOR, a category that appeared in the traditional OWASP Web Application Top 10. But APIs exposed these flaws at a scale and with a frequency that web applications did not. The stateless nature of REST, the direct mapping of resources to URL paths, and the absence of server-side session tracking meant that every single request carried its own authorization burden. When OWASP's API Security project formalized BOLA as a distinct category, it gave the community a shared vocabulary for a

problem that was already costing organizations millions in breaches.

This story illustrates a fundamental truth about API security. The most dangerous vulnerabilities are not the ones that require deep expertise to exploit. They are the ones that arise from architectural decisions, design patterns, and implementation choices that seem reasonable until an attacker looks at them from a different angle. APIs are inherently permissive by design. Their entire purpose is to enable communication between systems, and that openness creates a tension with security that must be managed at every layer of the stack.

Why APIs Are Different

Traditional web application security focused on protecting a perimeter: a browser sending requests to a server behind a firewall. The attack surface was relatively well-defined, and security tools like web application firewalls could inspect HTTP traffic for known attack patterns. APIs change this model fundamentally. They are consumed by machines, not humans, and they operate across trust boundaries that are far more porous than a corporate network perimeter. An API endpoint might be called by a mobile app in a user's pocket, a microservice in another cloud region, a third-party partner system, or an AI agent executing on behalf of an end user. Each consumer has different security requirements, different threat profiles, and different levels of trust.

The scale of the API ecosystem underscores the urgency of this problem. According to Postman's 2024 State of the API Report, 74 percent of organizations now adopt an API-first approach to development, up from 63 percent the previous year. Sixty-two percent of companies generate revenue directly from their APIs, and for 21 percent of organizations, APIs account for more than 75 percent of total revenue. At the same time, API-driven traffic on Postman's platform surged by 73 percent year-over-year, with AI-related API calls leading the growth. The IBM Cost of a Data Breach Report for 2024 recorded a global average breach cost of $4.88 million, a 10 percent increase from the $4.45 million average in 2023 and the largest year-over-year jump since the pandemic [10]. Breaches involving cloud environments, which are inherently API-driven, carry even higher costs. APIs are consistently among the most common attack vectors in these breaches.

What Makes API Security Hard

API security is hard for several interconnected reasons. First, APIs are stateless by design. Unlike a traditional web session that maintains server-side state, RESTful APIs expect each request to carry all the information needed to authorize and process it. This shifts the burden of security context onto every individual request and makes session management, token validation, and access control more complex.

Second, APIs expose business logic directly. A web application might hide its data model behind HTML forms and client-side validation, but an API is the data model. Every endpoint maps to a resource or operation, and the structure of those endpoints reveals the underlying architecture to anyone who can inspect them. Attackers do not need to guess what functionality exists; they can often discover it through documentation, schema introspection, or systematic enumeration.

Third, APIs operate at machine speed. A human attacker clicking through a web application generates perhaps ten requests per minute. An automated tool can send thousands of API requests per second, amplifying the impact of any vulnerability that allows resource exhaustion, data scraping, or credential stuffing. The economics of API attacks are fundamentally different from traditional web attacks: the cost to attack is low, the scale is enormous, and the potential payoff is high.

Fourth, the API ecosystem is fragmented. A single enterprise may expose hundreds or thousands of API endpoints across multiple protocols (REST, GraphQL, gRPC, SOAP), multiple clouds, and multiple development teams. Maintaining visibility into this landscape, enforcing consistent security policies, and ensuring that every endpoint meets minimum security standards is a significant operational challenge. The OWASP API Security Top 10 specifically identifies Improper Inventory Management as a critical risk category, reflecting the real-world prevalence of undocumented "shadow" APIs and abandoned "zombie" APIs that remain accessible but no longer receive security updates.

How This Book Is Organized

This book is structured to take you from foundational concepts through advanced topics in a logical progression. Each chapter builds on the material

that precedes it, but every chapter is also designed to stand alone as a reference you can return to when facing a specific challenge.

The first part of the book (Chapters 1-2) establishes context and vocabulary. You will learn about the API economy, why APIs present unique security challenges, and the technical foundations of different API styles and protocols including REST, GraphQL, gRPC, SOAP, WebSockets, and event-driven architectures.

The core security mechanisms are covered in Chapters 3-4. Authentication and authorization form the bedrock of API security, and you will find detailed treatment of OAuth 2.1, OpenID Connect, JWTs, mutual TLS, API keys, and fine-grained access control models. Cryptography for APIs covers TLS configuration, encryption algorithms, key management, and common cryptographic failures.

Chapters 5-7 focus on identifying and preventing vulnerabilities. Threat modeling methodologies adapted for API contexts help you anticipate attacks before writing code. The OWASP API Security Top 10 receives dedicated treatment with real-world examples, exploitation techniques, and mitigation strategies for each category. Secure API design principles and coding practices translate these concepts into concrete implementation patterns across multiple programming languages.

Infrastructure security is the domain of Chapter 8, which covers API gateways, service meshes, microservices security patterns, cloud-native architectures, and Kubernetes hardening. Chapter 9 addresses the security of the development pipeline itself, including CI/CD security, software supply chain protection, secrets management, and container image security.

Chapter 10 covers testing and monitoring: how to find vulnerabilities through manual testing, static analysis, dynamic analysis, interactive analysis, fuzzing, and penetration testing. It also covers logging, monitoring, observability, and incident response procedures specific to API environments.

Governance, compliance, and lifecycle management are addressed in Chapter 11, covering regulatory requirements (GDPR, CCPA, HIPAA, PCI-DSS), API inventory and discovery, versioning strategies, and third-party API security.

The final chapter, Chapter 12, looks at the frontier of API security. Zero-trust architectures provide a framework for rethinking trust in distributed systems. AI-enabled APIs and LLM integrations introduce entirely new attack surfaces including prompt injection, data leakage, and tool poisoning through

protocols like the Model Context Protocol. Emerging threats and trends round out the picture of where API security is heading.

Who This Book Is For

This book assumes familiarity with software development fundamentals, networking concepts (HTTP, DNS, TLS), and basic security principles. It is written for:

- **Developers** who build APIs and need to understand how to implement security controls correctly in their code
- **Security engineers** who assess, test, and harden API security across an organization
- **Architects** who design API platforms, microservices ecosystems, and integration patterns
- **DevSecOps practitioners** who automate security testing and enforce policies in CI/CD pipelines
- **Technical leaders** (engineering managers, CTOs, CISOs) who set security strategy and allocate resources

The code examples span Java, Go, Python, JavaScript/TypeScript, C#, Rust, and Kotlin. If you work primarily in one language, you can still benefit from the patterns and concepts presented in other languages, as the security principles are universal even if the syntax differs.

How to Use This Book

Read this book cover to cover for a comprehensive foundation. Alternatively, use it as a reference: jump to the chapter that addresses your immediate need and return to earlier chapters for background on concepts you encounter. The cross-references throughout the text will help you navigate between related topics.

Every code example is designed to be production-oriented, not academic. Configuration files are complete and usable. Architectural diagrams use ASCII art so they render correctly in any environment. Case studies are drawn from

real-world incidents and publicly disclosed vulnerabilities. Where multiple approaches exist, the trade-offs are laid out explicitly so you can make informed decisions for your context.

The landscape of API security evolves rapidly. New vulnerabilities are discovered, new standards are published, and new attack techniques emerge. This book is grounded in established standards and timeless principles that will remain relevant regardless of the specific tools or frameworks you use. But you should supplement it with ongoing attention to OWASP publications, NIST special publications, IETF RFCs, and the security advisories of the technologies you depend on.

Secure APIs are not an accident. They are the product of deliberate design decisions, rigorous implementation practices, continuous testing, and vigilant operation. This book gives you the knowledge to make those decisions with confidence.

Chapter 1: The API Landscape and Why Security Matters

The API Economy and Its Scale

The modern software landscape is built on APIs. Every time a mobile app fetches weather data, an e-commerce site processes a payment, a healthcare system exchanges patient records, or an AI agent queries a database to answer a question, an API mediates the interaction. What began as a technical mechanism for enabling communication between software components has evolved into the foundational architecture of the digital economy.

The numbers tell the story of this transformation. According to Postman's 2024 State of the API Report, which surveyed over 5,600 developers and API professionals worldwide, 74 percent of organizations now adopt an API-first approach to development, up from 66 percent in 2023 and just 19 percent in 2018 [11]. Sixty-two percent of respondents reported working with APIs that generate revenue, and for 21 percent of organizations, APIs account for more than 75 percent of total company revenue [11]. The average organization now manages more than 40 APIs, with large enterprises often maintaining hundreds or thousands across their ecosystems.

API traffic itself has exploded. Postman observed a 73 percent increase in AI-driven API traffic on its platform over a single year [11]. Cloud providers report that API gateway requests number in the trillions per month globally. The API economy, measured by the value of transactions and integrations mediated through APIs, is estimated at hundreds of billions of dollars annually and growing.

This growth is not limited to technology companies. Financial services, healthcare, government, retail, manufacturing, and every other sector has undergone API-driven transformation. Banks expose payment APIs for open banking compliance under regulations like PSD2 in Europe. Healthcare organizations use HL7 FHIR APIs to exchange clinical data. Government agencies

publish public data through RESTful endpoints. Retailers integrate with logistics providers, payment processors, and marketplace platforms through APIs.

The API economy creates enormous value, but it also creates enormous risk. Every API endpoint is a potential entry point for attackers. Every integration adds a dependency that could be compromised. Every third-party API introduces an external trust relationship that must be managed. The scale of the API landscape means that even well-intentioned security programs can struggle to maintain visibility and control.

From Monoliths to API-First Architectures

To understand why APIs present unique security challenges, it helps to understand how software architecture has evolved. Two decades ago, the dominant model was the monolithic application: a single codebase deployed as a single unit, typically behind a corporate firewall, accessed through a web browser. Security focused on protecting the perimeter: firewalls, intrusion detection systems, and web application firewalls filtered incoming traffic before it reached the application. The attack surface was relatively well-defined and manageable.

The shift to distributed architectures began with service-oriented architecture (SOA) in the mid-2000s, which decomposed monolithic applications into discrete services communicating through message buses and enterprise service buses. SOA introduced APIs as first-class citizens but kept them largely internal, behind the corporate network perimeter.

Microservices architecture, popularized by companies like Netflix and Amazon around 2014, took decomposition further. Each microservice is independently deployable, owns its own data store, and communicates with other services through lightweight APIs, typically REST over HTTP or gRPC over HTTP/2. The API became the primary interface for every component in the system, not just the external-facing layer.

The API-first philosophy went one step further: instead of building an application and then exposing an API on top of it, organizations began designing the API first and then implementing the application to serve that API. This approach, championed by companies like Stripe, Twilio, and GitHub, treats the API as the product rather than a byproduct of the product. The API contract

drives development, documentation is generated from the specification, and client libraries are published for multiple languages.

This architectural evolution has profound security implications. In the monolithic era, the perimeter was the primary defense. Inside the perimeter, services trusted each other implicitly because they shared the same network and often the same codebase. In the API-first world, trust must be explicit at every boundary. Every service-to-service call crosses a trust boundary that must be authenticated and authorized. The network perimeter has dissolved into a mesh of connections spanning multiple clouds, edge locations, and third-party environments.

Why APIs Are Different Attack Surfaces

APIs differ from traditional web applications in ways that fundamentally change the security landscape. Understanding these differences is essential for designing effective defenses.

Statelessness. RESTful APIs are stateless by design, meaning each request must contain all the information needed to process it. The server does not maintain session state between requests. This contrasts with traditional web applications that use server-side sessions to track authenticated users across multiple page views. Statelessness shifts the burden of security context onto every individual request. Authentication tokens, authorization claims, and security metadata must be carried with every API call, and the server must validate them independently each time. This creates opportunities for token theft, replay attacks, and validation bypasses that do not exist in stateful architectures.

Machine-to-machine communication. APIs are consumed by machines, not humans. A browser user clicking through a web application generates perhaps ten to twenty requests per minute. An automated API client can generate thousands of requests per second. This difference in scale changes the economics of attacks. Brute force attacks, credential stuffing, data scraping, and denial-of-service attacks are all amplified by the machine-speed nature of API consumption. Rate limiting becomes not just a performance concern but a security requirement.

Direct exposure of business logic. A web application might hide its data model behind HTML forms, client-side validation, and presentation-layer logic.

An API is the data model, exposed directly. Every endpoint maps to a resource or operation, and the structure of those endpoints reveals the underlying architecture. GraphQL APIs make this even more explicit by publishing their schema through introspection. Attackers do not need to guess what functionality exists; they can discover it through documentation, automated scanning, or systematic enumeration.

Multiple consumers with different trust levels. A single API endpoint may be called by a mobile app in a user's pocket, an internal microservice, a third-party partner system, and a public integration platform. Each consumer operates with a different level of trust, different security requirements, and different threat profiles. The API must enforce access controls that are fine-grained enough to distinguish between these consumers while remaining performant at scale.

Cross-origin complexity. APIs are frequently consumed from different origins than the server that hosts them. Cross-Origin Resource Sharing (CORS) policies, browser security models, and the need for cross-domain authentication create additional attack vectors including CSRF, credential leakage through referer headers, and misconfigured CORS policies that grant access to unauthorized origins.

Data density. A single API response can contain a large volume of structured data. Where a web page might display a few fields from a user profile, an API endpoint might return the entire user object with all nested relationships. This creates opportunities for excessive data exposure, where the API returns more information than the consumer needs or is authorized to see. GraphQL APIs amplify this risk by allowing clients to request exactly which fields they want, potentially exposing sensitive fields that the developer forgot to restrict at the resolver level.

The Cost of API Breaches

The financial impact of API security failures is substantial and growing. IBM's annual Cost of a Data Breach Report provides the most comprehensive analysis of breach costs across industries. The 2024 edition recorded a global average breach cost of $4.88 million, a 10 percent increase from the $4.45 million average in 2023 and the largest year-over-year jump since the pandemic. Breaches involving cloud environments, which are inherently API-driven, carry even higher costs.

API-specific breaches often involve large-scale data exfiltration. Three cases illustrate how API failures cascade into catastrophic incidents.

Case Study: Target Data Breach (2013). In November 2013, attackers compromised Target's network by stealing credentials belonging to Fazio Mechanical Services, a Pennsylvania-based HVAC contractor that had legitimate remote access to Target's vendor portal for billing and project management purposes [3]. The attackers used a spear-phishing campaign to harvest the HVAC vendor's login credentials, then leveraged those credentials to enter Target's network. From there, they moved laterally to payment card processing systems and deployed memory-scraping malware known as BlackPOS across thousands of point-of-sale terminals. The breach ultimately exposed approximately 40 million payment card numbers and 70 million records of personal customer data. The root cause was not a vulnerability in Target's own code but a failure in third-party access management: the HVAC vendor's credentials provided network access far broader than the narrow scope required for invoice submission. The incident cost Target approximately $292 million in legal settlements, fines, and remediation, and led to the resignation of the CEO and CIO. For API security, the lesson is clear: third-party integrations must be treated as attack surfaces. Vendor credentials should be scoped to the minimum necessary permissions, monitored for anomalous access patterns, and rotated on a regular schedule. This case is revisited in Chapter 11 in the context of third-party API security and vendor risk management.

Case Study: Capital One Data Breach (2019). In February 2019, a former AWS employee named Paige Thompson exploited a misconfigured web application firewall running on Capital One's AWS infrastructure to access approximately 100 million customer records, including credit applications, Social Security numbers, and bank account information [3]. The technical attack chain began with an SSRF vulnerability in Capital One's ModSecurity WAF instance. The WAF was deployed on an EC2 instance that had been granted an IAM role with overly permissive S3 bucket access. Thompson sent a specially crafted HTTP request header containing a Python code injection payload, which the misconfigured WAF executed. The code exploited the SSRF vector to query the AWS instance metadata service at `169.254.169.254`, retrieving IAM credentials that allowed access to Capital One's S3 storage buckets containing MongoDB database snapshots. The breach was not detected in real time because the attack traffic blended with legitimate AWS API calls from authorized services. Capital One's estimated costs exceeded $150 million, including a $80 million settlement with the CFPB, customer notification

expenses, and security infrastructure improvements. The architectural lesson is that a single misconfiguration at the infrastructure layer (in this case, an IAM role with wildcard S3 permissions attached to a WAF instance) can bypass every application-level security control. This case is examined in Chapter 6 under SSRF (API7) and in Chapter 8 under cloud-native security misconfigurations.

Case Study: Uber Data Breach (2016 and 2022). Uber experienced two distinct API-related breaches that illustrate different failure modes. In 2016, attackers obtained an AWS access key from a former Uber engineer and used it to download a file from an S3 bucket containing personal information for approximately 57 million driver and rider accounts, including names, email addresses, phone numbers, and in some cases driver's license numbers and bank account routing numbers [4]. Uber attempted to cover up the breach by paying the attackers $100,000 through a consulting contract with a cybersecurity firm, a decision that was later disclosed and resulted in additional regulatory scrutiny. The 2022 breach, orchestrated by the Lapsus$ extortion group, followed a different attack path: the attacker compromised an Uber contractor's credentials purchased on the dark web, used those credentials to access internal systems, approved an MFA prompt through social engineering, and gained access to Uber's Privileged Access Management platform [4]. The attacker then exfiltrated source code, vulnerability reports, IT asset management data, and Slack messages. While Uber stated that customer payment data was not accessed in the 2022 incident, the breach exposed internal security documentation and demonstrated how credential compromise combined with insufficient PAM controls can lead to comprehensive internal access. Both incidents are revisited in Chapter 6 under Broken Authentication (API2) and in Chapter 9 under secrets management failures.

Beyond direct financial losses, API breaches carry regulatory consequences. Under the EU General Data Protection Regulation (GDPR), fines can reach up to 4 percent of annual global turnover or EUR 20 million, whichever is greater. In 2024 alone, GDPR enforcement actions generated approximately EUR 1.2 billion in fines across Europe, bringing cumulative penalties to EUR 5.88 billion since the regulation took effect in May 2018 [44]. The California Consumer Privacy Act (CCPA), Health Insurance Portability and Accountability Act (HIPAA), and Payment Card Industry Data Security Standard (PCI-DSS) all impose requirements that directly impact API security, including data protection, access controls, audit logging, and breach notification.

The reputational damage of an API breach extends beyond fines and re-

mediation costs. Customers lose trust when they learn that their data was exposed through a security failure. Business partners reassess integration agreements. Insurance premiums increase. The competitive advantage of being the most connected, API-rich platform in your industry can evaporate overnight if security is perceived as weak.

How This Book Is Structured

This book takes a systematic approach to API security, organized into four parts that build on each other logically.

Part I: Foundations (Chapters 1-2) establishes the context and vocabulary you need. Chapter 1, which you are reading now, covers the API landscape and why security matters. Chapter 2 dives into the technical foundations of different API styles and protocols, examining the security properties of REST, GraphQL, gRPC, SOAP, WebSockets, and event-driven architectures. Understanding how each protocol works is prerequisite knowledge for securing it.

Part II: Core Security Mechanisms (Chapters 3-4) covers the mechanisms that verify identity and protect data. Chapter 3 provides a comprehensive treatment of authentication and authorization, including OAuth 2.1, OpenID Connect, JWTs, mutual TLS, API keys, token management, and fine-grained access control models. Chapter 4 covers the cryptographic primitives that protect API data in transit and at rest, including TLS configuration, encryption algorithms, key management, and common cryptographic failures.

Part III: Vulnerability Prevention and Infrastructure (Chapters 5-9) focuses on identifying and preventing attacks. Chapter 5 teaches systematic approaches to threat modeling for APIs. Chapter 6 provides a comprehensive deep dive into the OWASP API Security Top 10, with real-world examples, exploitation techniques, and mitigation strategies for each category. Chapter 7 translates security principles into concrete design decisions and code patterns across multiple programming languages. Chapter 8 covers the infrastructure components that secure APIs at scale: API gateways, service meshes, microservices patterns, and cloud-native security. Chapter 9 addresses the security of the development pipeline itself, including CI/CD security, software supply chain protection, and secrets management.

Part IV: Operations, Governance, and the Future (Chapters 10-12) covers the ongoing disciplines of API security. Chapter 10 addresses testing methodologies, monitoring, observability, and incident response. Chapter 11 covers

governance frameworks, regulatory compliance, API lifecycle management, and third-party API security. Chapter 12 looks at the frontier: zero-trust architectures, AI-enabled APIs, LLM integrations, the Model Context Protocol, and emerging threats.

Each chapter includes practical code examples, configuration files, architectural diagrams, case studies drawn from real-world incidents, and implementation checklists. The book is designed to be read sequentially for a comprehensive foundation, but every chapter also stands alone as a reference you can return to when facing a specific challenge. Cross-references throughout the text help you navigate between related topics.

The API security landscape evolves rapidly. New vulnerabilities are discovered, new standards are published, and new attack techniques emerge. This book is grounded in established standards and timeless principles that will remain relevant regardless of the specific tools or frameworks you use. But you should supplement it with ongoing attention to OWASP publications, NIST special publications, IETF RFCs, and the security advisories of the technologies you depend on. Secure APIs are not a one-time achievement; they are an ongoing discipline.

Chapter 2: API Fundamentals and Protocol Diversity

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

RESTful APIs: Conventions, Constraints, and Security Implications

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

GraphQL: Flexibility, Attack Surface, and Query Security

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

C# (ASP.NET Core): JWT Bearer Authentication with Policy-Based Authorization

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Rust (Axum): Type-Safe JWT Authentication with Request Extractors

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

gRPC and Protocol Buffers: Performance, Binary Serialization, and Trust

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

SOAP: Enterprise Legacy, XML Processing, and WS-Security

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

WebSockets and Real-Time APIs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Event-Driven APIs and Message Brokers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Protocol Comparison and Security Trade-offs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Chapter 3: Authentication and Authorization Foundations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

API Keys: Simplicity, Limitations, and Safe Usage

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

OAuth 2.1: Authorization Frameworks, Grant Types, and Security Best Practices

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

OpenID Connect: Identity Layer on Top of OAuth

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

JSON Web Tokens (JWT): Structure, Validation, and Common Mistakes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Mutual TLS (mTLS): Certificate-Based Authentication

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Token Management, Rotation, and Revocation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Delegated Authorization and Fine-Grained Access Control

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Comparison Matrix: Choosing the Right Mechanism

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Chapter 4: Cryptography for APIs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Transport Layer Security (TLS): Versions, Ciphers, and Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Encryption Algorithms: Symmetric vs Asymmetric for API Use Cases

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Key Management and Rotation Strategies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Hashing, Signing, and Integrity Verification

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Common Cryptographic Failures in API Implementations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Post-Quantum Considerations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Chapter 5: Threat Modeling for APIs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

What Is Threat Modeling and Why It Matters for APIs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

STRIDE Applied to API Architectures

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Data Flow Diagrams and Trust Boundaries

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Attack Trees and Scenario Development

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Automated Threat Modeling Tools

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Threat Scenario: Multi-Step BOLA Attack Chain

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Integrating Threat Modeling into the SDLC

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Chapter 6: The OWASP API Security Top 10

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

API1: Broken Object Level Authorization (BOLA)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

API2: Broken Authentication

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

API3: Broken Object Property Level Authorization (BOPLA)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

API4: Unrestricted Resource Consumption

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

API5: Broken Function Level Authorization (BFLA)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

API6: Unrestricted Access to Sensitive Business Flows

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

API7: Server Side Request Forgery (SSRF)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

API8: Security Misconfiguration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

API9: Improper Inventory Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

API10: Unsafe Consumption of APIs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Summary: The OWASP API Security Top 10 at a Glance

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Chapter 7: Secure API Design and Coding Practices

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Security by Design: Principles for API Architecture

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Input Validation and Output Encoding

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Error Handling That Doesn't Leak Information

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Business Logic Security and Anti-Automation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Rate Limiting, Throttling, and Quotas

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Secure Coding Patterns Across Languages

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Common Anti-Patterns and Implementation Mistakes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Chapter 8: API Infrastructure Security

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

API Gateways: Capabilities, Patterns, and Security Functions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Service Meshes: Sidecar Proxies, mTLS, and Policy Enforcement

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Microservices Security Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Cloud-Native API Security

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Kubernetes API Server and Pod-to-Pod Communication

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Infrastructure Case Study: The Capital One Breach Through an Infrastructure Lens

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Network Segmentation and Zero-Trust Networking

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Chapter 9: CI/CD, Supply Chain, and Secrets Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Secure CI/CD Pipelines for API Development

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Software Supply Chain Security and SBOMs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Secrets Management: Storage, Rotation, and Access

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Container Image Security

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Infrastructure as Code Security

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Dependency Management and Vulnerability Scanning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Chapter 10: Testing, Monitoring, and Incident Response

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Manual API Security Testing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Static Analysis (SAST) for API Code

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Dynamic Analysis (DAST) for Live APIs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Interactive Analysis (IAST) and Runtime Protection

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Fuzzing and Automated Vulnerability Discovery

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Penetration Testing Methodologies for APIs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Logging, Monitoring, and Observability for APIs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Incident Response for API Breaches

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Chapter 11: Governance, Compliance, and Lifecycle Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

API Governance Frameworks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Regulatory Requirements: GDPR, CCPA, HIPAA, PCI-DSS

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Privacy by Design for APIs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

API Inventory and Discovery

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

API Versioning Strategies and Security Implications

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

API Lifecycle Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Third-Party and Partner API Security

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Chapter 12: Zero Trust, AI, and Emerging Trends

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Zero-Trust Architecture for API Ecosystems

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

AI-Enabled APIs: New Attack Surfaces

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

LLM Integrations: Prompt Injection, Data Leakage, and Guardrails

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Model Context Protocol (MCP) and Agent Security

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Emerging Threats: API Abuse, Credential Stuffing at Scale

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

The Future of API Security

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

Conclusion: Building a Resilient API Security Posture

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.

References

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/apisecurity.