REPORT

Software Under Siege 2025

What every security leader needs to know about the blindspot in software defense

CONTRAST SECURITY

Table of Contents

Executive summary

Defenders who are responsible for protecting today's applications from attack face relentless pressure on two fronts: a surge in targeted attack activity from determined adversaries, and a growing backlog of serious vulnerabilities. Both trends are exacerbated by the widespread use of AI for software development and generating attacks.

Contrast's runtime security platform continuously monitors and reports on vulnerabilities and attacks that target applications, Application Programming Interfaces (APIs) and libraries, shining a spotlight on the usually invisible front lines of application-layer threats. This deep telemetry provides insights into the inner workings of real-world applications and APIs worldwide, showing that:

- **Apps and APIs have become the battleground of choice for modern attackers.** Application attacks are more prolific than ever before. The average application is exposed to 81 confirmed, viable attacks each month, that evade other defenses, alongside more than 10,000 probes and other unsuccessful attack attempts.
- **At the same time, applications have never been more exposed,** with an average of close to 30 serious, exploitable vulnerabilities per application.
- **The deck is stacked against software developers and AppSec teams.** Applications face an average of 17 new vulnerabilities per month, driven by ongoing development as well as vulnerabilities in third-party dependencies. This rate far exceeds the ability for AppSec and DevOps teams to patch, and the growing use of AI-generated code exacerbates the problem.
- **The challenge is just as daunting for SecOps teams.** Attackers are able to exploit new vulnerabilities in just a few days, while defenders often take months to spot and contain intrusions. A core challenge for SecOps analysts: Traditional detection and response tools employed in the SOC are blind to attacks that target the application layer.
- **A new approach changes the game:** It's urgent for organizations to move beyond traditional application defenses, such as WAF and EDR, and adopt a modern runtime approach to application defense, such as Application Detection and Response. Organizations that do so can eliminate their exposure to the most prevalent application attacks, including deserialization attacks, injection attacks and many more. This not only reduces the risk of breach but also significantly reduces the burden on overworked AppSec, SecOps and development teams.

Runtime data gives a real-world view of application risk

Contrast's data is collected from real-world running applications and Application Programming Interfaces (APIs), using a lightweight sensor that allows full visibility into the complete runtime context. This "inside-out" approach gives us continuous visibility into how applications behave and are targeted in real-world production environments.

As requests flow through monitored applications, the sensors track runtime behavior, including control flow, data flow and backend interactions such as file access, SQL queries and external service calls. This extensive context allows Contrast to determine not just whether a vulnerability exists in an application or library, but whether it is reachable by user input, triggerable in real-world execution paths, and exploitable given the specific data flows and environment in which the application runs.

This report is built from anonymized aggregate telemetry collected from thousands of live applications, incorporating more than 1.6 trillion security-critical observations every day. The result is a high-fidelity view of the most impactful attack techniques in play today, and which vulnerabilities matter most.

Software is under siege

Application-layer attacks have become one of the most common and consequential methods adversaries use to gain access and compromise organizations. These attacks target the custom code, APIs, and logic that power modern applications, often slipping past traditional detection tools such as Endpoint Detection and Response (EDR) and network-based defenses such as Web Application Firewalls (WAFs). Data from industry analysts highlights the challenge.

APPLICATION ATTACKS ARE PROLIFIC

The 2025 Verizon DBIR found that web application attacks were in the top 3 types of reported incidents in 2024.[1] It goes on to show consistent heavy use of vulnerability exploits over the last 5 years.

APPLICATION ATTACKS CAUSE ENORMOUS DAMAGE

IDC reports that application-related catalysts, such as a supply chain attack, unpatched vulnerabilities or a zero day, have triggered recent ransomware incidents for 35% of surveyed organizations,[2] with an average cost of $4.91 million per ransomware incident,[3] according to IBM.

APPLICATION VULNERABILITIES ARE A MAJOR SOURCE OF RISK

Data from Mandiant's M-Trends 2025 report reveals that 33% of breaches began with a vulnerability exploit.[4] Forrester's State of Application Security 2025 shows that application exploits are among the top external attack vectors.[5] Guidepoint reports that the number of new Known Exploited Vulnerabilities (KEVs) increased 75% year-over-year in Q1 2025.[6]

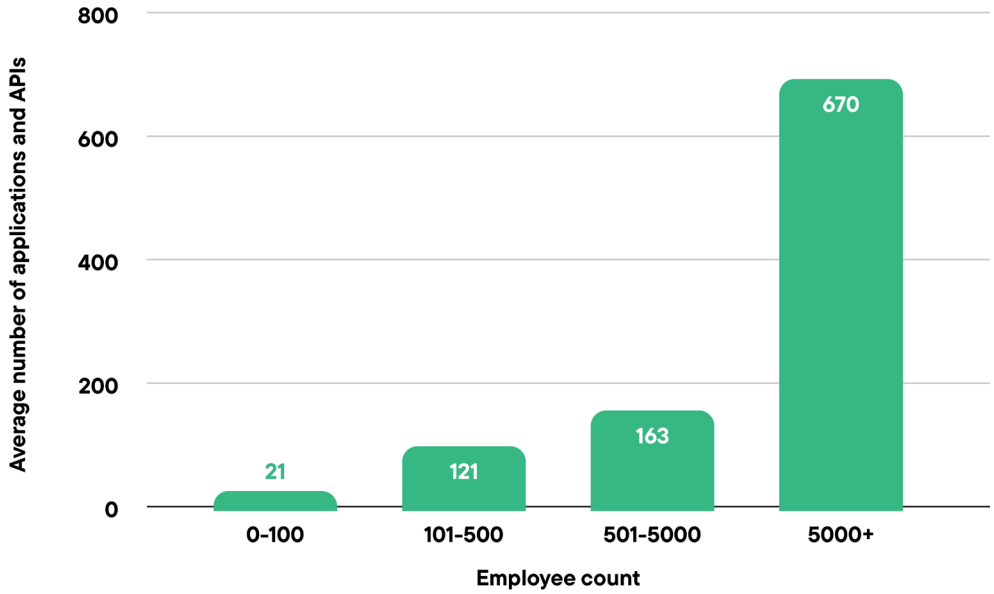
SOC TEAMS STRUGGLE TO DETECT BREACHES IN A TIMELY MANNER

Mandiant's M-Trends 2025 report indicates that more than half (57%) of intrusions are being discovered through external sources[7] rather than internal security measures. This suggests significant limitations in existing detection capabilities within many organizations, leading to lengthy delays in identifying and responding to threats.

Software is under siege (cont.)

On top of all this, the application attack footprint is large and getting exponentially bigger. Figure 1 below shows the average number of applications and APIs within a typical organization, broken down by size, showing that while a small organization may only have a couple of dozen applications and APIs to monitor, larger organizations have hundreds to defend from attack.

Figure 1: Average number of applications and APIs per organization, by employee count



Looking forward, IDC predicts that by 2028, there will be over 1 billion new cloud-native applications to protect.[8] The rapid adoption of AI and LLMs is increasing the challenge; Forrester reports that the number of AI-related APIs increased 807% in 2025, and the number of AI-related CVEs increased 1,025% from 2023 to 2024.[9]

The numbers tell a clear story, but the real-world impact is even more revealing. Many of the most significant ransomware campaigns and similar cyber attacks leverage applications as a point of initial entry. Some notable examples include:

	Threat actor	Attack description	Impact
MOVEit Transfer mass-exfiltration[10]	Clop ransomware group	Exploited MOVEit SQL injection CVE-2023-34362 to infiltrate and steal data from hundreds of organizations.	More than 380 victims' files were posted on leak sites, triggering regulatory disclosures worldwide. Average estimated cost per affected organization: $5.7 million.
ScreenConnect cloud & MSP takeover[11]	Unidentified nation-state threat actor	Exploited CVE-2025-3935 to hijack ConnectWise's ScreenConnect servers and access MSPs and downstream customer environments.	Researchers counted 4,338 internet-facing ScreenConnect servers; forensic work found a subset actively compromised, cascading unauthorized access to hundreds of MSP clients.
Deserialization vulnerability in Veeam Backup & Replication software[12]	Akira and Fog ransomware threat groups	Exploited CVE-2024-40711 on unpatched Veeam servers to gain initial access before engaging in further reconnaissance and lateral movement.	Attackers achieved full system compromise, deployed ransomware and extorted victims for cash. Rapid7 estimates Akira brought in more than $16.7 million in ransom payments alone in 2024.

In this report we'll explore the world of application-layer threats and vulnerabilities, and the actions that organizations can prioritize to reduce risk today. The report combines proprietary data from the Contrast Runtime Security Platform with additional data from trusted third parties to help security leaders understand the scope and nature of application-layer threats. It explores how these threats are evolving, why today's SOC tools often miss them, and what organizations can do to close that visibility gap and reduce the risk.

Our aim is to arm security leaders with data-driven insight and actionable guidance to better understand the scale of application-layer threats and to highlight new strategies for improving prevention, detection and response across this growing attack surface.

The Contrast Graph — unleashing deep application-layer observability

The Contrast Graph lies at the core of the Contrast runtime security platform, powering advanced capabilities including optional Agentic AI workflows that help teams respond faster and fix smarter. The Graph builds a real-time digital twin of an organization's application and API environment, mapping live attack paths; correlating runtime behavior; and exposing how vulnerabilities, threats and assets are connected.

This deep, dynamic context eliminates the guesswork that plagues traditional tools, enabling accurate, automated prioritization and remediation, so teams can focus on real risk and act with confidence. Key security workflows built on the Contrast Graph include:

- **Automated threat detection and enrichment:** Real-time attack detection enriched with execution context, data flows and vulnerability details from live application telemetry.
- **Application vulnerability monitoring:** Continuously monitors known and unknown vulnerabilities in code, libraries and frameworks with full exploitability context.
- **AI-powered remediation guidance:** Contrast's generative AI recommends precise remediation instructions for vulnerabilities, derived from execution context, data flows, library analysis and industry best practice.
- **AI enablement:** Contrast's high-fidelity telemetry offers an unprecedented level of visibility and context into application behavior. This gold-standard data provides the foundation that AI models need to deliver accurate, context-aware security insights throughout the application stack.

The software threat landscape: Attacks are prolific and continuous

The volume and variety of attacks targeting the application layer have expanded significantly. This shift reflects both the growing reliance on custom software and APIs and the evolving tactics of adversaries who have learned where defenses are often weakest. The primary challenge for SOC teams is that effective threat detection at the application layer is not feasible with traditional network or endpoint signals; it requires deeper visibility to spot attacks embedded deep within the application stack.

To build an effective defense, security leaders need a clear view of the patterns behind today's application-layer threats. What kinds of attacks are occurring most frequently? How do they vary by industry or technology stack? And what does that data suggest about attacker intent and strategy?

The following sections provide a breakdown of recent attack activity observed by Contrast. These insights are designed to help SOC teams prioritize detection and response efforts where they will have the greatest impact.

Attack volume

Application-layer attacks are a constant fact of life in the modern enterprise. Contrast's telemetry shows that the average application is targeted by attacks more than 14,000 times each month, which translates to approximately once every 3 minutes.

14,250

Attacks per application per month

Attackers touch running applications

Every 3 minutes

Understanding attack categories

Probe attacks	These typically represent broad, automated attempts to discover vulnerabilities. Contrast's runtime telemetry provides the ability to identify and separate out these spray-and-pray attack attempts that never reach an actual exploitable vulnerability. Probes aren't dangerous on their own, but they provide early indicators of targeting and also provide perspective on the overall threat landscape.
Suspicious attacks	Suspicious attacks go beyond basic scanning. These attacks show clear signs of malicious intent, often including exploit payloads, tampering attempts or evasion techniques. While they haven't been confirmed as successful exploits, they represent credible future threats that deserve attention.
Viable attacks	These are confirmed attacks against reachable and exploitable vulnerabilities. Viable attacks follow confirmed execution paths and manipulate application logic in ways that could lead to compromise. They represent the highest risk and demand immediate attention from security teams.

To understand what these threats look like, we drilled down to understand the nature of the underlying attacks:

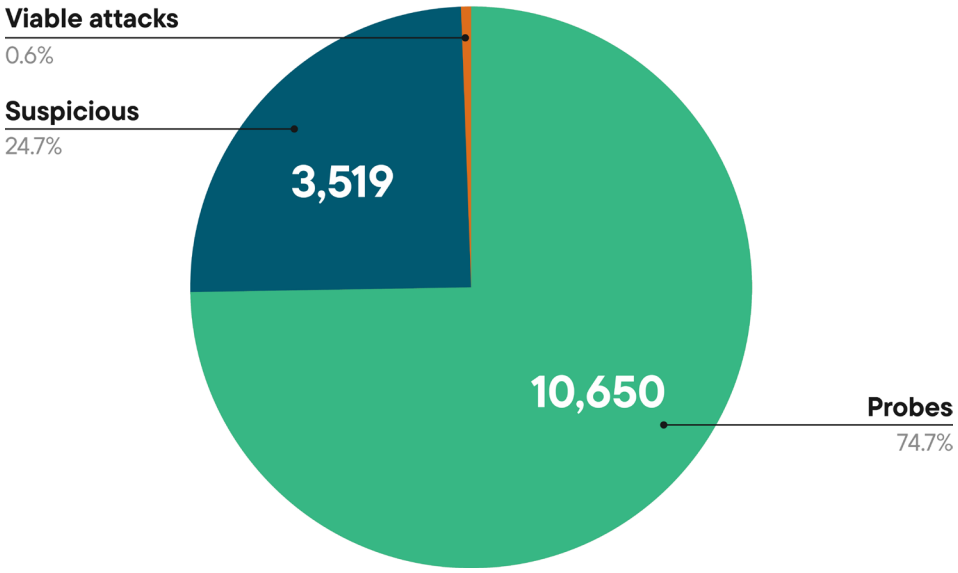


Figure 2: Average number of attacks per application per month, by type

Figure 2 illustrates the average number of attacks per application per month, broken into categories based on the viability of the underlying threat. The overwhelming majority (about 75%) of daily attack activity consists of probes, which are low-level reconnaissance attempts (often automated) frequently used by adversaries to map the environment, identify weaknesses or catalog services. While not immediately harmful, their volume underscores how frequently attackers scan for entry points, reinforcing the need for continuous visibility and monitoring at the perimeter and application layer.

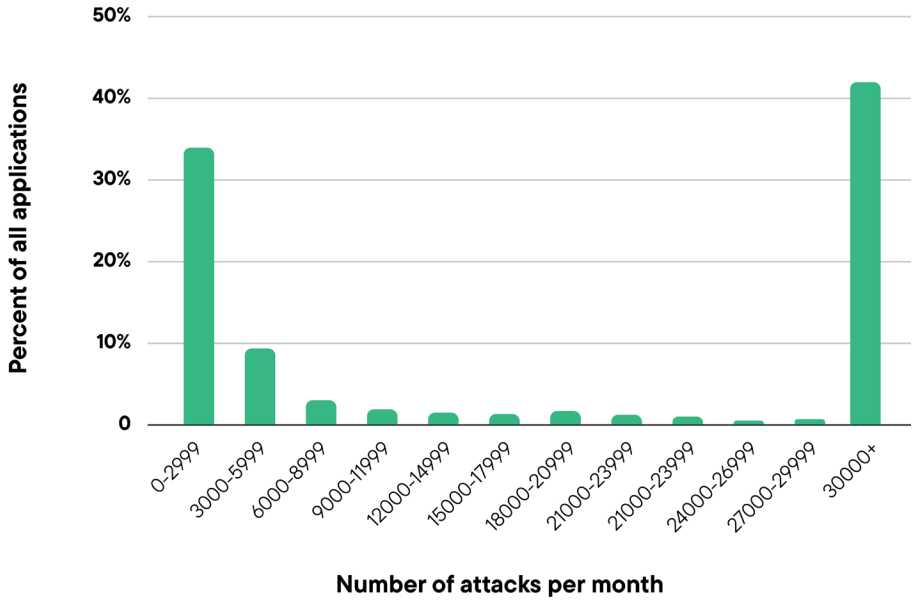
Suspicious activity accounts for most of the remaining attack activity, averaging more than 3,500 occurrences per application per month per application. These are behaviors that exceed benign baselines and warrant investigation. Although not all result in compromise, they can indicate active targeting and testing by attackers and often precede more deliberate actions.

Viable attacks are the most dangerous category we track. These are confirmed exploitation attempts that successfully reach and activate real vulnerabilities in the application. Unlike traditional tools that can bury teams with false positives, Contrast verifies each viable attack through runtime observation, proving that the vulnerability wasn't just targeted, but actually triggered.

On average, every application within an organization faces 81 viable attacks per month, each one targeting a real exploitable vulnerability. In an organization with hundreds of applications, this can quickly add up to thousands of real attacks, creating a huge burden for security teams to investigate and respond to.

The software threat landscape: Attacks are prolific and continuous (cont.)

Figure 3: Attack rate distribution



The histogram in Figure 3 breaks out these averages to reveal how attacks tend to be distributed across applications. The chart shows that a significant portion (more than 30%) of applications experience a relatively low number of attacks (0–2,999 per month). However, another large spike appears at the highest end of the spectrum, with more than 40% of applications receiving over 30,000 attacks monthly. These high-risk applications vary significantly by organization. They may represent applications that are high-value targets for attackers, applications running libraries or frameworks that are being highly targeted with automation, or simply applications that are more exposed to external threats. Smart organizations will analyze their personal threat landscape, identify their most targeted applications and align resources to where they can have the biggest impact, reducing overall risk.

Attack breakdowns

While the volume of application-layer attacks is striking, the nature of those attacks offers deeper insights. By examining which attack types are most prevalent and how they vary by environment, we can gain a clearer picture of attacker priorities and inform more targeted defense strategies.

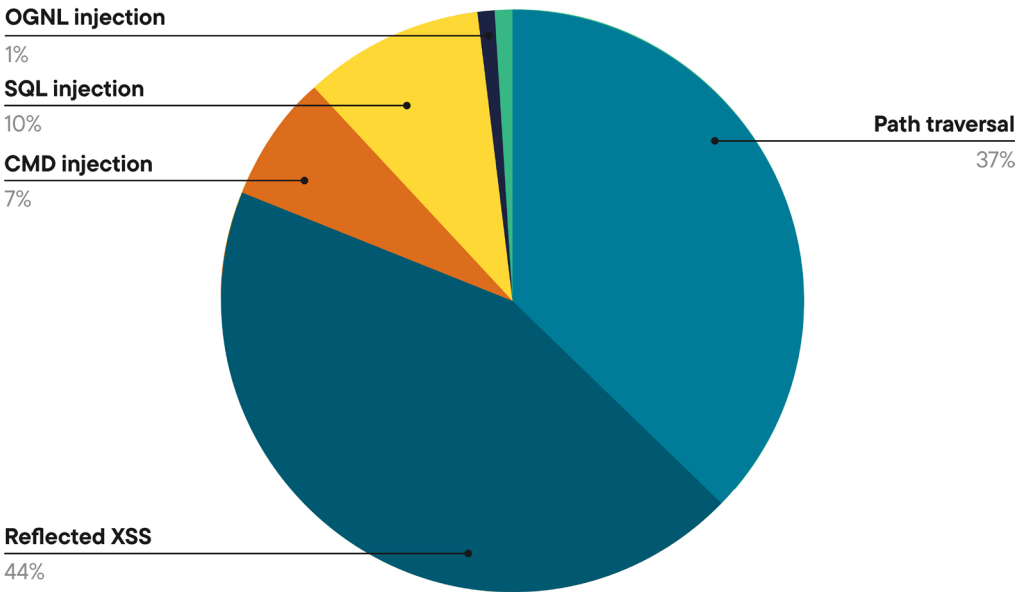


Figure 4: Top types of application probe attacks

In Figure 4, we see that a relatively small number of techniques account for nearly all of the observed probe activity targeting applications and APIs. This concentration shows that adversaries are leaning heavily on a core set of high-yield techniques that consistently produce results when employed across large volumes of targets.

This pattern of concentrated probe activity underscores how attackers prioritize proven, widely applicable techniques to maximize their impact. By repeatedly leveraging a small set of highly effective exploits, adversaries can efficiently identify and compromise vulnerable applications at scale.

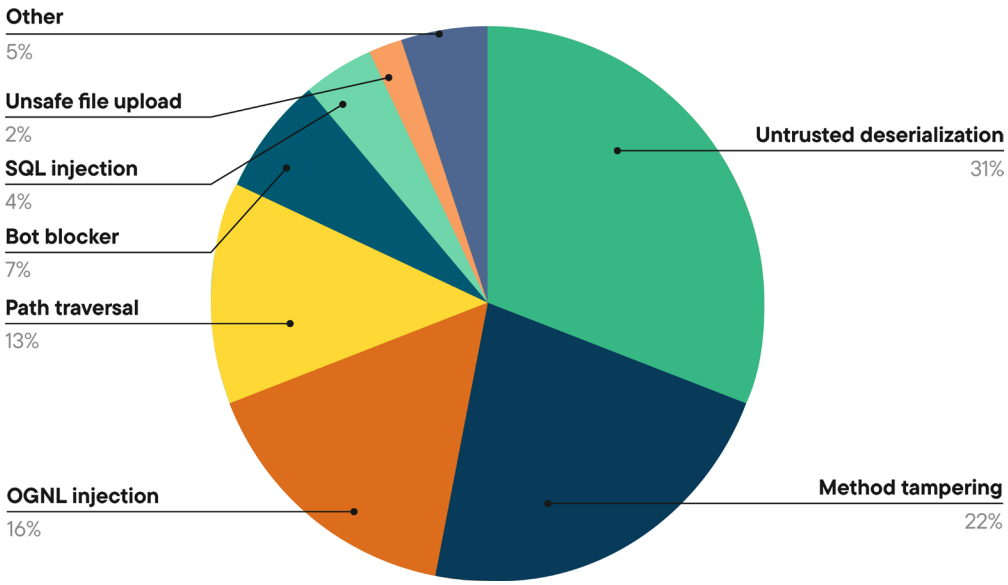
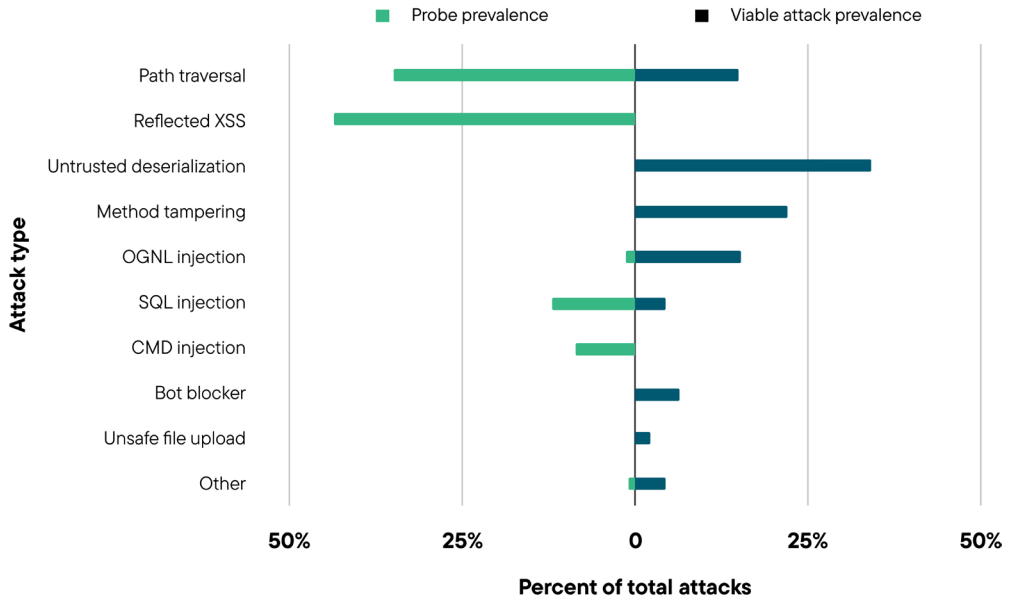


Figure 5: Top types of viable application attacks

Figure 5 shows that the breakdown of confirmed viable attacks paints a different picture, where a more diverse set of threats showed potential for impact in production environments. Untrusted deserialization, method tampering and OGNL injection top the list of most prevalent viable attacks.

In Figure 6, we put probe attacks side-by-side with confirmed, viable attacks to show a stark contrast. Attackers' spray-and-pray techniques, such as path traversal and reflected XSS are highly prevalent, as they are simple to execute at scale. "In practice, however, these techniques do not necessarily translate into the top successful exploitations against modern systems. This underscores the need for deeper context when prioritizing investigations and response; raw numbers rarely provide a complete and accurate picture of risk.

Figure 6: Prevalence of top probes vs. top viable attacks



The software threat landscape: Attacks are prolific and continuous (cont.)

When broken down by industry, the patterns vary depending on the sector's architecture, exposure and threat landscape. Figure 7 highlights the top attack techniques observed across several major industries, based on recent telemetry from real-world environments.

Figure 7: Top five viable attack techniques by industry vertical

	Finance	Health Care	Manufacturing	Technology
#1	SQL injection	Path traversal	JNDI injection	Method tampering
#2	Untrusted deserialization	Bot blocker	Method tampering	OGNL injection
#3	Method tampering	Unsafe file upload	Path traversal	Command injection process hardening
#4	OGNL injection	Method tampering	SQL injection	Path traversal
#5	Unsafe file upload	OGNL injection	OGNL injection	SQL injection

While some attack techniques, such as Method tampering and OGNL injection, appear across multiple sectors, others reflect industry-specific risks. JNDI injection is noteworthy in the Manufacturing sector. Attackers continue to scan broadly for vulnerable JNDI exposures, notably associated with the Log4Shell attacks that gained attention in late 2021. While many organizations have patched this vulnerability, manufacturers are often hobbled by slow OT patching cycles and a heavy reliance on third-party software that may bundle older components, leaving them exposed to these types of attacks.

The most common application-layer attack techniques also vary by development language, influenced by the frameworks, libraries and deployment models typical to each ecosystem. Figure 8 outlines the top observed attacks across popular languages.

Technique spotlight: Method tampering

What it is:

Method tampering (sometimes called HTTP verb tampering) is an attack against HTTP authentication or authorization systems that have implicit "allow all" settings or excessive permissions in their security configuration. This type of attack exploits misconfigurations or vulnerabilities in HTTP method authentication and access control mechanisms.

How attackers exploit it:

By manipulating the HTTP method associated with a request to a web application, attackers can force unauthorized actions like elevating privileges, altering transactions or bypassing authentication controls.

How to defend:

- Ensure only necessary HTTP methods are enabled on HTTP servers. Most web applications require only GET and POST methods. Use a simple allow list to permit only these essential methods while blocking access to high-risk verbs that are rarely needed and often abused.
- Runtime protection and instrumentation can detect unusual control flow triggered by tampered method inputs.

Method tampering is the most consistently prevalent attack across all languages, highlighting it as a universal risk. Untrusted deserialization also appears frequently, particularly in .NET, Java and Python, suggesting widespread insecure object-handling practices. Aside from a few commonalities, it's noteworthy that attacks vary widely with the programming language, pointing to stack-specific vulnerabilities that should guide targeted developer education and mitigation strategies.

Figure 8: Top five viable attack techniques by programming language

	.NET	Java	Node.js	Python
#1	SQL injection	Untrusted deserialization	Path traversal	XXE
#2	Method tampering	Method tampering	Unsafe file upload	Command injection
#3	Untrusted deserialization	OGNL injection	Bot blocker	Untrusted deserialization
#4	Unsafe file upload	Path traversal	Method tampering	Method tampering
#5	Path traversal	Command injection	SSJS injection	Unsafe file upload

Technique spotlight: Untrusted deserialization

What it is:

Untrusted deserialization attacks exploit vulnerabilities in libraries designed to unpack structured data such as JSON, Java objects or PHP serialization. Vulnerabilities arise when these serialized objects are parsed without strict validation or type constraints, opening the window for attackers to inject malicious code.

How attackers exploit it:

Attackers craft malicious serialized objects that, when processed by the application, can trigger remote code execution, escalate privileges or disrupt service. Detection is difficult because payloads appear to external security controls as opaque data blobs, meaning attacks can often bypass traditional security tools.

How to defend:

- Avoid deserialization of untrusted, user-controlled input whenever possible.
- Use safer serialization formats like JSON that don't support complex object references.
- Keep libraries and frameworks updated. Many deserialization attacks exploit known flaws in third-party libraries. Ensure all serialization-related components are up to date.
- Enable runtime protections that detect and prevent deserialization anomalies.

Takeaway: These patterns provide valuable intelligence for shaping SOC priorities and tuning detection rules. They also help teams identify where to focus prevention efforts based on their unique risk profile. Application-layer attack patterns are highly contextual and often invisible to traditional tools. For SOC teams, the ability to distinguish and prioritize these behaviors is essential to delivering a timely, effective defense.

Software vulnerabilities: The cracks are widening

Behind every successful application-layer attack is an unpatched, misconfigured or unknown vulnerability, either in the organization's own code or in the open-source libraries and frameworks it depends on. As applications grow more complex, so too does the attack surface they expose. In this section, we'll examine vulnerability data across a broad set of monitored applications to understand how common application vulnerabilities are and what kinds of flaws are most widespread.

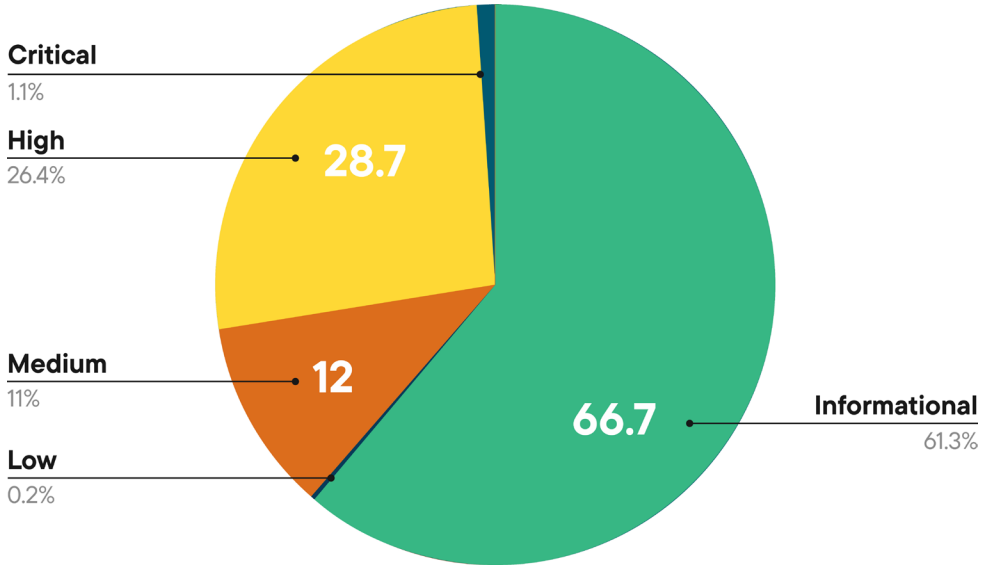


Figure 9: Average vulnerability findings per application, by severity

Figure 9 shows a breakdown of vulnerability observations by severity. While the average application contains dozens of vulnerability findings (an average of 109 per app), the majority of these are observations that pose little actual risk in production applications. The more serious concern lies in the nearly 30 vulnerabilities rated High or Critical (about 27% of the total).

What is a "Serious" vulnerability?

Understanding which vulnerabilities matter most starts with accuracy. Contrast's defend-from-within architecture offers deep insight into application internals, enabling precise identification of which vulnerabilities are truly reachable by attackers.

This deep visibility helps filter out the false positives and unreachable CVEs that often plague traditional SAST, DAST and SCA tools, reducing noise and ensuring a more accurate understanding of actual risk.

For purposes of this report, Serious vulnerabilities are those rated by Contrast as High or Critical severity. These represent the highest combination of likelihood of exploitation and potential impact; the vulnerabilities attackers are most likely to exploit and that can cause the most damage.

While these averages tell the overall story about the volumes of vulnerabilities, they gloss over important insights. Figure 10 helps to break down how risk is distributed across applications. While roughly 62% of applications contain at least one vulnerability finding, only about 15% contain one or more serious vulnerabilities. Serious vulnerabilities are not spread evenly across enterprise applications; they are clustered into a relatively small fraction of applications, underscoring the value of targeted remediation efforts that focus on the riskiest assets.

Figure 10: Percentage of applications with vulnerability findings

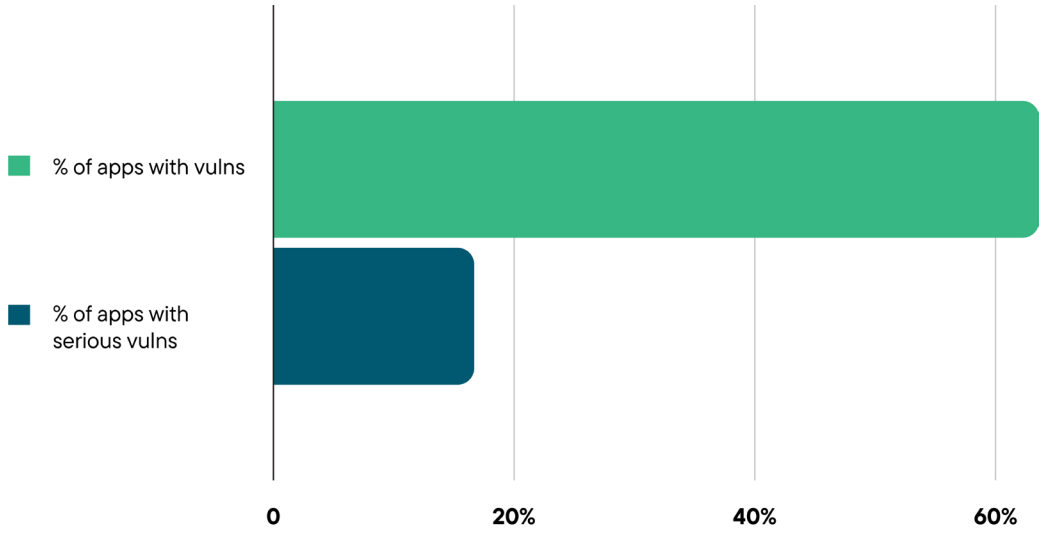
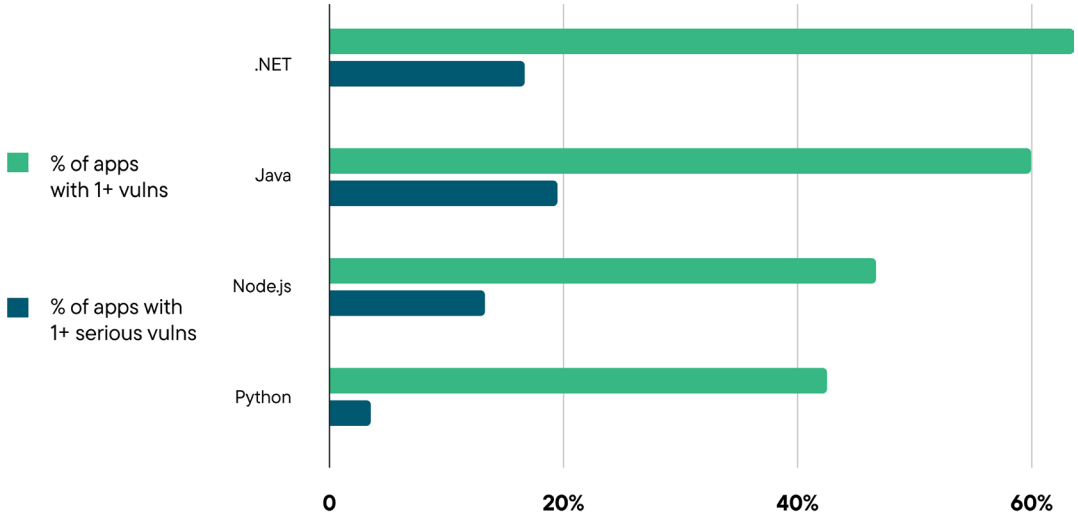


Figure 11 shows that vulnerability prevalence can vary significantly by programming language. .NET applications are the most likely to have at least one vulnerability finding (66%), but they have a low overall rate of serious issues (10%). In contrast, Java applications not only have a high overall vulnerability rate (60%) but also lead in serious vulnerability prevalence at nearly 20%. Node.js and Python applications show lower overall exposure, with Python standing out for having both the lowest overall (42%) and serious (2.7%) vulnerability rates.

Figure 11: Percentage of applications with a vulnerability by agent language



These differences highlight the need for language-specific risk assessments when it comes to building and defending applications. For example, an organization that relies heavily on Java might benefit from targeted investment in developer training and tooling in order to drive down the prevalence of new vulnerabilities in its custom applications.

Takeaway: Application vulnerabilities are pervasive and persistent. Understanding where those vulnerabilities lie and focusing remediation on the most serious, high-risk issues can help security teams reduce their exposure without overextending their resources.

But even with the best intentions and prioritization frameworks, defenders face an uphill battle when it comes to keeping pace with the speed of modern threats. In the next section, we examine how quickly attackers are moving and how difficult it is for defenders to catch up.

Defenders are fighting at a massive disadvantage

Modern attackers are moving faster than ever before, and they are exploiting vulnerabilities in a matter of days, not weeks or months. Unfortunately, defenders are not keeping pace. The gap between attacker speed and defender response has created a high-risk environment where even well-managed organizations may be vulnerable.

Speed of attackers

Recent research from Google and Mandiant shows that the average time between the disclosure of a new vulnerability and active exploitation in the wild is now just five days. That's a sharp decline from previous years, where attackers typically took more than three weeks to weaponize new CVEs.

CrowdStrike's 2025 Global Threat Report paints an even more urgent picture: The average breakout time (the time it takes for an attacker to move laterally after initial access) is just 48 minutes. In some cases, it has been recorded as low as 51 seconds.

5 days

From vulnerability disclosure to exploit[13]

48 minutes

From initial exploit to lateral movement[14]

These numbers underscore a critical reality: attackers are moving with increasing speed and precision, often faster than defenders can identify and react.

Speed of defenders

In contrast, the average time it takes defenders to detect and contain a breach is still measured in weeks or months. According to IBM's 2024 Cost of a Data Breach Report, the average time to identify a breach is 194 days, and the time to contain it is an additional 64 days.

194 days

Average time to identify a breach[15]

64 days

Average time to contain a breach[16]

The challenge is especially steep when it comes to application security. According to Splunk's State of Security 2025, DevSecOps represents the biggest skills gap reported by organizations, with 53% reporting significant skills shortages on their teams. At the same time, nearly 3 out of 4 (74%) identified DevSecOps as one of the top skills needed for a resilient future SOC.

The data reveals a growing asymmetry in the speed of cyber offense versus defense, with attackers operating at a pace that far outstrips defenders' ability to respond. This imbalance gives adversaries a significant tactical advantage, allowing them to exploit vulnerabilities, establish footholds and escalate attacks before most organizations are even aware an incident has occurred. The discrepancy underscores a systemic challenge in cybersecurity: Defenders are playing catch-up in an environment where speed is increasingly the decisive factor. Closing this gap will require automation, faster detection and a shift toward proactive defense strategies.

Why EDR and WAF alone miss application attacks

Research from Contrast Labs reveals that common tools like EDR and WAFs often fail to detect application-layer attacks at the earliest stages, in time to stop a breach.

WAFs, positioned at the perimeter, struggled to distinguish between harmless noise and real threats. Without runtime context, WAFs failed to spot many attacks, allowing applications to be compromised while simultaneously generating an overwhelming volume of false positive alerts.

EDRs, meanwhile, operate at a layer that is frequently too deep to spot application layer threats in time to prevent a breach. Once the compromised app executed the attacker's payload, the activity appeared to the EDR as a legitimate, in-process action. With no application context, the EDR treated it as benign and failed to raise alerts until long after the damage was done.

Contrast's testing used real-world exploits like SQL injection, Log4Shell, and command injection to evaluate how these tools perform. Results showed that ADR (Application Detection and Response) provided superior detection by using deep application context to recognize both effective exploits and harmless probes. This precision enables SOC teams to focus on actual threats while reducing alert fatigue.

WAFs and EDRs serve as important layers in enterprise application defense, but they are insufficient in the battle against today's cyber adversaries. ADR fills a critical visibility gap, one that becomes more urgent as application-layer threats increase in speed and sophistication.

Time to patch

While attackers are accelerating their exploitation timelines, defenders remain mired in lengthy remediation cycles. While a new vulnerability may be exploited in five days, our data shows that the average time to remediate even the most critical vulnerabilities is 84 days. This extended timeline is difficult to compress; patching a production vulnerability requires careful change management, evaluation of dependencies and regression risk, and may be further slowed by limited maintenance windows. That figure alone highlights a dangerous disconnect between threat velocity and security team capacity.

Even when vulnerabilities are being actively worked, the throughput is low. On average, development and AppSec teams are remediating just six vulnerabilities per application per month. For most organizations, that pace is not nearly enough to keep up with the volume of newly discovered issues, let alone clear the backlog.

84 days

Average time to remediate critical application vulnerabilities

6 vulns/app/month

Average number of vulnerabilities remediated per month per application

Defenders are fighting at a massive disadvantage (cont.)

While defenders are working to reduce their exposure, new vulnerabilities are continuously discovered or introduced. Figure 12 shows that the average application sees more than 17 new vulnerabilities each month, with at least two of them rated as Serious. This "Vulnerability Escape Rate" (VER) measures the amount of work defenders have on their plates in order to maintain a steady-state of risk.

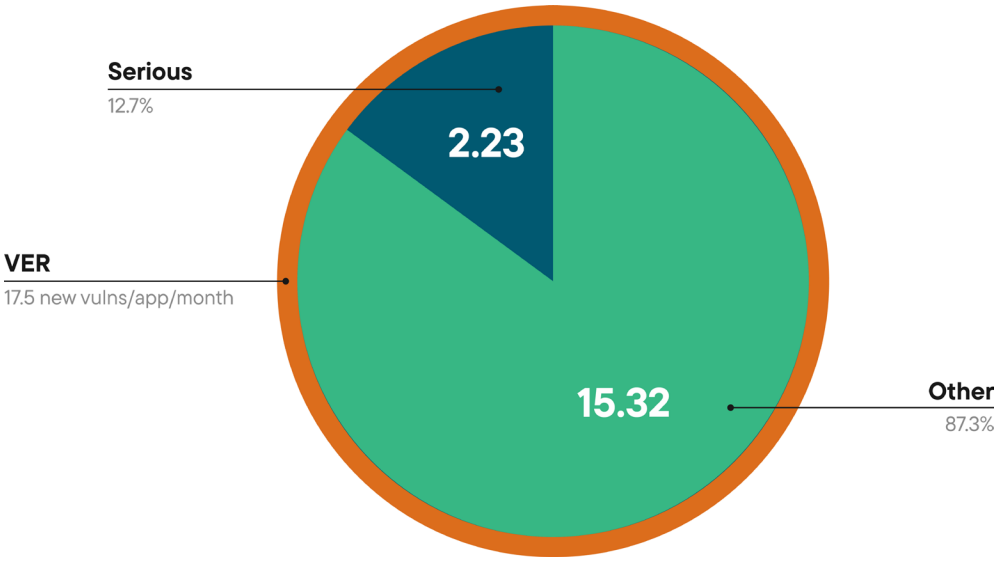


Figure 12: Vulnerability escape rate per application, by severity

When the VER exceeds patching capacity (Figure 13), organizations face a serious challenge, as the vulnerability backlog grows without bound month after month. Organizations that fail to take action to bend this curve can find themselves buried in 10s or 100s of thousands of vulnerabilities in just a matter of a year or two.

Looking forward, we expect that new techniques such as AI-generated code fixes may help improve the rate of patching, but this is counterbalanced by the overall increase in vulnerable AI-generated software. We'll explore this in more detail in future reports, as technology develops and patterns emerge.

Regardless, while we can see that the total volume of new vulnerabilities far outpaces remediation efforts, it's encouraging to see that patching capacity (six vulns/app/month) currently exceeds the rate at which serious vulnerabilities are introduced (about two vulns/app/month).

Figure 13: Vulnerability backlog growth over time

This highlights the value of a proactive, risk-based approach to managing application vulnerabilities, enabling teams to focus their limited resources on the issues that matter most and significantly reduce real-world exposure.

Takeaway: To reduce exposure without overwhelming teams, organizations need to rethink how and where they apply controls, focusing not only on patching but also on runtime protection and smarter prioritization. These stats point to a harsh truth: Many organizations are losing ground in their efforts to patch vulnerabilities and respond to incoming threats.

Proactive application defense changes the game

The data in this report highlights a clear and urgent reality: Attackers are exploiting application-layer weaknesses faster than most organizations can detect or respond. Traditional security operations, built around network traffic, endpoint telemetry and log aggregation, are not equipped to handle the complexity and speed of modern application-layer threats.

An effective approach to managing application cyber risk relies on two critical, complementary workflows:

Risk-based application-layer vulnerability management

We've seen that AppSec and Development teams are often buried in theoretical vulnerabilities; however, typically only a handful have the potential to lead to a damaging application breach.

By focusing attention on the few vulnerabilities that truly matter, security teams can dramatically reduce their exposure to attack.

Organizations with a mature risk-based program to manage application vulnerabilities can essentially eliminate exploitable vulnerabilities from their production applications.

Real-time detection and blocking for application-layer attacks

Even in the best-managed environment, zero-day vulnerabilities happen, and patching takes time. For these reasons, it's critical for security teams to have a plan for detecting and blocking application exploits in production.

Application Detection and Response (ADR) fills the application security gap by eliminating blindspots and protecting applications and APIs from within.

Implementing proactive application defense allows organizations to block the viable attacks that target unpatched vulnerabilities.

Proactive application defense in the real world

Want to see how deploying proactive application defense techniques has helped organizations like yours to slash risk?

See customer success

Combined, these two approaches stop the vast majority of observed application-layer threats in their tracks, freeing up security teams to focus on strategic initiatives while dramatically reducing the risk of a damaging breach.

In order to manage this risk, security teams need to evolve their strategies to address this critical visibility gap. That begins with recognizing that application-layer attacks are not edge cases; they are central to today's threat landscape. To reduce exposure and take back control, defenders should look for solutions that can:

- Deliver deep visibility into application behavior in real time, enabling detection of logic-based, input-driven and credential-based threats that bypass EDRs and firewalls.
- Support a risk-based approach to vulnerability management, helping teams identify and prioritize the issues most likely to be exploited, rather than chasing down every code defect.
- Block exploitation attempts at runtime, particularly when vulnerabilities exist in production systems that cannot be immediately patched.
- Integrate cleanly into the SOC workflow, providing actionable signals that analysts can investigate and correlate across the broader kill chain.

Learn more

Contrast Application Detection and Response (ADR) is designed to meet these needs. It combines runtime exploit detection with deep contextual understanding of application logic and vulnerabilities, giving defenders a fighting chance to detect attacks as they unfold, before a breach occurs.

You can experience what it looks like to detect application-layer attacks in real time and learn how Contrast ADR can help modernize your security stack. Experience a guided walkthrough or get a hands-on demonstration that addresses the attacks on your applications and APIs.

Try Contrast

[1] Verizon 2025 Data Breach Investigations Report, Apr 23, 2025
[2] IDC Market Insights: Application Detection and Response, Feb 2025
[3] IBM Cost of a Data Breach Report 2024, July 30, 2024
[4] Google Mandiant M-Trends 2025, Apr 23, 2025
[5] Forrester Research, The State Of Application Security, 2025
[6] GuidePoint Security, GRIT 2025 Q1 Ransomware & Cyber Threat Report
[7] Google Mandiant M-Trends 2025, Apr 23, 2025
[8] IDC Market Insights: Application Detection and Response, Feb 2025
[9] Forrester Research, The State Of Application Security, 2025
[10] Clop Ransomware Likely Sitting on MOVEit Transfer Vulnerability (CVE-2023-34362) Since 2021, June 8, 2023
[11] ScreenConnect 25.2.4 Security Patch, April 24, 2025
[12] CVE-2024-40711 - Veeam Backup and Replication deserialization vulnerability exploited by ransomware actors, October 30, 2024
[13] How Low Can You Go? An Analysis of 2023 Time-to-Exploit Trends. Oct 15, 2024
[14] CrowdStrike 2025 Global Threat Report, Feb 27, 2025
[15] IBM Cost of a Data Breach Report 2024, July 30, 2024
[16] IBM Cost of a Data Breach Report 2024, July 30, 2024

Contrast Security is the world's leader in Runtime Application Security, embedding code analysis and attack prevention directly into software. Contrast's patented security instrumentation enables powerful Application Security Testing and Application Detection and Response, allowing developers, AppSec teams and SecOps teams to better protect and defend their applications against the ever-evolving threat landscape.

contrastsecurity.com

6800 Koll Center Parkway
Ste 235
Pleasanton, CA 94566
Phone: 888.371.1333