THALES
Building a future we can all trust
CYBERSECURITY

imperva

Imperva API Threat Report

The API Battleground

How attackers weaponize business logic — Metrics, Industry breakdowns, trends & defense priorities

> ACROSS 4,000+ MONITORED ENVIRONMENTS, WE RECORDED MORE THAN 40,000 API INCIDENTS.

Table of Contents

1. Executive Summary

In the first half of 2025 (H1 2025), Imperva observed a decisive shift: APIs are now the primary battlefield. Across **4,000+ monitored environments,** we recorded **more than 40,000 API incidents.** While APIs account for roughly **14% of all attacks,** they attract ~44% of advanced bot activity, showing attackers concentrate their smartest automation on API logic rather than noisy scans. Most alarming: we observed an application-layer DDoS spike that reached **15 million requests per second (RPS)** against a financial API — a clear demonstration that attackers now combine scale with stealth.

The most damaging attacks today are valid API calls that bend business logic — promo loops, gift-card cracking, targeted data scraping, and credential-driven account takeovers. These requests look normal to signature-based tools and static rate limits, so defenders drown in alerts for malformed traffic while the real attacks slip through the very logic that powers your business.

> “
>
> WE OBSERVED AN APPLICATION-LAYER DDoS SPIKE THAT REACHED 15 MILLION REQUESTS PER SECOND AGAINST A FINANCIAL API
>
> ”

What Defenders Must Do Differently

Add business context and runtime intelligence on top of traditional controls. That means continuous API discovery, runtime contract/schema enforcement, object-level authorization, behavior-driven bot detection tied to business KPIs (promo redemptions, refund spikes, reservation rates), and adaptive throttling. These capabilities turn “valid” requests into defensible events, not missed threats.

Key Truths From Our Telemetry

Attackers focus on where money, identity and workflows live — data-access, checkout/payment, and authentication endpoints.

Business-logic abuse (BOLA) and parameter tampering are the dominant vectors; signature-only defenses and coarse rate limits miss them.

Shadow and partner APIs create the largest operational blind spots; combined with headless browsers and botnets, attackers can extract value while blending into normal traffic.

BACK TO TABLE OF CONTENTS

Operational Model (Fast Wins)

DISCOVER >> ASSESS >> MITIGATE

Discover every endpoint (public, private, shadow); assess their business impact and risk; mitigate with runtime enforcement, adaptive bot controls, and DDoS protection. These steps yield immediate, measurable risk reduction.

Top Actions For Leaders

1. **Implement continuous API discovery and classify endpoints by business impact.**
2. **Enforce runtime schema validation and object-level authorization for high-risk APIs.**
3. **Deploy context-aware bot mitigation and adaptive throttling for checkout/auth/data endpoints.**
4. **Operationalize API ownership (product + security), publish a short set of API KPIs to the board, and run API-specific tabletop exercises.**
5. **API-aware telemetry and business-context detection to stop the attacks that look "normal" but are anything but.**

Why Act Now

Whether you're defending a global bank, a healthcare network, or an AI startup, APIs now determine risk to revenue, trust, and compliance. This report gives you the "why," the "what," and the practical "how" to secure your APIs before the next large-scale campaign becomes your headline.

Bottom Line

Traditional controls are necessary—but not sufficient. Make discovery, runtime intelligence, and behavior-driven defenses the core of your API security posture.

BACK TO TABLE OF CONTENTS

2. About the Data & Methodology

In just the first half of 2025, our Threat Research team at Imperva collected and examined real-world API attack data from thousands of customer environments around the globe. We wanted to understand exactly how bad actors are probing, breaking, and bending API logic—so you can see the full picture and defend against it.

Here's What We Looked At:

40,000+ API incidents

in the **first six months of 2025**, from small probes to full-scale breaches

Bot telemetry & fingerprinting

tracking both web-based and mobile-app bots **to see how they hide and behave**

CVE exploit tracking

focused on known troublemakers like Log4j, Oracle WebLogic, and Joomla vulnerabilities

DDoS forensics

spotlighting attacks that overwhelmed APIs**—like the 15 million-RPS application-layer flood we stopped on a major financial endpoint**

Endpoint behavior analysis

including request volumes, spikes, and weird patterns **that signal abuse**

BACK TO TABLE OF CONTENTS

Our Approach

1. **Categorize techniques** (scraping, account takeover, fraud) and map them to the affected endpoints (authentication, checkout, data-access).

2. **Correlate behaviors** by flagging outliers—whether it's a sudden burst of traffic or a slow, steady trickle that drips under standard alerts.

3. **Segment by industry,** so you know which sectors are under the heaviest fire.

4. **Synthesize business logic insights** —we didn't just count attacks; we dug into *how* and *why* they worked, so your team can plug the gaps before attackers exploit them.

Limitations

Dataset reflects Imperva customer footprint and controlled labs; absolute global volumes may differ, but behavioral trends and relative distributions are robust.

3. The API Threat Landscape

3.1 APIs Are the Primary Attack Surface

APIs expose business logic — not just data. They execute account changes, payments, promotions, and access controls. In our analysis, attackers treat APIs as direct money channels: exploiting workflows yields immediate financial or identity returns. Whereas web UI attacks often require human interaction, API attacks scale automatically: one script can enumerate millions of resources or drive thousands of promo redemptions per minute.

Every Day, Your Teams Use APIs To:

Automate workflows (think instant data syncs and real-time updates)

Connect with partners (third-party apps, payment gateways, analytics tools)

**Deliver smooth user experiences
(no more "click refresh"— everything happens behind the scenes)**

Expose core business logic (pricing rules, credit checks, order approvals)

But here's the problem:

Those same APIs can be turned against you. Attackers send perfectly valid requests that look harmless but:

Drain your promos by looping discount codes

Steal sensitive data through hidden "shadow" endpoints

Hijack user accounts with stolen credentials

Overwhelm checkout flows to steal money or hoard inventory

Traditional tools can't spot these subtle abuses, leaving you drowning in alerts for malformed traffic while the real attacks slip through the very business logic that makes APIs so powerful—and so vulnerable.

BACK TO TABLE OF CONTENTS

3.2 API Attack Volume & Growth Patterns

We base these findings on Imperva's global telemetry—drawing from WAF logs, API gateway records, bot management sensors, and DDoS mitigation feeds across 4,000+ customer environments. Our Threat Research team combines behavioral analytics and ML-driven fingerprinting to separate "smart" bots from human and benign traffic, then correlates that with real incident data. This rigorous approach gives us high confidence in the trends below backed by both our internal data and industry reports showing similar shifts.

Key Observations

A Historic Spike in API Attacks

- **44% of advanced bot traffic now attacks APIs,** up from under 30% just two years ago. In contrast, only 10% of these smart bots target traditional web pages.

Why This Matters

APIs power critical workflows—authentication, payments, data queries—making them more valuable to attackers than static web pages.

Volume of Attacks Is Skyrocketing

- **40,000+ API incidents recorded in the first six months of 2025**—an average of 220 incidents per day.
- Extrapolating this rate suggests the **second half of 2025 could see 80,000+ incidents** if defenses don't adapt.

DDoS Goes Application-Layer

- We observed **a record 15 million RPS application-layer DDoS against a financial API** — far exceeding the scale normally seen in web-app DDoS campaigns and demonstrating attackers' growing ability to weaponize API traffic.
- **68% of all API-focused DDoS traffic hit financial services**, highlighting the sector's heavy reliance on APIs for real-time transactions.

Consistent Industry Patterns

- **Financial services lead, followed by e-commerce and healthcare—**sectors where APIs handle money, personal data, and sensitive records.
- Regional breakdown shows Americas (particularly the U.S.) taking **76% of bot-driven API attacks,** with Western Europe and APAC growing steadily.

What This Tells Us

1. APIs Are The New Front Line

Attackers know your APIs do the real work—so they attack them directly.

2. Scale is Crushing

Hundreds of thousands of fake API calls can drown your service if you don't have real-time defenses in place.

3. Industry Focus Matters

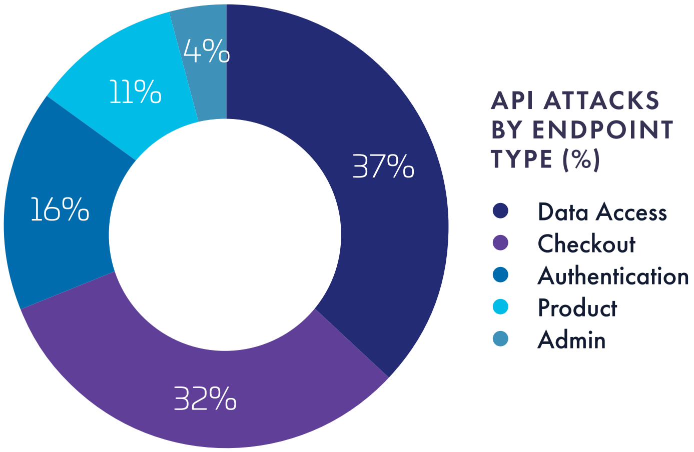
Financial services feel the heat most, but no sector is safe—retail, healthcare, even small SaaS apps see rising bot activity.

Bottom Line

If you treat APIs like "just another web app," you're leaving your most critical systems exposed. It's time to shift your defenses to protect the API layer first—and every statistic above shows why.

3.3 Which Endpoints Attackers Target (and why)

Attackers don't spray your entire API surface at random—they zero in on the endpoints that matter most to their goals. By studying traffic patterns across thousands of protected APIs, Imperva Threat Research has identified clear favorites among bad actors. Understanding why these endpoints are targeted—and how attackers adapt—lets you focus your defenses where they'll have the greatest impact.



BACK TO TABLE OF CONTENTS

Data-Access Endpoints (~37% of API Attacks)

Why It Is

Endpoints that return user profiles, transaction histories, product catalogs, or any sensitive dataset.

Why Attackers Focus Here

Valuable intelligence
Stolen PII and business data can be sold, used for phishing campaigns, or leveraged to customize fraud.

Low friction
Many read-only APIs lack strict write-protect measures or anomaly checks, making them easier to scrape at scale.

Stealthy exfiltration
Bots can throttle their requests to fly under volume-based alerts, blending in with normal usage patterns.

Imperva's Key Takeaways

- **We observed bots switching from broad "dump everything" scraping to targeted grabs of high-value fields (e.g., email + payment method) when they detect rate limiting on full records.**
- **Fast-moving verticals like e-commerce see spikes in data scraping around product launches, as attackers hunt for competitive pricing or launch their own dark-web shops.**

Checkout & Payment Endpoints (~32%)

Why It Is

Endpoints that handle cart updates, order placement, promo-code application, and payment authorization.

Why Attackers Focus Here

Direct revenue theft
Successful attacks translate immediately into free goods or stolen funds.

Complex workflows
Multiple steps (cart >> promo >> payment) introduce logic gaps—bots exploit any missing validation.

Coupon abuse
Automated loops can stack or reuse discount codes faster than your manual rule updates.

Imperva's Key Takeaways

- **"Promo-loop" bots use a feedback loop: they try a code, detect success/failure, then pivot to the next code without pausing—often 1000+ attempts in minutes.**
- **Checkout-flow attacks spike during high-traffic events (Black Friday, product drops), leveraging the rush to camouflage fraudulent transactions.**

Authentication Endpoints (~16%)

Why It Is

Login, token-exchange, and session-validation APIs.

Why Attackers Focus Here

Account takeover (ATO)
Once inside, attackers can access personal or financial data and potentially pivot to other systems.

Token theft
Stolen JWTs (JSON Web Tokens) or cookies allow seamless session hijacking without repeated logins.

Weak MFA coverage
Many APIs still don't enforce multi-factor on every critical action, leaving gaps bots can slip through.

Imperva's Key Takeaways

- **Credential-stuffing bots now use distributed networks of residential proxies and randomized user-agents to stay below lockout thresholds—making slow, stealthy ATO campaigns increasingly common.**
- **We saw a 40% rise in login-failure anomalies in the first half of 2025, especially on APIs without adaptive MFA or risk-based challenges.**

Gift-Card & Promo-Validation Endpoints (~5%)

Why It Is

Endpoints that check gift-card balances or validate promotional codes.

Why Attackers Focus Here

Low-value, high-volume fraud
Small amounts per request reduce alerting risk but add up quickly.

Forgotten endpoints
These are often marked as "internal" or "legacy," receiving fewer security reviews.

Automated reconnaissance
Bots cycle through thousands of codes until valid ones pop up.

Imperva's Key Takeaways

- **Gift-card cracking campaigns often run overnight or during low-traffic windows, exploiting off-peak velocity baselines.**
- **Introducing step-up challenges (e.g., SMS OTP) after a few balance checks stopped 90% of these automated probes in our tests.**

Shadow or Misconfigured Endpoints (~3%)

Why It Is

Hidden or poorly documented APIs left active after development or testing.

Why Attackers Focus Here

Zero protections
These endpoints often bypass WAFs, bot filters, and rate limits entirely.

Unexpected functionality
They may expose admin or debug operations, offering a direct line into internal systems.

Imperva's Key Takeaways

- **On average, we detect 10–20% more API endpoints than organizations believe they have, many of which are under protected.**
- **Regular, automated API discovery scans reduced shadow-API attack volume by 60% when combined with immediate blocking rules.**

Bottom Line

Most API attacks concentrate on a handful of critical endpoints—data-access, checkout, and authentication—because that's where attackers find the highest return on their effort. By focusing discovery, protection, and monitoring on these key areas (and hunting down forgotten or shadow APIs), you can dramatically reduce your risk and stop the attacks that really matter.

Insight

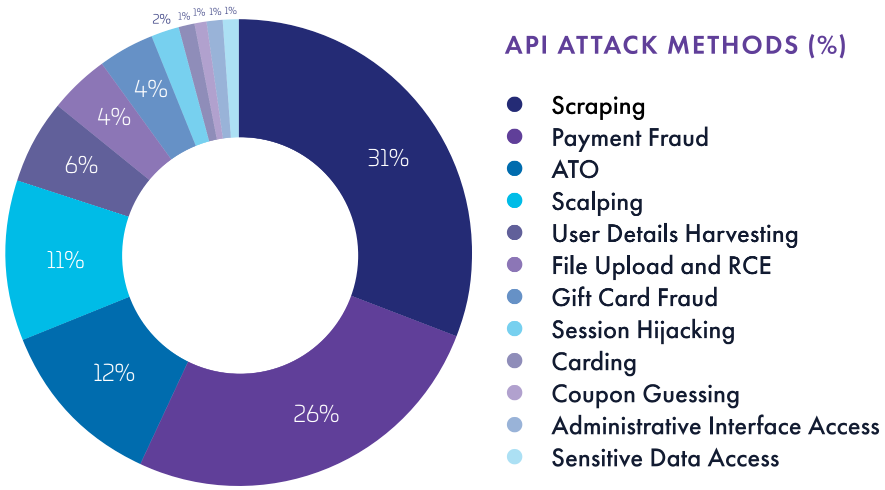
These patterns suggest intentional, highly strategic reconnaissance, not random exploitation.

BACK TO TABLE OF CONTENTS

4. Threat Actor Behaviors & Tactics

4.1 Attack Methods

In our analysis of the first half of 2025, we've seen these attack methods in action across thousands of API environments. Each tactic isn't just a theory—it's pulled straight from Imperva telemetry and real-world investigations. Understanding these patterns, and how attackers adapt, is key to building defenses that actually work.


API ATTACK METHODS (%)

Data Scraping

(~31% of API Bot Attacks)

We saw bots quietly pull data from "data-access" APIs, often grabbing only the highest-value fields (email, purchase history, pricing) to stay under volume alerts. Attackers do this because read requests are low-risk and easy to monetize — the data fuels fraud, targeted phishing, or resale.

Takeaway

Treat every read as sensitive; add field-level filtering, strict rate policies per user, and anomaly detection tuned to record-by-record exfil patterns.

Account Takeover (~12%)

Credential-stuffing bots probe login APIs using leaked username/ password lists; once they get in, they change payment info or access private data. Attackers favor this because stolen credentials are cheap and account access is high value.

Takeaway

Deploy credential-stuffing defenses (device fingerprinting, distributed lockout logic), add adaptive MFA for abnormal login patterns, and monitor post-login behavior for signs of fraud.

Payment & Coupon Fraud (~26%)

Bots repeatedly exercise checkout and promotion endpoints to stack coupons, validate stolen cards, or execute fake transactions. These attacks work because checkout flows have many logic steps and inconsistent validation — one weak spot lets the whole flow be abused.

Takeaway

Enforce server-side promo rules, apply adaptive velocity limits on promo use, and require risk-based step-up checks on suspicious transactions.

Scalping & Inventory Hoarding (~11%)

Bots automate reservations and checkout to grab limited items the moment they drop, then resell them. This is effective because human users can't match the speed, and bots exploit brief windows of weak validation.

Takeaway

Protect release events with tokenized queues, per-user purchase caps, and bot challenges that slow unknown clients without harming real users.

Gift-Card Cracking (~4%)

Automated scripts cycle through gift-card or voucher numbers against balance-check APIs until valid codes surface. It works because many balance endpoints are low-friction and lack velocity controls.

Takeaway

Add rate limits tailored to balance checks, require authentication or step-up for multiple checks, and monitor off-peak spikes (these campaigns often run overnight).

Remote Code Execution (~4%)

Attackers probe for known CVEs (e.g., Log4j) in middleware and gateways to run arbitrary code on servers. A single successful RCE can give full access and is therefore a top prize.

Takeaway

Prioritize dependency audits and rapid patching, block known exploit signatures at the edge, and monitor unusual child processes or outbound connections from API hosts.

Session Hijacking (~2%)

After stealing or guessing tokens, attackers replay session tokens or JWTs against API endpoints to act as real users. This bypasses typical login checks and can be hard to detect.

Takeaway

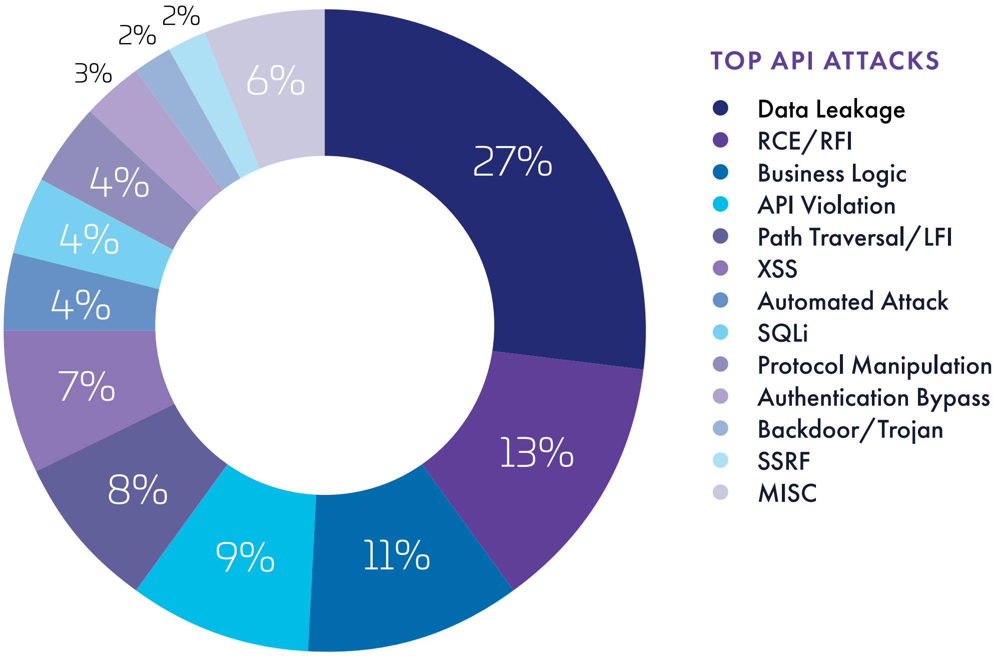
Make tokens short-lived and scoped, validate token signatures and device context, and implement replay detection for repeated token use from new IPs/devices.

Bottom Line

Attackers aren’t randomly probing your APIs—they’re strategically targeting the most valuable operations: data retrieval, payment flows, authentication, and inventory management. To fight back, focus your defenses where the business impact is highest: protect data-access endpoints, lock down checkout and promo logic, harden login APIs, and monitor for token misuse. By understanding each tactic and its real-world use case, you can tailor controls that stop these sophisticated attacks in their tracks.

BACK TO TABLE OF CONTENTS

4.2 Top API Attacks — High-Level Categories & Insights



Attackers aren't guessing — they know exactly what works.

The first half of 2025 reveals a clear pattern: threat actors are focusing on specific, high-impact API weaknesses. From silent data leaks to full system compromise, each tactic reflects a strategic choice, not random noise.

Key Observations

Data Leakage Is the Silent Menace (~27%)

In the first half of 2025 we saw many attacks that simply read data from APIs—often record by record to stay under volume alarms. This happens because APIs frequently expose too many fields by default or lack object-level checks, and bots have learned to scrape just the high-value pieces (email, payment info).

Takeaway

Enforce object-level authorization and field filtering so "read" calls only return what the caller is allowed to see.

RCE Remains a Top Concern (~13%)

Attackers still probe APIs for known CVEs (Log4j, WebLogic, etc.) because a single successful exploit can give them full control of a host. These probes are cheap to run at scale and often succeed where unpatched libraries remain in the stack.

Takeaway

Prioritize patching and place runtime protections/WAF rules close to middleware layers to block exploit payloads.

Business-Logic Abuse Demands Contextual Defenses (~11%)

Bots now imitate normal user flows—looping promo codes, faking transactions, or exploiting refund logic—so their requests look valid but damage business rules. This happens because feature teams rarely test for abuse scenarios and defenses focus on malformed traffic instead of workflow anomalies.

Takeaway

Monitor business metrics (promo redemptions, refunds, reservations) and enforce runtime contract checks to spot "valid-but-fraudulent" activity.

Injection Attacks Persist (~11%)

Classic injection attacks remain common where input validation is weak or legacy code is in use. Attackers can manipulate queries or responses and gain access to data or user sessions when endpoints don't sanitize inputs.

Takeaway

Use parameterized queries, strict input validation, and test APIs regularly with focused injection tests.

API Violations (~9%)

Many hits labeled as "violations" are bots calling undocumented methods or sending unexpected fields — often revealing test endpoints or forgotten features. These reveal gaps between what's documented and what's running in production.

Takeaway

Validate traffic against OpenAPI/GraphQL schemas and shut or secure any endpoints that aren't part of the documented contract.

Path Traversal & LFI Are Low-Noise, High-Reward (~8%)

When file-handling endpoints don't normalize or validate file paths, attackers use "../" tricks to read server files or include malicious files. These bugs persist because file handling often gets little security review compared with core logic.

Takeaway

Normalize and validate paths, sandbox file operations, and restrict file access to safe directories only.

Protocol Manipulation (~4%)

Advanced bots tweak HTTP/2 frames, headers, or chunked encodings to evade signature-based filters and slip malicious payloads past simple WAFs. This works because many defenses don't fully reassemble or validate complex protocol frames before applying rules.

Takeaway

Use API-aware gateways that reconstruct and validate full requests, not just pattern-match raw payloads.

Authentication Bypass Undercuts Access Controls (~3%)

Some endpoints accept weak tokens or skip strict checks for internal convenience; attackers exploit that inconsistency to call protected APIs without proper credentials. These gaps often come from inconsistent token validation across microservices or lax partner APIs.

Takeaway

Standardize token validation, shorten token lifetimes, and require strict scopes or mutual TLS for sensitive calls.

Deeper Insights & Takeaways

A Quarter of All Attacks Steal Data

Even if you patch every RCE CVE, you'll still lose information unless you lock down "read" permissions with object-level controls.

Logic Attacks Require Behavioral Defenses

Since business-logic abuses look like valid requests, you need anomaly-based monitoring—tracking unusual coupon usage or odd transaction sequences.

Protocol Tricks Undermine Signature Rules

Relying solely on signature-based WAFs leaves you blind to HTTP/2 or chunked-encoding exploits; deep-packet inspection or dedicated API gateways are essential.

Hidden Endpoints Lurk Behind Violations

Frequent "API violations" often point to undocumented or forgotten APIs—audit and close them or bring them under full protection.

Bottom Line

Attackers diversify their methods—scraping data, running code, tweaking logic, and even bending the HTTP protocol itself. By knowing which tactics dominate and why they work, you can deploy targeted defenses: strict object-level authorization for data access, schema validation and behavior analytics for logic abuse, comprehensive patching for RCE, and deep-protocol inspection to catch evasive attacks.

That's how you turn the pie chart on **Page 16** from a list of threats into a roadmap for stronger API security.

BACK TO TABLE OF CONTENTS

4.3 The API Logic Exploit Crisis (BOLA & business-logic abuse)

Real-World Examples

Promo-Loop Attack

A bot applies the same discount code 1,000 times in minutes—draining thousands in unintended credits.

Cart-Validation Abuse

By tweaking quantity parameters, attackers simulate legitimate orders, reserving inventory without payment.

Gift-Card Cycling

Automated scripts try millions of gift-card numbers until valid one's surface, then redeem balances before detection.

Shadow-Endpoint Bypass

Hidden test APIs (left over from development) let attackers skip multi-step authentication and jump straight to account data.

Why Business-Logic Attacks Are So Dangerous

Invisible to WAFs and Signatures

Because each request follows the documented API contract, traditional firewalls and signature-based systems see them as harmless.

High Impact, Low Noise

Bots can quietly run thousands of requests per minute, draining funds or harvesting data without triggering volume alarms.

Context-Specific Exploits

Every API is unique—attackers study your exact workflows (promo codes, checkout steps, gift-card checks) and automate them at scale.

Why This Matters

Traditional security controls don't recognize these as malicious. They're "valid" requests doing invalid things.

4.4 Tools & Automation Used By Attackers

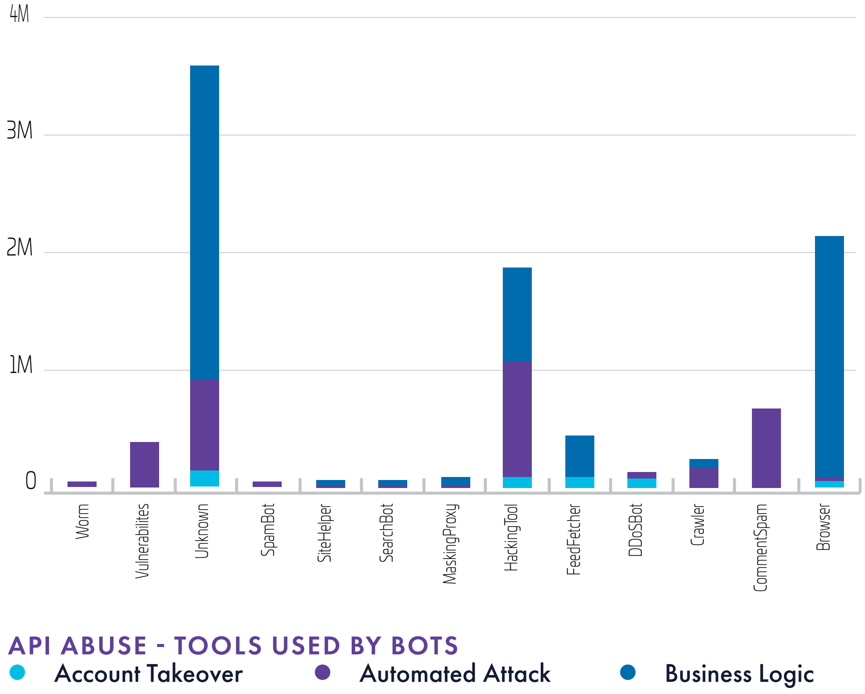
Our Threat Research team observed that attackers now rely on a compact, powerful toolkit—headless browsers (Puppeteer, Selenium), proxy/botnet pools, payload generators (Burp, Postman), and ready-made exploit frameworks—to probe and abuse APIs while pretending to be real users. These tools let bots render JavaScript, rotate IPs, fuzz parameters, and launch CVE-based exploits, making simple signature or IP-blocking defenses ineffective.

Below are the most common tools we see:

Browser Impersonation As An Attack Tool
Attackers now make bots look like real browsers — Chrome, Safari, and others. These bots can run JavaScript, load pages, and act like humans, which lets them slip past simple checks and user-agent filters.

Botnets & Proxy Pools
Rotate IPs and geo-locations to avoid rate limits and reputation blocks, making distributed attacks look "normal."

Payload Generators (Burp, Postman Scripts)
Automate custom request chains and parameter fuzzing to find logic gaps or hidden endpoints.



API ABUSE - TOOLS USED BY BOTS

Exploitation Frameworks (Metasploit, CVE kits) Supply ready-made exploits (e.g., Log4j) that can turn a single vulnerability into a full server compromise.

Custom Scripts & Credential Lists
Stitch together cracked credential dumps, token replay tools, and workflow automations for credential stuffing and ATO campaigns.

Modern bots combine these tools to look human — simulating mouse movement, timing variation, and full browser behavior. That makes signature-only defenses ineffective.

Quick Defense Insight

Treat requests as suspect until proven human: use behavioral signals + device fingerprinting, enforce runtime schema checks, and apply risk-based friction (PoW / step-up MFA) on sensitive actions.

Bottom Line

The attacker toolkit is cheap and powerful. Your defenses must be behavioral, context-aware, and tied to business logic to stay ahead.

BACK TO TABLE OF CONTENTS

5. Emerging Exploit Trends

(H2 2025 signals) (BOLA & Business-Logic Abuse)

First Half 2025 Snapshot

Attackers are not inventing new physics — they're combining smarter automation, cheap infrastructure, and weak operational hygiene to turn small gaps into big wins. Below are the top evolving tactics we observed, each with a short explanation and a practical "highlight" you can act on.

5.1 Misconfigured Third-Party Integrations

Bots increasingly probe vendor/SaaS/cloud APIs that are loosely connected to core systems. These endpoints often have broad permissions and minimal vetting. We saw attackers abuse a misconfigured marketing API as a backdoor into user data – essentially "logging in through the partner." Our interpretation: your partners' APIs are part of your attack surface. Audit their scopes, remove unused privileges, and include them in continuous API discovery and pen-testing.

5.2 Parameter Tampering

Instead of malformed requests, bots now tweak query parameters and request sequences to subvert business logic. For example, our analysis found scripts adjusting price or quantity fields mid-checkout to generate free orders, or sequentially querying user accounts to escalate privileges. These look syntactically valid – a WAF sees no SQL or XSS. The root cause is trusting client input. We conclude that defenses must validate semantics: enforce server-side rules (e.g. "price must match known SKU pricing"), perform runtime schema checks, and monitor unusual patterns in business KPIs (e.g. an order total that doesn't match price & quantity).

5.3 Shadow / Unauthenticated APIs

Attackers routinely scan for hidden or test endpoints that slipped past inventories. These dev-backdoors or mobile APIs often lack auth or throttling. We documented cases where a forgotten API route allowed data dumps or bypassed multistep flows entirely. This is pure operational drift: rapid deployments leaving blind spots. In our analysis, shutting these down is a top priority. We advise forcing all traffic through a central gateway and discovery system and applying the same object-level auth and rate limits to internal or shadow routes as public ones.

These represent new blind spots requiring fresh thinking in API governance. Common thread across tactics: attackers seek maximum payoff for minimal noise — they prefer to abuse legitimate functionality, hide in normal traffic patterns, and exploit operational blind spots (shadow APIs, third-party trust, inconsistent auth).

6. Business & Regulatory Impact

API breaches don't stay in the data center — they hit the whole business. Here's what's at stake and why leaders must care.

Key Consequences

Financial Loss

Fraud, theft, promo abuse, and subtle revenue leakage directly reduce income.

Reputation Damage

Negative press, customer churn, and lasting loss of trust that's hard to rebuild.

Regulatory Fines & Legal Risk

GDPR, HIPAA, PCI-DSS and similar rules can trigger major penalties and costly remediation.

Because APIs connect payments, identities, and sensitive records, a single exploit often cascades customers complain, payments fail, regulators get involved, and normal operations slow or stop. That's why API security is not just a tech problem — it's a business-continuity issue.

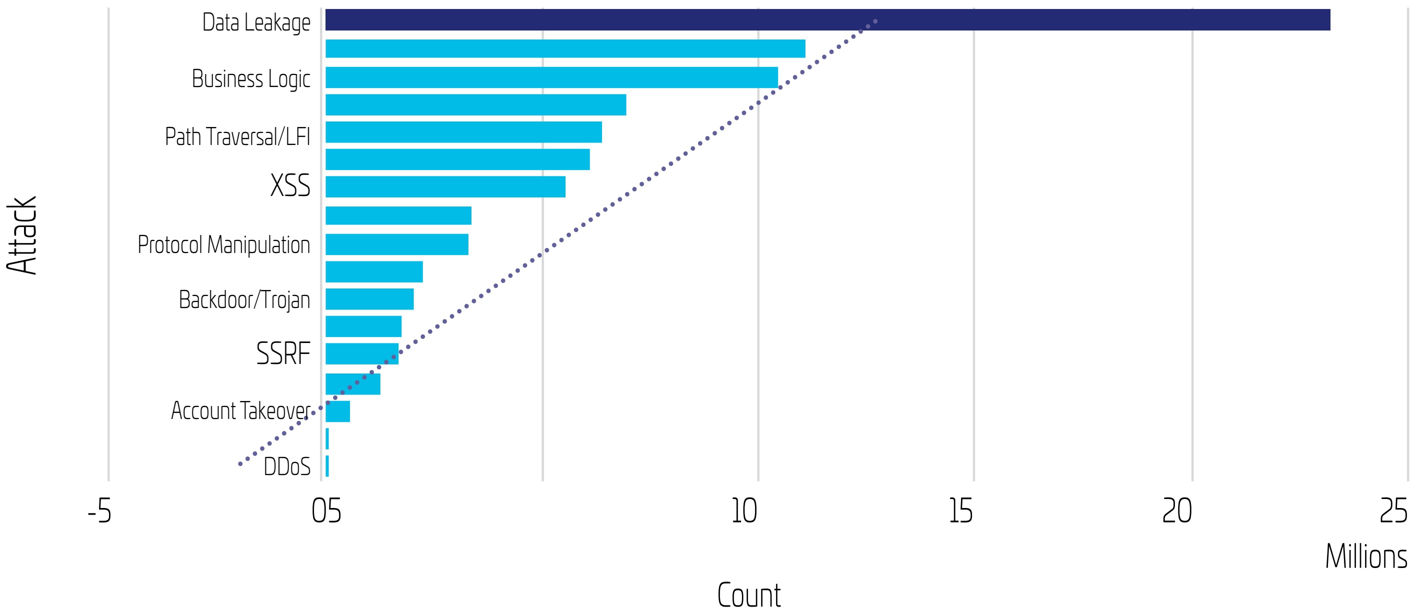
Bottom Line

Protect the APIs that would break revenue, trust, or compliance if they fail — and make API security a board-level priority.

7. Types Of Attacks Against API Endpoints

In the first half of 2025, across every CWAF-protected API endpoint we monitor, attacks fall into a few clear buckets. The chart below shows their relative frequency—with data leakage leading, followed by remote-code-execution (RCE) probes, then automated scanning, parameter-tampering, and injection attempts

API ATTACK FREQUENCY



Interpreting The Chart

Data Leakage Dominate

Nearly one-third of all incidents are simple "read" requests that shouldn't have succeeded. Attackers exploit endpoints that return too much data.

RCE Comes Next

A quarter of attacks aim to drop malicious payloads—proof that known CVEs in API middleware remain a critical risk.

Scanning & Tampering Follow

Attackers continuously map your API landscape, then manipulate parameters to bypass business rules.

Why This Matters

1. **BOLA & Missing Auth Are Silent Killers**
Data-leakage attacks thrive where APIs assume every request is valid. Without per-object authorization checks, attackers can "walk" through your data, one record at a time.

2. **Unpatched CVEs Fuel RCE Campaigns**
Even well-maintained environments see daily probes for Log4j and WebLogic flaws. A single unpatched library can let attackers pivot into internal systems.

3. **Shadow APIs Enable Reconnaissance**
Automated scanners uncover forgotten or undocumented endpoints—often the weakest links. Attackers then target those with parameter tampering or injection.

4. **Business Logic Needs Its Own Guardrails**
Simple input sanitization stops classic injections, but context-aware defenses (schema enforcement, strict contract validation) are required to block tampering and automated abuse.

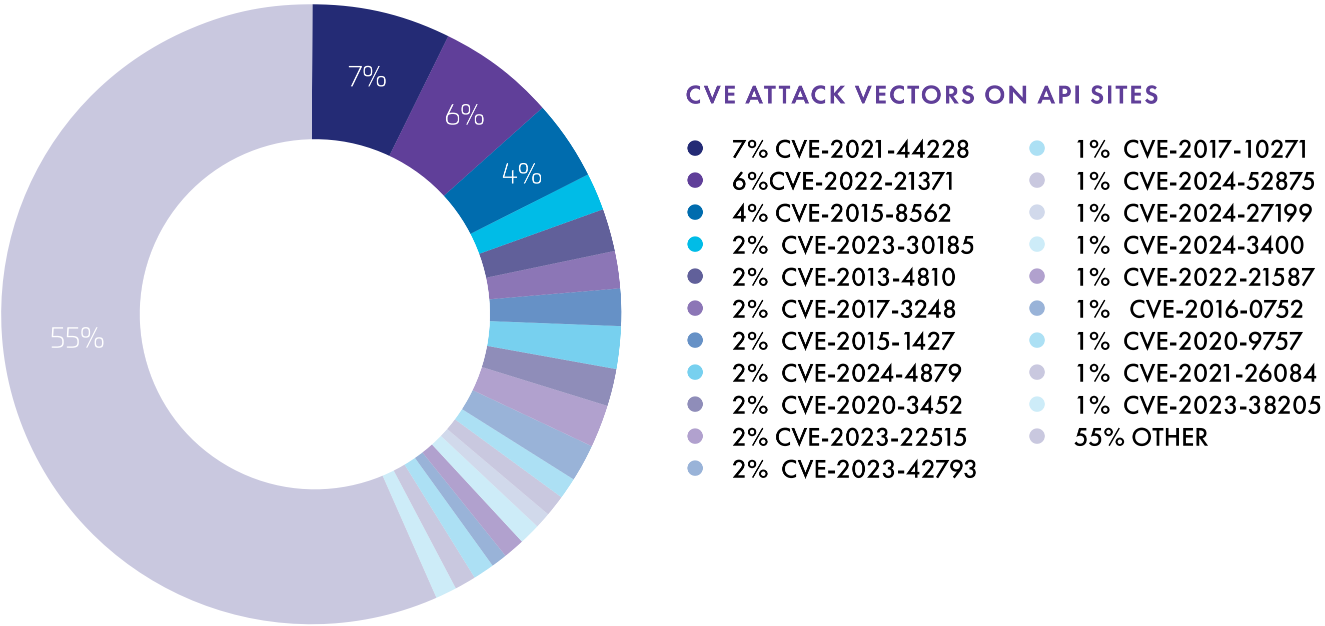
Bottom Line

Treat every API endpoint as untrusted until proven otherwise. Enforce object-level authorization on all "read" requests, keep platforms fully patched, and deploy runtime schema validation to catch extra or malformed fields.
By understanding which attack types dominate—and why—you can apply the right defenses exactly where they're needed most.

BACK TO TABLE OF CONTENTS

7.1 Common CVE Attack Vectors on API Sites

In the first half of 2025, our telemetry shows that when attackers go hunting for known software flaws on API endpoints, they focus overwhelmingly on three families of vulnerabilities. The pie chart below breaks down the share of CVE-based probes we saw:



Log4j Exploits
~50%
of all CVE probes

Oracle WebLogic Flaws
~30%

Joomla Vulnerabilities
~20%

Why these CVEs?

Log4j (the "Log4Shell" family) remains the most targeted because it's easy to weaponize via simple log messages—and many API gateways still include Java logging libraries.

Oracle WebLogic servers' power countless back-end services and sometimes sit in front of APIs, making them prime pivot points for attackers.

Joomla is less widespread in enterprise, but misconfigured or outdated CMS-to-API integrations create an open door for automated exploit tools.

Interpreting the Pie Chart

Half of all CVE probes hit Log4j.

Attackers still scan for "${jndi:ldap://...}" payloads in request headers or bodies because a single successful exploit can drop a shell without touching your API code.

About one-third target WebLogic

Bots look for its WLS components—once in, they can move laterally from the management console into microservices behind your APIs.

Roughly one in five focus on Joomla.

These are often forgotten or side-mounted integrations where the CMS serves data to your APIs, so a Joomla flaw becomes an API flaw.

BACK TO TABLE OF CONTENTS

What This Means for You

1.

Prioritize Patching for Log4j

Even if your main code is .NET or Node.js, check every Java library in your API stack and update to the latest patched version.

2.

Harden Your API Gateway

Place a hardened WAF or Runtime Application Self-Protection (RASP) in front of any WebLogic–backed services. Block known exploit payload patterns before they reach the server.

3.

Audit CMS Integrations

If you pull content from a Joomla site into your API, treat it as part of your attack surface: keep it current, limit its API privileges, and monitor it for odd requests.

4.

Use Signature and Behavioral Defenses Together

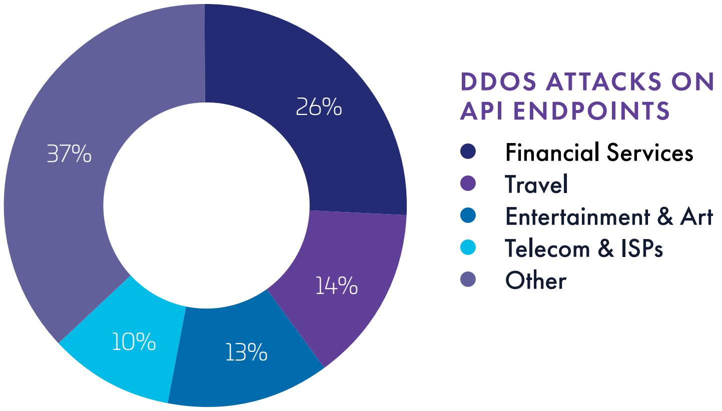
Signatures catch the obvious "${jndi:…" strings, but behavioral analytics spot the follow-on activities (like unexpected outbound LDAP calls) that indicate a bypass.

Bottom Line

Attackers will keep probing the same high-value CVEs as long as they work. By understanding which flaws they scan for—and why—you can focus your patching, WAF rules, and monitoring on these top three vectors, dramatically reducing your API's exposure to automated exploit campaigns.

7.2 Top targeted Industries for API Attacks

Some industries are more heavily targeted because their high API density and sensitive business logic expose the most valuable functions—payments, balance checks, booking systems, SMS/OTP, and user profiles. This section breaks down who's being targeted, why attackers care, and what to protect first.



BACK TO TABLE OF CONTENTS

1. Financial Services

(~26%) take the lion's share of API attacks because APIs there give attackers direct access to money and transactional controls—balance checks, transfers, payment authorizations and KYC flows are all high-value targets.

Takeaway
Treat every financial API as crown-jewel infrastructure — enforce object-level auth, step-up MFA, and immediate token revocation workflows.

2. Travel

(~14%) is targeted for the same economic reasons as e-commerce—bookings, ticketing and loyalty points can be scalped or resold, and fare/inventory data is valuable for undercutting or arbitrage.

Takeaway
Protect booking and fare endpoints with tokenized queues, per-user caps for purchases, and behavior checks during booking spikes.

3. Entertainment & Art

(~13%) draws attacks aimed at streaming accounts, ticket sales and promotion abuse (scalping + credential stuffing). Across these sectors attackers use a mix of quiet scraping, credential stuffing, business-logic abuse and occasional loud plays (RCE or DDoS) when payoff is high or for extortion.

Takeaway
Harden account-login flows with adaptive MFA and limit mass-purchase patterns around ticket drops and promotions.

4. Telecom & ISPs

(~10%) are attractive because their APIs often touch subscription billing, SIM/profile data and SMS/OTP channels (which attackers can abuse for account takeover or to turn services into proxy/bot infrastructure).

Takeaway
Lock down SMS/OTP and provisioning APIs with strict rate limits, multi-party verification, and anomaly detection on SIM/profile changes.

7.3 Common Pattern Across Industries

Across these industries, a common theme emerges: high API density + high-value logic = biggest target. Attackers run campaigns that trigger business processes: imagine thousands of discounted tickets holds or profit frauds sneaking through normal purchasing flows. The Imperva data note that these attacks "fly under generic alerts," demanding endpoint-specific KPIs (e.g. refund spikes, failed payments) to catch them.

Our interpretation is that any API touching finance or identity must have object-level auth and rate controls. For example, we advise tokenizing ticket purchases to enforce one-per-customer and treating promotion redemptions as security signals with caps. In short, defenders must assume "zero trust" even for internal endpoints – every request (even from a logged user) needs fine-grained checks.

BACK TO TABLE OF CONTENTS

8. Strategic Guidance For Executives & CISOs

Together, these recommendations ensure your organization not only closes today's API security gaps but builds a resilient, forward-looking posture that stays ahead of evolving threats.

1. Map API-to-Business Impact

Catalog every API by what it controls – money, PII, critical workflows, etc. Assign each a risk score (Imperva suggests using stats like "37% of attacks target data-access APIs" to prioritize). Track this in risk registers and continuity plans.

2. Assign "API Owners"

For each critical API, designate a product owner (for functionality) and a security liaison (for defense). Embed security champions in dev teams to enforce threat-modeling and schema testing from day one.

3. Metrics & Reporting

Surface API security metrics to executives. For example, show dashboards of total APIs, percentage protected by WAF/bot-mitigation, number of high-risk incidents, and "shadow APIs discovered" per quarter. Translate these to business impact (e.g. estimated revenue at risk from checkout bots) to build budget cases.

4. Integrate into Architecture

Move to a Zero-Trust API model. Require mutual TLS and strict ACLs on every service-to-service call, even internal ones. Use a hardened API gateway as the single-entry point and policy enforcer – no bypass routes.

5. Continuous Testing & Sharing

Institutionalize ongoing red team exercises that focus on API business logic (e.g. large-scale promotion abuse). Join industry threat-sharing groups to swap anonymized API attack signatures and bot fingerprints. The API threat landscape evolves quickly; staying ahead means learning from peers.

6. Business-Level Communication

Translate API risks to business stakeholders. Show Finance how bots starving your checkout affect revenue, or IT how failed OTPs affect churn. In our interpretation, framing API incidents as business problems (not just IT glitches) is the key to executive buy-in.

9. Defense Best Practices For Practitioners (Playbook)

Based on the data, we advise the following hands-on controls:

1. Continuous API Discovery & Classification

Combine network sniffing and automated schema scanning to find every API (especially hidden/test ones). Many orgs find 10–20% more active endpoints than their docs list. Tag each endpoint by data sensitivity (PII, financial, IP) using ML or heuristics. Prioritize protection of those high-risk APIs (note: Imperva saw data-access and payment APIs suffer ~37% and 32% of attacks respectively).

2. Schema Validation & Contract Enforcement

Adopt strict OpenAPI/GraphQL schemas. In CI/CD, reject any update that deviates from approved request/response models. At runtime, enforce incoming payloads to match the schema. This nullifies many parameter-tampering tricks: extra malicious fields or missing mandatory checks will be caught at the gateway.

3. Behavioral Anomaly Detection

Build normal-use baselines per endpoint (calls/hour, geo-distribution, payload stats). Assign a risk score to each request combining factors like IP reputation, bot-like behavior (e.g. no mouse moves), and endpoint sensitivity. For example, an API call from a new device at odd hours on a high-value endpoint should be challenged. Automated systems can escalate or block requests that exceed risk thresholds (e.g. require a CAPTCHA or MFA step).

4. Adaptive Rate Limits & Bot Management

Use dynamic throttling: instead of static limits, throttle based on risk context. A surge to a customer-billing API might trigger aggressive slow-downs, while trusted clients see normal throughput. Layer on silent behavioral fingerprinting (mouse/touch emulation, JS checks) to detect stealthy bots. Then apply targeted challenges (proof-of-work puzzles or light-weight checks) only to suspicious actors, preserving UX for real users.

5. Strong Authentication /Authorization

Implement fine-grained tokens – issue short-lived JWTs tied to minimal scopes, and rotate/revoke them on each session. This limits session hijacking. Require step-up MFA for sensitive operations (fund transfers, admin tasks) but only when needed (i.e. if behavioral signals spike). In practice, this balance prevents credential-stuffing and replay while keeping friction low for bona fide users.

6. Supply Chain & Configuration Hygiene

Continuously scan your API framework and microservice libraries for known vulnerabilities (Log4j, WebLogic, Joomla, etc.) – Imperva sees repeated probes against these years after disclosure. Lint and test infrastructure-as-code (Terraform, CloudFormation) to avoid misconfigurations (e.g. inadvertent public APIs). In short, patch ruthlessly and vet any third-party API integration.

7. Monitoring & Rapid Response

Centralize your logs from WAF, API gateway, bot defense, and SIEM. Correlate anomalies across layers – a DDoS burst coinciding with a credential stuffing spike is at elltale sign of a multi-vector campaign.

8. Prepare API-Specific Incident Playbooks

Steps to revoke tokens, rotate keys, deploy emergency rate-limits, and re-validate API contracts post-incident.

10. Conclusion: Toward Adaptive API Security

Our interpretations from the 2025 data are clear: protecting APIs demands more than firewalls and static rules. Imperva's research emphasizes that "full visibility of every endpoint, behavior-based detection... and protections embedded into the workflows" are now table stakes. In practice this means treating API security as a continuous, business-driven practice, not a one-off IT project. Discover every live endpoint, gauge its business value, and secure it with context-aware, adaptive defenses. As Imperva puts it, the next six months will only see volume and sophistication of API attacks grow, so the "best time to act was yesterday, the next best time is now". By combining threat intelligence, strong governance, and smart tech (from discovery to ML-based bot management), security leaders can turn APIs from weak spots into well-guarded gateways.

To help you move from insight to action, download **Imperva's API Security Buyer's Guide** — the operational companion to this report. This guide hands you the criteria, context, and confidence to select a solution that not only plugs the gaps today — but lets you take full control of your API security posture.

Appendix — Glossary

- **BOLA**
  Broken Object Level Authorization
- **ATO**
  Account Takeover
- **RCE**
  Remote Code Execution
- **PoW**
  Proof of Work
- **JWT**
  JSON Web Token

BACK TO TABLE OF CONTENTS

THALES
Building a future we can all trust

CYBERSECURITY

imperva

Contact us

For all office locations and contact information, please visit **cpl.thalesgroup.com/contact-us**

thalesgroup.com