WHITEPAPER

How ADR + WAF raises the bar on application protection

Introduction

The past year saw an 280% increase in application exploitation, reflecting either attackers' growing sophistication or their growing awareness of just how vulnerable applications and APIs can be. The Verizon Data Breach Investigations Report (DBIR)[1] estimates that 68% of incidents are caused by known software vulnerabilities or internal applications. Additionally, a staggering 78% of cybersecurity professionals reported experiencing an API security incident in the past 12 months.

The application layer, comprising server-side applications and APIs, is critical for security, given its important role in business operations. This essential layer handles all company data, including sensitive data like Personally Identifiable Information (PII) and Personal Health Information (PHI). It is typically connected with databases and other applications that can operate outside the organization. Cybercriminals are increasingly targeting this blind spot, as evidenced by the rising number of application-layer attacks.

To defend against these attacks, organizations often rely on Web Application Firewalls (WAFs). However, statistics from Contrast Security highlight the limitations of WAFs and the crucial role of Application Detection and Response (ADR) as a complementary security measure. In February 2025, Contrast found that the average app/API was hit with over 70 attacks per month that were able to navigate the maze of code and find their way to the exact kind of vulnerability they targeted, despite the WAF and other tooling in place.

This whitepaper is intended to highlight two key technologies deployed in the fight against application attacks: WAFs — pervasive in use in many networks across companies of all sizes and sectors; and ADR — not as pervasive, but certainly more effective in sounding the right alarm, at the right time, for the successful attacks.

Understanding WAFs

A WAF is designed to filter, monitor and block HTTP traffic to and from a web application. WAFs act as a shield at the perimeter, inspecting incoming traffic for malicious payloads associated with known attack patterns.

A WAF is able to watch application-level traffic and makes decisions as to allow or disallow that traffic based on the data that is visible over the network. WAF security typically performs SSL termination to watch decrypted traffic for pattern-matching or volumetric attacks.

WAFs offer several advantages:

- **Ease of deployment:** WAFs often require minimal configuration.
- **Perimeter defense:** WAFs positioned at the network or cloud perimeter are uniquely positioned to defend against Distributed Denial of Service (DDoS) attacks and known bad actors.
- **Centralized management:** Security teams can manage WAF rules from a central location, simplifying security policy enforcement.
- **Protection from common attacks:** WAFs effectively block common attacks like SQL injection and Cross-Site Scripting (XSS) by identifying known attack signatures.

However, WAF security tools also have limitations:

- **Limited visibility:** WAFs rely solely on the content of the request (payload) for analysis, making them vulnerable to attacks that exploit application logic flaws.
- **False positives:** Due to their reliance on generic signatures, WAFs can generate false positives, blocking legitimate traffic and disrupting application functionality.
- **Vulnerability to zero-day attacks:** WAFs are ineffective against new and unknown vulnerabilities (zero-day attacks) until their signatures are added to the WAF rule set.
- **Lack of contextual information:** WAFs provide limited information about the attack, making it difficult for security teams to understand the scope and impact of an incident.

WAF		Contrast ADR
Strength	**Weakness**	**Solution**
Protects against common web attacks such as Distributed Denial of Service (DDoS) attacks and certain cross-site scripting attacks. Reduces load off your application servers by blocking network traffic of simple and common web application attacks.	Relies on static signatures or known patterns to identify threats: two methods that sophisticated attackers can evade. High number of false positives or alerts that aren't clearly actionable.	Contrast ADR provides deep visibility into the application layer, allowing you to detect and block attacks at their source before they can cause damage or spread throughout your environment. ADR is designed to minimize false positives and provide actionable insights, enabling you to focus on the most critical threats.

How Security Operations Center (SOC) teams think about WAFs and applications

Do all application attack attempts merit an equal alarm sound? or should "successful application attacks" merit a higher alarm?

If that sounds like a no-brainer, then how can we sift through the attack noise and distinguish what is important and relevant to the security of our application? What is an attack irrelevant to our application?

Waiting for an application to be compromised can lead to catastrophic consequences. Speed is critical to detect an attack before it compromises an application or critical data.

SOCs exist with one primary goal: to detect threats and respond appropriately to stop them. Responsibilities can be quite broad and provide a mission-critical function for the organization. SOC teams often oversee Endpoint Detection and Response (EDR), log aggregation policy, deployment and asset management for Security Information and Event Management (SIEM) platforms, and database encryption, among other roles. They also commonly support security engineering and design implementation and oversee the deployment and management of an automated incident response tool... and the list goes on.

SOC teams, despite the many tools they have in place to protect their environments and infrastructure, typically lack adequate coverage for applications and APIs. For example, EDR and Cloud Detection and Response (CDR) solutions are invaluable in the escalating fight against increasingly sophisticated adversaries. However, these tools do not provide comprehensive protection, particularly for applications and APIs.

While SOC teams have tools that automate certain processes and tasks, much of the workload is cumbersome. Keeping up with the massive volume of alerts on a daily basis, in addition to other responsibilities, is immensely challenging. In fact, many SOCs are so overwhelmed by the sheer number of false-positive alerts from their WAFs that they've been forced to disable those alerts entirely. This creates a dangerous blind spot where real attacks can go unnoticed.

Furthermore, teams face the added challenge of being bombarded with alerts that don't convey enough context to help prioritize the true threats. Even when an alert is accurate and not a false positive — far less common than it should be — the SOC doesn't have an easy time understanding the blast radius, impact, etc.

Because WAFs are limited by their reliance on network traffic analysis, they lack visibility into the application itself. This results in a high number of false positives, creating an overwhelming number of alerts. WAFs provide very limited, high-level visibility into the behavior of applications in production, making it difficult to identify, understand and stop emerging threats.

In addition to the extremely limited ability of current tools to protect applications, organizations commonly have issues with staff and expertise — or siloed security and development teams. This makes it a struggle to communicate and collaborate, slowing the detection of and response to attacks on the application layer.

Consider that an estimated 26-billion threats target applications and APIs per month. Inevitably, alert fatigue sets in and security teams ignore or tune out potentially important notifications. This creates serious risk for missed attacks and delayed incident response.

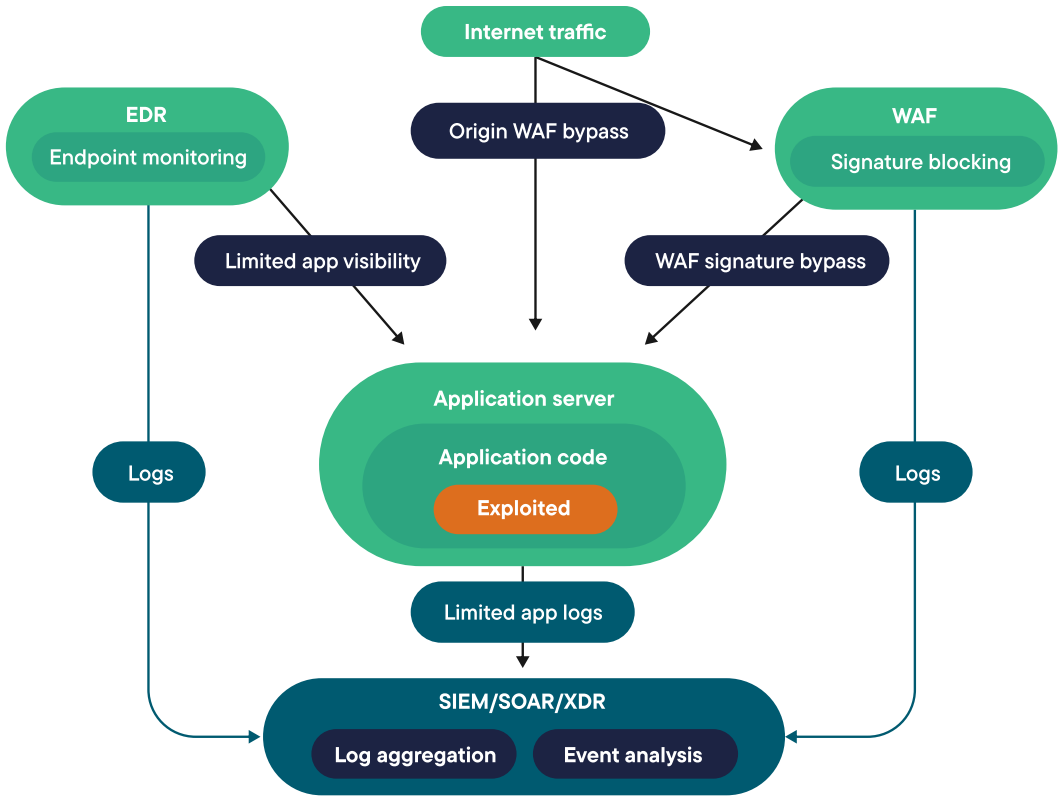
Applications and APIs are a part of the attack surface that's exposed to the internet, and ADR empowers SOC teams to identify vulnerabilities, detect threats and stop attacks that target this critical layer.

ADR allows SOC teams to look for code vulnerabilities in production, which helps them validate the development team's security work. Monitoring for both vulnerabilities and attacks in production can provide a full picture of an organization's Application Security (AppSec) threat position and how to manage it.

Bottom line: Sophisticated attackers use novel techniques to evade traditional detection methods, leaving organizations vulnerable to most application-layer attacks and zero days. Without the right people, processes and tools at the ready, it is extremely difficult to keep pace with the evolving threat landscape.

Without ADR, SOC teams lack visibility into applications and APIs and the threats targeting them, making it impossible to detect and understand app-layer intrusions until they make their way into other systems. Letting the threat actor get a foothold is not a desirable method of identifying attacks!

It's time to secure applications from within.



Understanding ADR

ADR complements WAFs by providing deeper application security from within. In order to see and stop modern application attacks in time — before the damage is done — security teams need to extend their reach beyond networks and endpoints and into the applications themselves. ADR eliminates application blindspots and protects applications and APIs by providing unparalleled visibility, accuracy and protection. Adding ADR provides otherwise missing visibility into application-level activity, allowing security teams to detect and block attacks before the attacker can create a bad outcome.

ADR provides the SOC with the needed visibility to see application attacks in real time, accelerating detection and response to anomalous activity within the application code itself. Blocking known and unknown vulnerabilities, including zero days, ADR empowers organizations to stop application threats sooner. This gives the SOC the context it needs to make effective blocking decisions. Continuous vulnerability assessment and seamless integration with Extended Detection and Response (XDR), plus Security Information and Event Management (SIEM) platforms, provide context-rich information about application-level threats. Securing applications from the inside means security teams can effectively detect and respond to attacks in the application layer through actionable, context-rich alerts (not just more noise).

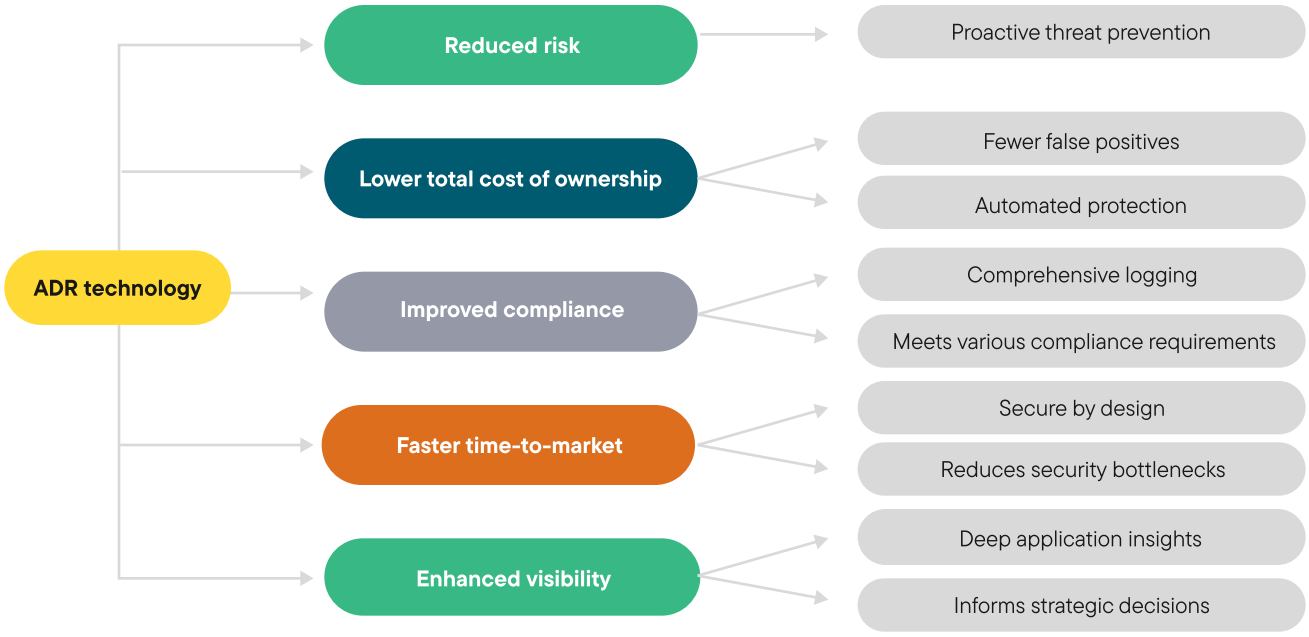
ADR can track data flows, identify vulnerable code execution and detect anomalous activity within the application logic. Incident responders get full execution context and comprehensive playbooks to contain and remediate application threats quickly. Developers and AppSec teams receive detailed execution path details down to the line of code from the specific targeted function, enabling them to fix vulnerabilities with less hassle.

The ADR agent secures applications from within by gathering security telemetry using various security instrumentation techniques, including code scanning, library scanning, application instrumentation, configuration file scanning and others. With the advantage of internal positioning inside the application layer, ADR has the context necessary to spot attacks on both known and unknown vulnerabilities, including zero-day attacks at the application layer that WAFs miss.

ADR detects attacks on production applications and blocks them in real time. It then generates alerts with supporting telemetry to drive fast and effective incident response. Detailed playbooks, application alerts and telemetry ensure that SOC teams are equipped with the data and expertise they need.

ADR gives SOC teams visibility into APIs and code in production, detecting anomalous behavior across the application stack by leveraging in-app agents that monitor security-relevant application behavior continuously while code is running.

Taking an “inside-out” approach, ADR can detect vulnerabilities in custom code and Open Source Software (OSS) code that only appears at runtime. ADR transmits threat and attack data to the SOC for incident response workflows, including sending alerts. It can also enrich alerts with data about the state of the attack and the relevant vulnerabilities it exploits. ADR data can become part of playbooks that drive incident response workflows.



The power of combining WAF and ADR for application security

Each individual application or API gets attacked an average of 45 times a month. These are real attacks, not the types of false positives that drive security teams crazy. This demonstrates the limitations of relying solely on WAFs.

Security activity per application, February 2025

By combining WAFs with ADR, organizations can achieve a layered defense:

- WAFs act as the first line of defense, blocking very obvious and common attacks and preventing malicious traffic from reaching the application.
- ADR provides deeper inspection as it can see how input values are being transformed by the application, where these values are being used, and whether they hit a vulnerable part of an application, thus providing protection from logic flaws, zero-day attacks and data breaches that might bypass a WAF.

This layered approach offers several benefits:

- **Comprehensive security:** Addresses a wider range of threats, including both known and unknown vulnerabilities.
- **Reduced risk:** Minimizes the attack surface and potential damage from breaches.
- **Improved efficiency:** SWAF can be configured to handle only common attacks and to let other traffic flow through without spending time on analysis. ADR would then focus on more complex threats, as it sees what happens from within the application and does not have to rely on probability analysis or guesswork like a WAF would.from logic flaws, zero-day attacks and data breaches that might bypass a WAF.

Benefits of ADR + WAFs: Two real-life examples

To get a sense of why it's beneficial to deploy ADR and a WAF, consider what happens in a deserialization attack and in an attack targeting the Log4j vulnerability.

Example one: Deserialization attacks

Deserialization attacks exploit a fundamental process within many applications. Applications often serialize data — converting objects into a format suitable for storage or transmission. Later, this data is deserialized back into objects for use by the application. Attackers can craft malicious payloads that, when deserialized, trigger unexpected or harmful actions within the application.

WAFs primarily analyze the content of incoming requests. They might spot an unusually structured data payload but often lack the context to understand how an application will process that data during deserialization. This means attacks that exploit application logic flaws in the deserialization process can slip through a WAF's defenses.

An attacker might craft a payload that, when deserialized, executes arbitrary code on the server (remote code execution). A WAF, analyzing the payload itself, might see nothing inherently malicious. It wouldn't understand that the deserialization process would lead to the dangerous execution of attacker-supplied code.

Understanding the nature of a serialized object is only possible within the application. Only there the content is revealed and can be analyzed based on its true nature and its usage by the application.

The instrumentation-based approach provides key benefits:

ADR operates within the application, monitoring its behavior at runtime. During deserialization, it observes how the data is processed — where it flows within the application and what actions it influences.

If the deserialized data triggers suspicious or unexpected actions, ADR recognizes these as potential attacks, blocking the request from causing further harm and alerting security teams.

ADR can neatly integrate with the application's error handling. If an attack is detected, it generates an exception, just like the application would for invalid input. This smoothly halts the abnormal usage, allowing the application to respond gracefully.

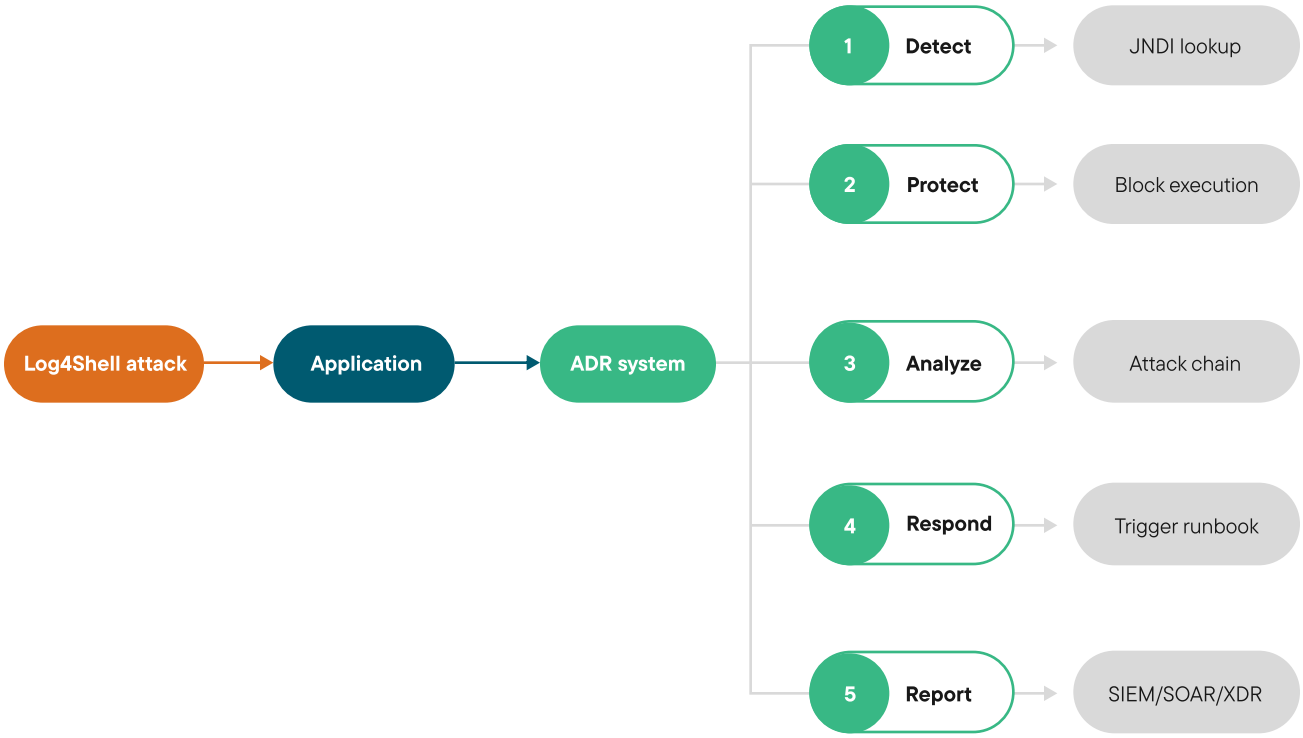
Example two: Log4Shell

The widespread Log4Shell vulnerability (2021) was a prime example. Attackers exploited a weakness in the popular logging library to inject malicious payloads that could take control of systems. Even as of December 2024, it still shows up in lists of Top 10 attacks[2]. Besides its massive impact — the scope of which hasn't yet been determined — it's probably the most widespread vulnerability ever, experts say. Officially identified as CVE-2021-442283[3], it carries a severity score of 10 out of 10 (CVSS v3.1) from the Common Vulnerability Scoring System (CVSS). Three years after Log4Shell was discovered, 12% of Java applications are still running vulnerable versions of the library.

Until specific signatures were released, WAFs had little chance of stopping Log4Shell attacks. In comparison, the data flow analysis capabilities of ADR would have allowed it to identify the abnormal Log4Shell payload behavior during execution, preventing exploitation even before the vulnerability became public knowledge.

ADR provides deep, real-time visibility and protection directly within the application layer. Until now, no other detection and response solution directly monitored and analyzed application behavior to detect real-time anomalies, attacks and vulnerabilities.

The technology enables organizations to trace attackers through all significant parts of an organization's IT infrastructure. Attackers choose to target applications and APIs that are connected to the organization's most valuable data. With ADR, analysts can track lateral movement from its point of origin — in applications and APIs — and stop the incursion before it becomes persistent.



Conclusion

While WAFs remain a valuable security tool, ADR offers a complementary and essential layer of protection by analyzing application behavior at runtime. By combining WAF and ADR, organizations can achieve a more comprehensive and future-proof application security posture, significantly reducing the risk of successful cyberattacks.

> "ADR's role in identifying and mitigating critical threats, along with providing essential application insights that security teams frequently lack, positions it as an essential tool. That's one of the reasons I see it as a growing cybersecurity category," said Katie Norton, Research Manager, DevSecOps and Software Supply Chain Security at IDC.

Contrast ADR is not just another security product. It's a game-changer, locking the door to keep out application and API attacks that can invisibly threaten your data and your business and opening up the power of continuous application-layer detection and response. With Contrast ADR, you empower your defenders with the observability and control they need in order to detect, respond to and block threats that target custom applications and APIs. It lets you strengthen application protection in a manner that's tightly integrated with existing security operations tools and workflows. Fill the gaps left by traditional detection and response tools: Safeguard your applications with Contrast ADR.

See Contrast ADR for yourself

[1] Verizon Data Breach Investigations Report (DBIR)
[2] Top 10 attacks
[3] CVE-2021-44228

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