WHITE PAPER

Contrast SECURITY

2024

The Case for Application Detection and Response (ADR)

ABOUT THE AUTHOR

Jeff Williams brings more than 20 years of security leadership experience as co-founder and Chief Technology Officer of Contrast Security. Previously, Jeff was co-founder and CEO of Aspect Security, a successful and innovative application security consulting company acquired by EY. Jeff is also a founder and major contributor to OWASP, where he served as Global Chairman for 10 years and created the OWASP Top 10 and several other popular open-source libraries and tools. Jeff serves as an application security advisor to the PCI Council, NIST, OASIS, CycloneDX, OWASP Foundation, Eclipse Foundation, and many companies and agencies. Jeff has a BA from Virginia, an MA from George Mason and a JD from Georgetown.

Why do our applications and APIs keep getting breached?

I've been doing Application Security (AppSec) for over 20 years, including defense systems, election systems, financial, utilities, airlines and more. I taught at the NSA's National Cryptologic School, wrote the OWASP Top Ten, and have performed manual penetration testing and code review on critical applications for hundreds of large organizations.

Believe me, I know how difficult it is to consistently write secure code. In the 22 years since I wrote the OWASP Top Ten, our progress has been glacial. The average application has dozens of serious vulnerabilities in both custom code and libraries. Some have many more. I'm deeply disappointed that we haven't been able to bend the curve in our software vulnerability epidemic.

Many of you may have bet your business on software, and that bet depends on the success of "shift left" efforts to write secure code. I spent a lot of my career with this same mindset. Unfortunately, it's become clear to me and should be clear to you that this is an extremely risky bet that even companies with the most advanced AppSec programs are losing.

Given this unfortunate situation, we absolutely must have excellent protection against web attacks. Unfortunately, since the late 1990s when web application firewalls (WAFs) emerged, there has been essentially no progress. It's hard to imagine that web application and application programming interface (API) defense has stagnated during decades of massive innovation in protocols, data structures and architecture, but here we are. Most organizations don't even have their WAFs blocking attacks. They just generate noisy alerts, ignored by operations.

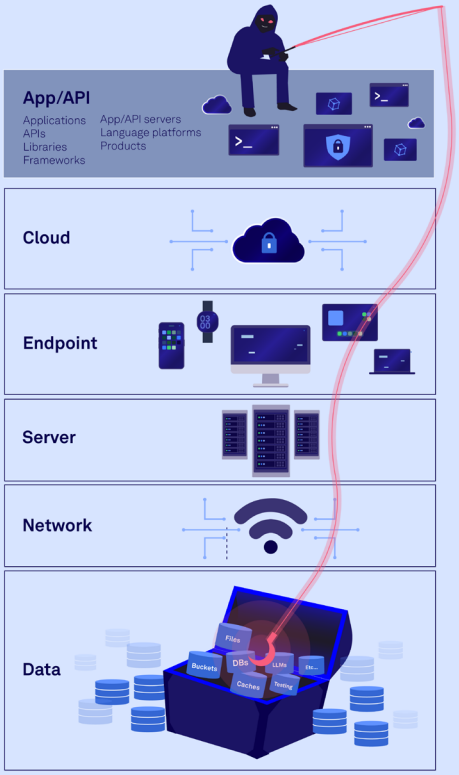
All the other layers of the stack have moved away from perimeter devices enforcing security on network traffic. Other areas of cybersecurity have moved to XDR, EDR, NDR, CDR, SDR, SIEM, SOAR and CNAPP — a "detection and response" approach using agents, real-time telemetry and advanced data analysis techniques to identify anomalous behavior. But AppSec in operations is in the Stone Age, with teams still clinging to WAFs like Signourney Weaver and her proximity detector in the movie "Alien."

In this white paper, I'm excited to discuss the Application Detection and Response (ADR) strategy and technology you need to keep your enterprise safe.

Why protect applications and APIs in production and operations?

The application layer is the lifeblood of modern business operations and security. We all trust software with everything important in our lives, including our finances, healthcare, government and social life. However, this crucial layer has become an increasingly attractive target for cybercriminals. According to the 2024 Verizon Data Breach Investigations Report[1], web application and API breaches are in the top 3 attack vectors, and the web application threat vector is in the top 2 for ransomware installs. Yet, the application layer remains woefully under-protected.

The enterprise application layer is sprawling, inscrutable, highly complex, highly vulnerable and controls all access to data



The complexity of securing the application layer cannot be overstated. Today's applications are dynamic, composed of dozens of repositories, hundreds of libraries, many APIs and microservices, serverless functions, and containers, deployed across multi-cloud environments — all of which, when assembled, introduce myriad vulnerabilities. According to Ponemon[2], enterprises have an expanding AppSec vulnerability backlog that already averages hundreds of thousands of application and API vulnerabilities. The sheer volume and velocity of changes in modern application development exacerbate the challenge, making it difficult for traditional security measures to keep pace.

The domain of AppSec has traditionally fallen to developers, engineers and dedicated AppSec individuals. Yet, other security oversight has gone to the security operations (SecOps) team. It is time to democratize security visibility for apps and APIs, allowing shared responsibility and improving the efficacy of the operations team.

This paper explores the pressing issue of deficient AppSec in production and operations. It delves into the limitations of existing solutions and highlights the need for a more comprehensive approach. We will introduce a groundbreaking methodology called Application Detection and Response (ADR), designed to provide continuous protection and real-time visibility into AppSec. ADR promises to bridge the gap left by traditional security measures, ensuring that the application layer is no longer a blindspot in an organization's cybersecurity strategy.

Current AppSec detection and response: Why the three traditional AppSec measures fall short

Today, enterprises have three options for detecting and responding to AppSec incidents in production. Two options, WAFs and secure software development, hail from the early 2000s, and neither was designed to address the full spectrum of threats that modern applications face in production. The third option — modern tools like extended detection and response (XDR) and cloud-native application protection platform (CNAPP) — simply doesn't cover the application layer. As a result, organizations lack visibility into application behavior in real time, leaving them blind to active threats and unable to respond effectively.

1. **WAFs** are useful for blocking simple attacks against known threats but suffer from significant limitations. They operate at the perimeter, inspecting incoming and outgoing traffic to identify malicious patterns. Because WAFs have no visibility into the internal workings of applications, they are unable to accurately detect attacks. Because of this weakness, WAFs have become notorious for both under-blocking and over-blocking. They also require extensive tuning and maintenance. While the flood of false positive alerts burdens security teams, the real danger is that many real attacks can slip through the WAF undetected.

2. **Traditional security operations tools** are very useful, but they lack visibility into AppSec. These tools include XDR; endpoint detection and response (EDR); network detection and response (NDR); cloud detection and response (CDR); security information and event management (SIEM); security orchestration, automation and response (SOAR); and CNAPP. The effectiveness of these tools is limited by the quality and comprehensiveness of the data they receive from sensors and log files. They use integrations and agents that gather logs and events from hosts, containers, network devices and cloud environments. But since none of their sensors has visibility into the security of applications and APIs, these tools simply cannot help with detection and response for AppSec incidents.

3. **Secure software development aims** to detect vulnerabilities and eliminate them during the development life cycle, before software gets into production. While this sounds good in theory, AST tools, such as static (SAST), dynamic (DAST) and open-source Software Composition Analysis (SCA) tools, have succeeded in identifying vulnerabilities but have not reduced or fixed them. In fact, the average number of vulnerabilities has increased from 24.2 to 43.6 in the 20 years since the OWASP Top Ten was first released. The mean time to respond/remediate (MTTR) vulnerabilities found with SAST tools is currently at 290 days[3].

Current AppSec detection and response: Why the three traditional AppSec measures fall short (cont.)

None of these methods is sufficient in isolation. Even when combined, they fail to provide the comprehensive protection needed for today's complex application environments.

In discussions with CISOs, AppSec teams and security operations centers (SOCs), a recurring theme emerges: Organizations are eager to learn more about the application and API security blindspot and express a strong desire for solutions that can fill it. CISOs express concern over the inability to see and respond to application-level threats, acknowledging the inadequacy of current tools. Quotes from industry leaders highlight the urgency of addressing this gap and the necessity for innovation in this space.

For instance, one CISO remarked,
"We have robust defenses for our networks and endpoints, but when it comes to our applications, we are essentially flying blind."

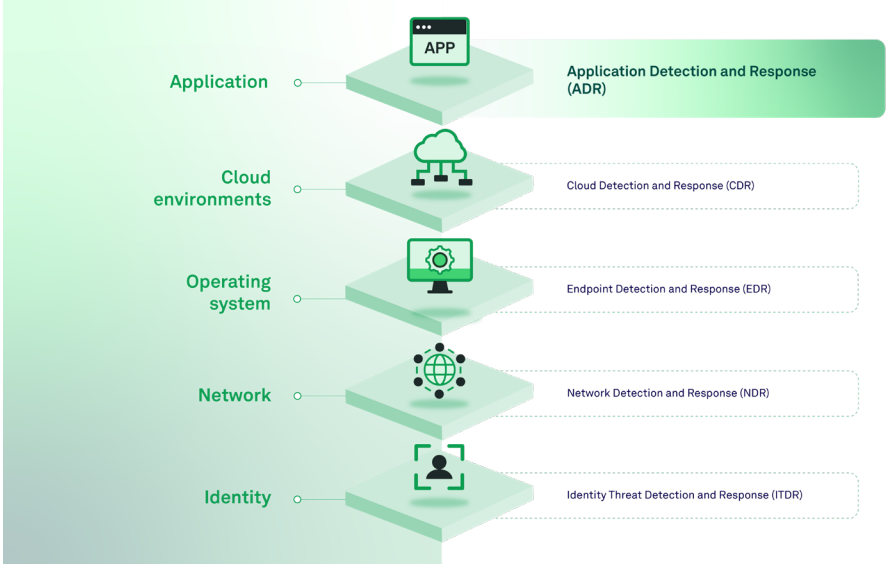
Another security leader noted,
"The increasing complexity of our application environment means that traditional security measures are no longer sufficient. We need deeper visibility and more effective response capabilities."

Introducing ADR

As the complexity and interconnectivity of modern applications continue to grow, so does the sophistication of cyber threats targeting them. Traditional security measures have proven inadequate in providing the necessary visibility and protection at the application layer. Fortunately, a new security technology has emerged called ADR.

ADR instruments applications and APIs to protect, detect and respond to threats in real time. ADR provides continuous visibility into the security behavior of the entire software stack, identifies anomalies that indicate security incidents, automatically takes action to mitigate these threats, and provides highly contextual feedback to operations and development teams.

ADR fills the critical AppSec gap left by other detection and response solutions by providing deep, real-time visibility and protection directly within the application layer. Security teams have added detection and response methods for other attack vectors: EDR for devices such as user laptops, CDR for threats to the cloud environment, identity threat detection and response (ITDR) for monitoring of identities, and so on. Those technologies then feed their telemetry into next-gen SIEM, XDR and CNAPP platforms. But, until now, none directly monitored and analyzed application behavior to detect anomalies and vulnerabilities in real time.



Filling this gap with ADR enables organizations to trace attackers through all the major parts of an organization's IT infrastructure. Attackers choose to target applications and APIs because they are directly 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. This capability enables SecOps team members to suppress lateral movement and decrease the dwell time of adversaries. Later in this paper, we'll discuss Contrast's approach to integrating ADR with XDR, SIEM and CNAPP platforms to give SecOps teams the proverbial "single pane of glass."

Use cases supported by ADR

1. **Protect applications/APIs —** Operations must deploy security measures that proactively prevent vulnerabilities within the app/API layer from being exploited. With ADR, teams can achieve this by embedding security measures directly within the running application, protecting the entire application stack, including custom code, frameworks, open-source libraries, app/API servers and runtime platforms.

    - Install into your standard application and API platforms
    - Install across a cluster with Kubernetes operator

2. **Detect applications/APIs —** Operations must be able to detect app/API threats quickly and accurately. By leveraging ADR, teams can use sophisticated algorithms to continuously monitor applications, detect anomalous behavior in real time and provide immediate alerts with extensive contextual information for quick triage of confirmed security incidents.

    - Flag anomalous application behavior and correlate with XDR, SIEM, CNAPP models
    - Instantly reference the security blueprint for the processing of any HTTP request
    - Highly accurate detection with minimal false positive alerts

3. **Respond to applications/APIs —** Operations teams must be prepared to respond promptly to identified threats by initiating predefined response actions. ADR enables teams to automatically block malicious traffic, isolate compromised components and notify security teams with detailed context, effectively containing and neutralizing security incidents to minimize impact and maintain application integrity.

    - Prevent exploitation of known vulnerabilities (Common Vulnerabilities and Exposures, or CVEs)
    - Prevent exploitation of zero-day vulnerabilities in both custom code and libraries

4. **Ensure applications/APIs observability and compliance —** It is impossible to manage the security of a frighteningly complex application layer without a detailed architecture of how it all works. ADR automatically generates detailed, real-time security blueprints of every application and API, including how they connect with each other. These blueprints help teams ensure compliance with regulatory requirements and enable effective security governance across the organization.

    - Continuous visibility into security activity across the enterprise application layer
    - Seamless integration with XDR, SIEM, SOAR and CNAPP for full-spectrum security

A tale of two scenarios: The unsafe deserialization incident

Let's consider a real-world situation that is occurring right now in many enterprises.

Imagine an enterprise web application that exchanges data with a Javascript user interface in the browser. The developer simply followed common coding patterns and "serialized" data objects in the browser into a stream of bytes that are "deserialized" back into objects in the web application. The developer didn't realize that they had inadvertently introduced a serious unsafe deserialization vulnerability. The company's traditional SAST and DAST tools failed to detect the vulnerability, and it made it into production. This is sadly a common occurrence for many organizations, resulting in the inclusion of "Unsafe Deserialization" in the OWASP Top Ten.

BEFORE ADR

An attacker happens on the deserialization vulnerability and crafts a malicious serialized object that includes a payload designed to execute arbitrary code on the server. When the serialized object arrives at the application, the code automatically deserializes it. As part of this deserialization process, the payload contained is executed, enabling the attacker to run arbitrary code on the server — a complete takeover of the server.

Unfortunately, the company's WAF didn't stop the attack from reaching the application. The WAF doesn't have visibility into the serialized object and therefore can't tell that the malicious payload is any different from normal traffic from a legitimate user.

In addition, the company's EDR, XDR, SIEM, SOAR and CNAPP solutions also missed the attack. These platforms are primarily focused on monitoring endpoints, networks and logs for known attack patterns and anomalies. The attack within the serialized object is invisible to these security platforms because it's buried in the serialized data and nothing is logged for this operation.

With arbitrary code execution on the server, the attacker begins to expand the incident. They use the initial foothold to explore the internal network, seeking additional vulnerabilities and high-value targets. The attacker leverages the compromised server to move laterally across the network — a process known as island hopping — compromising other internal hosts and accessing sensitive data.

A tale of two scenarios: The unsafe deserialization incident (cont.)

AFTER ADR

Now let's consider the same scenario, but with ADR deployed. The ADR platform continuously monitors the entire application stack in real time. During routine operations, ADR detects an unsafe deserialization vulnerability as it is being exploited. The system generates a detailed incident report containing:

- The complete HTTP request details, including the payload.
- A stack trace of the deserialization operation, captured directly from the running code.
- A contextual diagram providing the security context of the route being attacked.

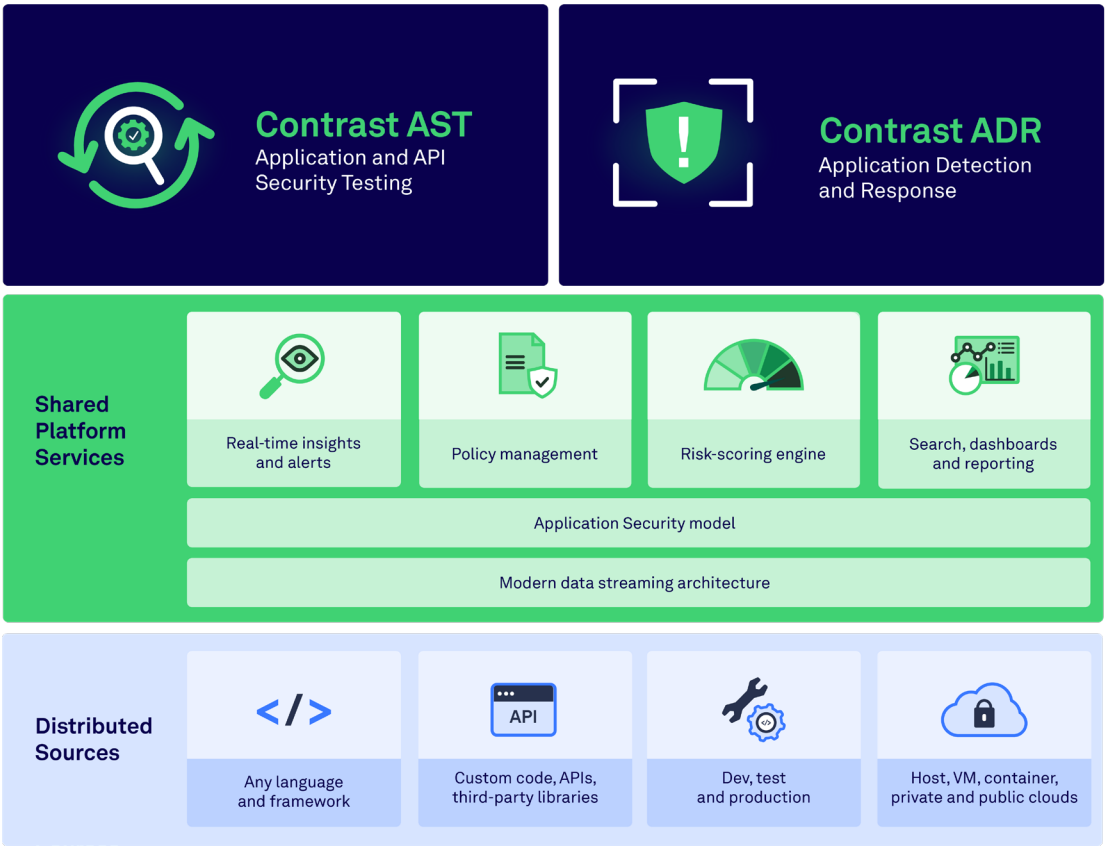
Upon reviewing the incident details, the security team decides to enable ADR's automatic response feature to block the attack. ADR's automatic response is configured to sandbox the deserialization process, preventing any operating system calls and other anomalous behaviors associated with the exploit.

With the automatic response enabled, ADR immediately intervenes when the attack is attempted again, blocking the malicious payload and preventing code execution. The security team verifies that the attack is successfully prevented in real time. The team still receives alerts from ADR, allowing them to remain informed of attempted exploits. And developers still receive details of the underlying code vulnerability, confident that ADR is protecting the application while the code is updated.

Had ADR been deployed from the start, the situation would have been proactively managed. ADR's deep visibility and continuous monitoring capabilities would have stopped the attacker before the attackers could gain a foothold and expand the incident. Even with the critical deserialization vulnerability present in the code, ADR ensures safety. It protects the entire application stack, spanning known and unknown vulnerabilities — whether the vulnerability lies in custom code, an open-source library, a framework or the application server.

The Contrast Runtime Security Platform: The technology behind ADR

The effectiveness of ADR hinges on a robust underlying technology that can seamlessly integrate with the development, operations and security processes. The Contrast Runtime Security Platform provides this foundation, offering a comprehensive solution for embedding security within the application runtime, ensuring real-time protection, detection and response capabilities.



As shown in this architecture diagram, the Contrast platform is designed from the ground up as an integrated approach to application and API security, rather than a mashup of unrelated technologies. It starts with fully distributed, lightweight security instrumentation that monitors AppSec behavior from within the running applications and APIs. This telemetry feeds our modern data streaming architecture, from which Contrast builds and maintains a sophisticated model of AppSec across an enterprise, also known as a Digital Security Twin (DST). This model enables highly accurate issues and incidents, contextual risk rating, real-time notifications, and much more.

The Contrast Runtime Security Platform: The technology behind ADR (cont.)

Real-time alerts and insights

- Contrast uses real-time data to assess the severity of security incidents, triggering immediate alerts with context and remediation guidance.
- Contrast constantly monitors applications across environments, analyzing changes and flagging policy violations.
- Contrast delivers critical security information directly to the right teams, through the tools they already use for seamless integration.

Risk-scoring engine

- Contrast's dynamic risk-scoring engine prioritizes security efforts by considering factors like business impact, threat landscape, security maturity and vulnerability details.
- The risk-scoring engine helps development teams focus on fixing high-risk vulnerabilities and empowers operations teams to respond quickly to incidents with in-depth code-level details.

AppSec model

- Contrast creates a DST of your enterprise application ecosystem. This model is a real-time, integrated view covering inventory, attack surface, vulnerabilities, threats, defenses, connections and more.
- Capable of handling hundreds or thousands of applications, the DST enables unparalleled analysis, precise risk prioritization and effective incident response within a single model.

Search, dashboarding and reporting

- Contrast provides rich dashboards and powerful analytics for a complete view of AppSec posture across the entire portfolio.
- Dashboards are provided and tailored to different roles (development, security, etc.) with role-based access control, plus the ability to query and analyze data for deeper insights.

Centralized policy management

- Contrast allows organizations to manage all aspects of AppSec in real time, from vulnerability assessment to compliance, and across their entire application portfolio.
- New security rules can be added instantly and customized across all applications, without the need for additional scans or redeployments.

Modern data-streaming architecture

- Contrast's distributed architecture efficiently ingests and analyzes large volumes of security data from various sources across all environments (development, QA, production, cloud, etc.).
- Real-time vulnerability and attack telemetry informs SecOps and developers, enabling rapid identification and response to security threats.

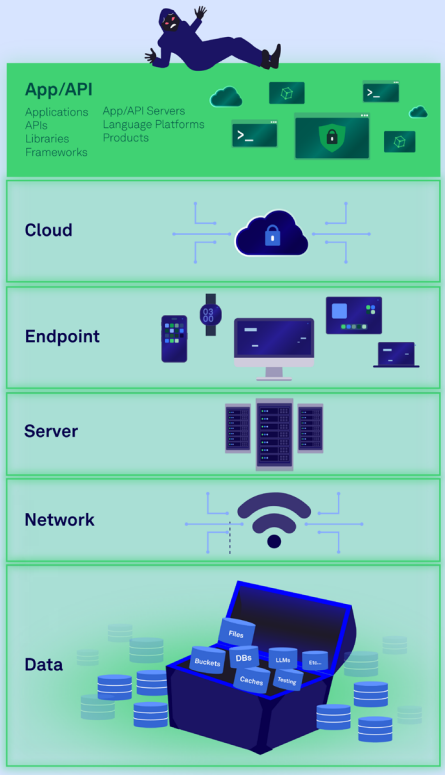
The Contrast Runtime Security Platform: The technology behind ADR (cont.)

The Contrast Runtime Security Platform is already in use in hundreds of thousands of critical applications and APIs in many of the world's largest companies. Our Runtime Security Platform monitors and protects trillions of dangerous function calls every day.

Contrast Runtime Security

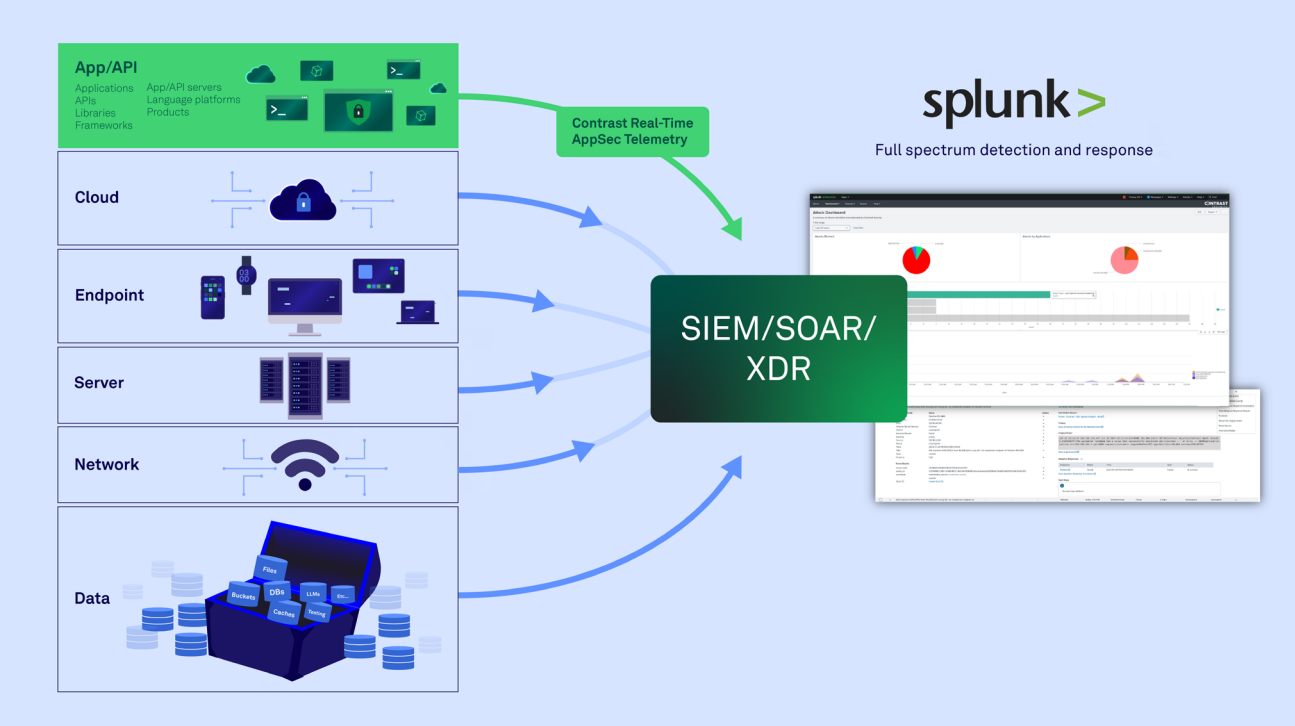
Monitors and protects your App/API layer from within to keep your data safe

Contrast automatically hardens software by adding missing security checks and efficiently preventing exploits



Operationalizing ADR to achieve full-spectrum detection and response

Let's imagine a financial services organization that has decided to enhance its AppSec operations by deploying Contrast ADR. Initially, their focus was on leveraging the platform's advanced instrumentation and real-time monitoring capabilities to detect and respond to threats within their application environment. As soon as Contrast was deployed, the security team started seeing an unprecedented level of detail and insights into the application's behavior and potential vulnerabilities.



Step 1: Integration with Splunk for event management

Impressed by the granularity and relevance of the data Contrast collects, the financial services organization's security team sought to integrate this valuable information into their existing SIEM system, Splunk. The team used syslog to stream events from Contrast into Splunk, adhering to the Common Information Model (CIM) for standardized event data.

Once the events were flowing into Splunk, the financial services organization's security analysts utilized the Contrast Splunk plugin to visualize the data. The integration allowed them to seamlessly incorporate Contrast's AppSec telemetry into their existing monitoring and triage processes. The rich, contextual information provided by Contrast enabled the team to identify and prioritize incidents more effectively. They could now view detailed attack patterns, understand the impact of vulnerabilities in real time and correlate AppSec events with other data sources within Splunk, streamlining their incident response workflow.

Operationalizing ADR to achieve full-spectrum detection and response (cont.)

Step 2: Enhancing CNAPP with application behavior

Recognizing the broader value of integrating AppSec insights with their CNAPP, the financial services organization also connected Contrast to Wiz. This integration allowed them to gain a holistic view of their infrastructure and its security posture, bridging the gap between infrastructure and AppSec.

With Contrast feeding detailed AppSec data into Wiz, the security team could drill down into each workload to see a comprehensive security architecture. They were able to visualize the interconnections between different components, understand the security implications of each connection and highlight incidents in the broader context of their cloud environment. This provided a deeper understanding of how vulnerabilities and threats at the application layer could impact the overall security of their infrastructure.

Step 3: Operationalizing ADR with enhanced workflows

The integration of Contrast with Splunk and Wiz enabled the financial services organization's security operations team to identify and respond to application and API security incidents without changing their existing workflows. The Splunk plugin facilitated easy access to Contrast data, making it a natural extension of their existing security operations workflows. In Wiz, the team could map out the full security architecture of their applications and infrastructure, identify critical vulnerabilities, and understand their potential impact on the organization.

This story underscores the transformative impact of deploying ADR within a robust security ecosystem. By embedding security within the application runtime and integrating it with comprehensive monitoring and management platforms, organizations can extend their protection across the entire application layer.

Business case for ADR: The strategic and financial benefits

The adoption of ADR offers a compelling business case for organizations looking to enhance their security posture, reduce costs and drive innovation.

Enhancing security posture — The landscape of cyber threats is evolving at an unprecedented pace, with applications and APIs becoming prime targets for sophisticated attacks. Traditional security measures often fail to provide the necessary visibility and protection at the application layer, leaving organizations vulnerable to breaches that can have devastating consequences. ADR offers a transformative solution, enabling real-time monitoring, detection and unmatched protection against threats within the application environment. By embedding security directly into the application runtime, ADR ensures that vulnerabilities are detected and mitigated before they can be exploited, significantly enhancing the organization's overall security posture.

Cost savings from automated vulnerability mitigation — The financial impact of a data breach can be staggering, encompassing direct costs such as fines and legal fees, as well as indirect costs like reputational damage and customer churn. Implementing ADR can lead to substantial cost savings by automating the detection and mitigation of vulnerabilities, reducing the likelihood of successful attacks. Automated vulnerability management minimizes the need for manual intervention, freeing up valuable resources and allowing security teams to focus on more strategic initiatives. Moreover, the early identification and remediation of vulnerabilities can prevent costly incidents, safeguarding the organization's bottom line.

CVE-2023-22527 Atlassian Confluence – template injection
CVE-2023-34040 Spring/Kafka – unsafe deserialization
CVE-2023-22965 Spring4Shell – malicious data binding
CVE-2021-44228 Log4Shell – JNDI injection RCE
CVE-2021-26084 Atlassian Confluence EL injection
CVE-2020-17530 Apache Struts2 – EL injection
CVE-2020-11651 Python Salt – authentication bypass
CVE-2020-11652 Python Salt – directory traversal
CVE-2020-9484 Apache Tomcat – unsafe deserialization
CVE-2019-2725 WebLogic – unsafe deserialization
CVE-2019-0230 Apache Struts2 – EL injection
CVE-2018-11776 Apache Struts2 – EL injection
CVE-2016-0792 Jenkins XStream – unsafe deserialization

These CVEs were prevented by Contrast long before they were disclosed publicly.

Improved development productivity and innovation — One of the key challenges in AppSec is balancing the need for robust protection with the demands of high-velocity software development. Traditional security tools often generate high volumes of false positives, leading to alert fatigue and slowing down development cycles. ADR addresses this issue by providing highly accurate, rich security insights that reduce false positives; enabling risk prioritization using production context; and streamlining the remediation process. By integrating security into the development pipeline, ADR supports DevSecOps practices, enabling development teams to innovate rapidly without compromising security.

Ensuring compliance with regulatory requirements — Regulatory compliance is a critical aspect of modern business operations, with stringent requirements imposed by the National Institute of Standards and Technology (NIST), the Payment Card Industry (PCI), the General Data Protection Regulation (GDPR) and others. ADR is critical to meet the Securities and Exchange Commission's (SEC's) new investigation and disclosure requirements. ADR helps organizations maintain compliance by providing continuous monitoring and detailed reporting of security activities.

In an era where the cost of a data breach can be catastrophic, the strategic implementation of ADR is not just a security imperative but a sound business decision. Organizations that invest in ADR position themselves to better navigate the complexities of the modern threat landscape, ensuring resilience, trust and long-term success.

Use cases supported by ADR

ADR represents a significant leap forward in securing the application layer, addressing the shortcomings of traditional security measures and filling critical gaps left by existing solutions.

ADR provides a transformative approach to AppSec by delivering real-time monitoring, deep behavioral analysis and automated response capabilities. It enhances protection against sophisticated cyber threats, ensuring vulnerabilities are detected and mitigated before they can be exploited. By integrating security directly into the application runtime, ADR offers continuous visibility and control over application behavior, significantly reducing the risk of successful attacks. Additionally, ADR's contextual insights and precise risk scoring enable security teams to prioritize remediation efforts effectively, improving overall incident response and resilience.

IMMEDIATE STEPS TO TAKE:

Evaluate your AppSec gap: Assess existing security defenses and evaluate whether they provide effective protection against application and API attacks.

Join me for a guided demo of ADR: Evaluate[4] Contrast ADR's ability to provide real-time visibility, continuous monitoring and automated response capabilities to applications and APIs in production.

Integrate with existing tools: Leverage integrations with XDR, SIEM and CNAPP platforms to enhance overall security visibility and incident response workflows.

Recognize value: Evaluate the benefits that ADR will bring to your organization.

Organizations must recognize the critical importance of securing their application layer and take proactive steps to integrate ADR into their cybersecurity framework. This strategic investment will yield long-term benefits, safeguarding the organization's digital assets, enhancing operational resilience and securing a competitive advantage in the marketplace. Now is the time to act, to ensure your organization is not only protected but also positioned for success in an ever-evolving digital world.

---

[1] Verizon 2024 Data Breach Investigations Report, p. 31

[2] The State of Vulnerability Management in DevSecOps

[3] Veracode State of Software Security (PDF)

[4] ADR demo

Contrast Security is the world's leading provider of security technology that enables software applications to protect themselves against cyberattacks, heralding the new era of self-protecting software. Contrast's patented deep-security instrumentation is the breakthrough technology that enables highly accurate assessment and always-on protection of an entire application portfolio, without disruptive scanning or expensive security experts. Only Contrast has sensors that work actively inside applications to uncover vulnerabilities, prevent data breaches and secure the entire enterprise from development, to operations, to production.

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

Contrast SECURITY

contrastsecurity.com