STATE-OF-THE-ART RASP IS APPSEC EXACTLY WHERE IT IS NEEDED—IN PRODUCTION RUNTIMES
EXECUTIVE OVERVIEW

One-quarter of all successful data breaches last year were the result of unprotected web application vulnerabilities.¹ This points to a lack of effective protection provided by traditional perimeter-based application security (AppSec) solutions. Instrumentation-based solutions like runtime application self-protection (RASP) deliver more AppSec value than perimeter solutions like web application firewalls (WAFs) because of continuous runtime observability from inside the application—namely, protecting vulnerabilities from the inside out.

This eBook examines these issues and how RASP delivers a state-of-the-art approach to application protection with accurate runtime observability, precision threat detection and blocking, seamless deployment, elastic scalability, and compliance with regulatory requirements. Readers gain the information needed to evaluate RASP solutions and how they supplement their perimeter defenses (WAF).

RESEARCH SHOWS THAT 71% OF APPLICATIONS IN PRODUCTION CONTAIN ONE OR MORE HIGH-SEVERITY FLAWS.²

TABLE OF CONTENTS

01 APPLICATIONS NEED BETTER PROTECTION—FROM THE INSIDE

02 APPSEC VISIBILITY DRIVES ACCURATE PROTECTION

03 ACCURATE DETECTION AND BLOCKING

04 EASE OF DEPLOYMENT

05 SCALABILITY

06 COMPLIANCE

07 SELF-PROTECTING APPLICATIONS IMPROVE SECURITY, EFFICIENCY
APPLICATIONS NEED BETTER PROTECTION—FROM THE INSIDE
APPLICATIONS NEED BETTER PROTECTION—FROM THE INSIDE

Traditional perimeter defenses lack visibility into running applications. Existing solutions like WAFs sit in front of an application separately, which means that they cannot see any context within the running code to determine if a potential threat should be blocked. Traditional perimeter defenses use signature-based methods to spot only known threats. This results in a high degree of inaccuracy.³ Perimeter solutions are also not fully application programming interface (API)-enabled and require significant manual support from staff. In many instances, security teams complain they need full-time personnel to manage and tweak perimeter rules.

In addition to the above, network protection has been moving closer and closer to applications—from firewalls to intrusion prevention systems (IPS) to web application firewalls (WAFs).⁴ The good news is an instrumentation-based solution with runtime application self-protection (RASP) automatically defends software code from the inside—as close to what is being protected as possible.

FAR TOO MANY COMPANIES HAVE LITTLE OR NO APPLICATION SECURITY AT ALL, OPTING INSTEAD TO DEPLOY NETWORK SECURITY CONTROLS (PERIMETER PROTECTION) AROUND APPLICATIONS.⁵

³ “Runtime Application Self-Protection (RASP), Investigation of the Effectiveness of a RASP Solution in Protecting Known Vulnerable Target Applications,” SANS Pen Testing, April 15, 2019
⁴ Ibid.
RASP delivers highly accurate observability of real-time events as they happen in the application runtime, rather than guessing at the perimeter with modeling predictions that may or may not be true. RASP instruments the application runtime to go beyond full traceability with complete call stack (custom and libraries), exact line of code, full runtime data trace, and originating HTTP request/response.

In doing so, RASP observes application runtime behavior and runtime data flows, without requiring the recalibration of statistical and other models the way a WAF would. In addition, the false positives and false negatives (missed threats) that frequently occur with perimeter solutions are eliminated with RASP precision runtime observability.

**CORE RASP COMPONENTS**

As a complement to or replacement for perimeter-based security solutions, RASP can provide state-of-the-art AppSec with capabilities that include:

- Automated blocking and alerting
- Reduced false positives
- Customized control of runtime events
- NIST/PCI compliance
- Instant AppSec scalability
- Continuous application-runtime observability
- Application data-flow visualizations
- Removal of disruptions and bottlenecks
- DevOps-native process fit
- Ease of deployment
APPSEC VISIBILITY DRIVES ACCURATE PROTECTION
APPSEC VISIBILITY DRIVES ACCURATE PROTECTION

Perimeter AppSec solutions lack continuous runtime observability capabilities to determine whether an application is actually vulnerable to exploitation. This is because the perimeter location of a traditional defense (WAF) simply cannot see the runtime contextual details to know what is happening inside the application. Without continuous runtime observability, runtime AppSec is reliant on partially blind defenses, which elevates the risk of a breach disruption and likelihood of a subsequent data loss and impacted reputation degradation.

RASP also reduces false positives due to accuracy and visibility inside the application runtime. Its position inside the runtime allows it to directly see, identify, and ignore non-damaging attacker probes while focusing solely on blocking protection against threats that are directly observed to be in a position to exploit an actual application vulnerability.

RASP delivers continuous, always-on security instrumentation that is a part of the application itself. This state-of-the-art, autonomous approach to AppSec ensures embedded runtime security and application runtime observability. In fact, when it comes to unknown threats and zero-day attacks, application runtime visibility is the only effective protection as signature-based perimeter defenses often miss them (viz., false negatives).
RASP ALLOWS AN APPLICATION TO RUN SECURITY CHECKS CONTINUOUSLY AND RESPOND TO LIVE ATTACKS BY TERMINATING THE BAD ACTOR’S SESSION AND ALERTING THE INFOSEC TEAM OF THE ATTACK.6

ACCURATE DETECTION AND BLOCKING
ACCURATE DETECTION AND BLOCKING

Over half of cybersecurity professionals indicate their organization is at moderate or extreme risk due to staffing shortages, and AppSec is an area where the gaps are the most glaring.\(^7\) Perimeter solutions create noise (i.e., a large number of false positives), which increases remediation workflows. Overflowing alert queues cause security fatigue that can prompt teams to turn off blocking mode on WAFs due to excessive noise, for they are unable to differentiate a real attack (exploit) from an attempted attack (probe).

RASP makes applications self-protecting so security teams can keep AppSec teams focused on the attacks that really matter. Similarly, development teams can focus on new code delivery rather than fixing old code. This is exceptionally valuable due to the fact that vulnerabilities discovered after the release of a product are about 100 times more expensive to resolve than those found in the requirement and design phases.\(^8\) However, if embedded and continuous protection are included in applications in production, then both security and DevOps teams can conserve limited team resources.

ONLY DETECTING REAL ATTACKS OR ATTEMPTED EXPLOITS ALLOWS A RASP TO ELIMINATE THE ALERT FATIGUE THAT COMES WITH A WAF, WHICH RESULTS IN OPERATIONAL INEFFECTIVENESS AND HIGHER RISK.\(^9\)

---

04 EASE OF DEPLOYMENT
EASE OF DEPLOYMENT

Perimeter-based solutions are often slow and difficult to deploy and maintain. They require coordination with network teams to ensure they see the right traffic. Perimeter defenses also require setup of static rules that need constant redefinition, which greatly increases operating expenses (OpEx). Because RASP is instrumented self-protection, it simplifies deployment and maintenance workflows and reduces total cost of solution ownership for security teams. Unlike perimeter-based approaches, a RASP solution:

- Offers single integration cost per application
- Provides consistent “no-surprise” deployments and runtime remediation-verification for developers
- Defers patching and remediation activities in favor of production defenses
- Eliminates redeployments and delays after remediation
- Reduces expert staff dependencies and associated costs (no tuning, no configuration, no pen testing, no perimeter solution maintenance)

OVER HALF OF CYBERSECURITY PROFESSIONALS INDICATE THEIR ORGANIZATION IS AT MODERATE OR EXTREME RISK DUE TO STAFF SHORTAGES, AND APPSEC IS AN AREA WHERE THE GAPS ARE THE MOST GLARING. ¹⁰

SCALABILITY
SCALABILITY

State-of-the-art AppSec must scale seamlessly with an application and keep pace with the elastic nature of digital transformation. However, traditional perimeter solutions (WAFs) need to be expertly tuned with each new code deployment. These static defenses are separate from the application and not part of the runtime. Thus, they also need to be redeployed if a protected application moves to a new server or changes infrastructure (such as expanding into a new cloud deployment). All of these requirements obstruct DevOps workflows and inhibit business growth while incurring significant OpEx.

Unlike perimeter defenses, runtime embedded self-protection scales seamlessly as part of an application. For example, if an application creates copies of itself on multiple server instances to serve a distributed user base, RASP defenses will seamlessly scale within every instance of an application in complete lockstep—all without configuration or tuning, no matter where the application is deployed. Additionally, if placed on virtual or cloud servers, RASP benefits from the added CPU and memory resources right alongside the application.
06

COMPLIANCE
Industry standards and regulatory legislation for cybersecurity and privacy are becoming increasingly strict and specific in their requirements. These recent changes challenge perimeter defenses, which have always struggled with compliancy requirements. The latest releases from the National Institute of Standards and Technology (NIST SP 800-53) and Payment Card Industry Software Security Standards (PCI SSS 9.1, 10.2, and 11.2) require state-of-the-art security instrumentation for “application self-protection at runtime” to reduce the susceptibility of software to attacks by monitoring inputs and blocking those inputs that could allow attacks.\(^{11}\)

**NIST’S FINAL DRAFT OF SP 800-53, REVISION 5 INCLUDES A NEW SAFEGUARD STANDARD SI-7 (17), WHICH REQUIRES STATE-OF-THE-ART RUNTIME APPLICATION SELF-PROTECTION (RASP).**\(^{12}\)


SELF-PROTECTING APPLICATIONS IMPROVE SECURITY, EFFICIENCY
SELF-PROTECTING APPLICATIONS IMPROVE SECURITY, EFFICIENCY

Modern applications must become self-protecting in order to effectively defend against the constant onslaught of sophisticated attacks. RASP is specifically designed to protect against both known and unknown threats as a result of being embedded within the application runtime itself. RASP instrumentation delivers visibility and observability across the full context of running application runtime events (including code, libraries, and APIs) to defend vulnerabilities with superior accuracy.

And because RASP is inseparable from the actual application, deployment and scaling become effortless. Security teams can reduce dependency on manual workflows and focus on true strategic and business-critical tasks. Finally, in addition to reducing application breach risk, RASP provides immediate compliance with mainstream industry standards like NIST and PCI SSS.

THE GLOBAL RASP MARKET IS EXPECTED TO GROW AT A COMPOUND ANNUAL GROWTH RATE (CAGR) OF 34.5% FROM 2019 TO 2027—STARTING FROM $430.8 MILLION USD IN 2018.13

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.