The Continuous Application Security Handbook

Unify Security Strategy Across Development & Operations

FALL 2016

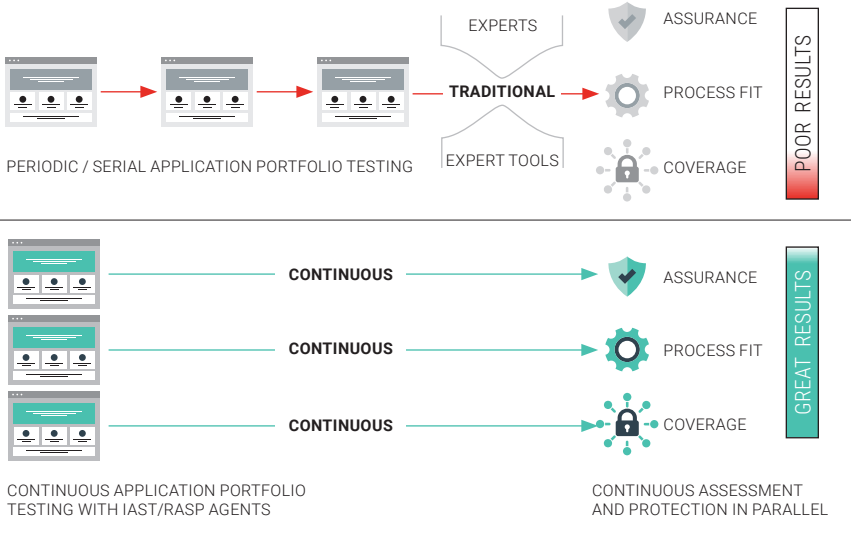
We reject the old paradigm of **periodic and serial** scanning, hacking, and patching, which has proven expensive and ineffective. Instead, Continuous Application Security (CAS) relies on security instrumentation in every application. This instrumentation provides security visibility, assessment, and protection **in real time and in parallel** across the entire application portfolio. CAS is a unified program covering the entire software lifecycle, including both development and production, designed to create a clear line of sight from the threat to strong defenses, and ultimately to assurance.

INTRODUCTION

We recognize that software is the key to protecting both our enterprise and the people who entrust their information to us. We understand the need for vigilance in understanding the threats we face and our own weaknesses. And we know that our defenses must be continuously monitored to ensure their ongoing effectiveness.

01	We choose to protect our entire application portfolio **over focusing on a small number of critical applications**
02	We choose to standardize and simplify our enterprise application security architecture **over reinventing security on every project**
03	We choose to define, verify, and enforce security with automated sensors **over papering security policies, requirements, standards, and guidelines**
04	We choose to empower applications to protect themselves from attacks **over relying on external tools and technologies that lack the proper context**
05	We choose to treat security issues like any other software problem **over establishing security-specific systems and processes**
06	We choose to make security decisions informed by portfolio-wide visibility **over taking blind risks without up-to-date information**

Figure 1. Continuous Application Security Removes Bottlenecks



Contents

EXECUTIVE SUMMARY

Today, every organization has become a software company. The increasing dependence on automation demands that software survive and thrive despite an increasingly hostile environment. Insecure code has become the leading security risk and, increasingly, the leading business risk as well. It's irresponsible at every level to ignore this risk while doubling-down on anti-virus solutions and firewalls — neither of which protects applications.

Even well-established application security programs often can't operate at the speed and scale required because they rely on activities only experts can perform and tools that only experts can operate. This approach disrupts the software lifecycle and is incompatible with modern high-speed software development. Including the technology and human cost, the annual ***per-application*** cost for these programs can range from $50,000 to $100,000 per year.

A continuous application security (CAS) program empowers ordinary developers to reliably build and operate secure applications and APIs by transforming paper-based security policy and guidance into "security as code" through instrumentation-based security enforcement. CAS enables development, security, and operations to work together effectively, at the pace of modern software development and at global enterprise scale. CAS builds an application security program in three layers:

Layer 1: Development and operations get ***fully automated*** security support

Layer 2: Security experts deliver ***security*** as code

Layer 3: Management makes informed decisions with detailed ***security analytics***

Contrast Enterprise enables these three steps with a highly scalable architecture that includes distributed agents and a centralized dashboard. Contrast agents empower — through instrumentation — every application in the portfolio to analyze, enforce, and communicate about application security. Every Contrast agent is controlled centrally from the Contrast TeamServer dashboard, which provides full visibility and policy control across the entire portfolio.

Organizations practicing CAS quickly determine how a new risk affects them, design a defense strategy, and measure their progress to 100% coverage. This ***risk agility*** allows teams to respond quickly and confidently in a rapidly changing threat environment.

This handbook explores how to assemble an application security program by implementing eight functions within an enterprise. All of these activities can start small and prove their cost-effectiveness quickly.

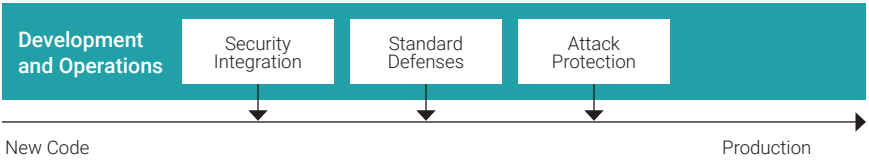
> "Remember, the goal is to be the attacker's nightmare, not be the developer's nightmare or business people's nightmare"
>
> **Gunnar Peterson**
> *ArcTec Group Managing Principal*

GETTING STARTED WITH CAS

In contrast to most security programs, CAS is designed to **accelerate** the software lifecycle by enabling software teams to deliver and operate secure applications **themselves**. The primary responsibility for security is squarely placed on software development and operations organizations. They will build standard security controls, ensure their code passes automated security verification, and ensure their applications are protected against attack. Here's how it works:

Level 1: Development and operations get **fully automated** security support

Development and operations teams are cleared to build and deploy software without any extra security steps or "gates" as long as it passes fully automated security verification.



Development teams rely on Contrast Enterprise with IAST (Interactive Application Security Testing) to instantly detect vulnerabilities in the code as it is created, without any external support. Standard defenses can be used to build secure applications and remediate vulnerabilities. Development can use Contrast Enterprise with RASP (Runtime Application Self-Protection) in place to ensure that attacks are properly detected and blocked.

Level 2: Security experts deliver **security as code**

Application security focuses on creating and improving the organization's application security architecture, and accelerating secure software development and operations based on the threat from attackers and internal research efforts.

Application security experts can translate their research into new sensors and deploy them into the development process through Contrast Enterprise. These sensors can detect new vulnerabilities and block new attacks across the entire application portfolio in real time. Contrast Enterprise also gathers inventory details automatically and collects considerable threat intelligence based on attacks.

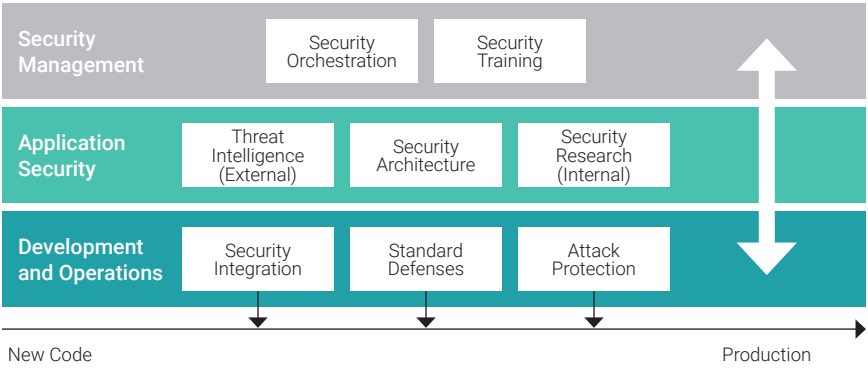
Level 3: Management makes informed decisions with **detailed security analytics**

Management ensures that all the application security activities are producing their expected results efficiently and properly prioritizing initiatives. Contrast Enterprise informs management with real-time visibility across the entire portfolio.

With this approach, security experts focus on threat intelligence and security research. This work becomes part of the security architecture and is implemented by development and assessed continuously in perpetuity.

Continuous Application Security

Here's the whole picture: There are only eight activities required to create a Continuous Application Security Program (see CAS section, starting on page 9), each of which is easy to implement and measure. Useful results are generated in a matter of weeks with minimal staff.



> "[We see] huge tensions where the software development guys are tasked with glittering features pretty quickly and the security group is designed to slow that down so that the output is more secured, and they wind up thinking both are enemies rather than allies in the whole process."
>
> **Wayne Jackson**
> *Sonatype CEO*

WHAT'S YOUR ANNUAL PER-APPLICATION APPLICATION SECURITY COST?

The goal with Continuous Application Security is to dramatically improve the cost-effectiveness of application security efforts by increasing security and lowering costs. Below is a worksheet filled out with a traditional static analysis service in mind. The numbers may be dramatically different if a manual process or an instrumentation-based approach is used. But understanding these costs is critical to improving results.

Table 1. Cost of Traditional Approach

Cost Area	Description	Cost – Static (SAST) Example
License Cost	Typical annual licenses for an application security tool can range from $5-10K on an annual basis. This cost is $0 if relying on a manual pen test and/or manual code review.	Static SAST **$10,000**
Analysis	Actually assessing an application typically takes 2-4 weeks for a manual review; 1 week for an automated scan.	Service **$0**
Triage	Typical results from automated tools have a significant number of false alarms that must be eliminated using expert knowledge. Plan on several days per assessment; zero for manual reviews.	50 security hours **$10,000**
Reporting	Every vulnerability needs to get risk rated, written up, tracked, reported, and closed. Security dashboards need to be created from all the data. Figure one day per assessment.	25 risk hours **$5,000**
Remediation	Full cost to remediate and deploy fixes. A typical application has 22 vulnerabilities at 20 hours each at $100/hr totaling roughly $44,000.	440 dev hours **$44,000**
Retest	The retest verifies that issues identified have been fixed appropriately. Typically the retest costs about 25% of original assessment.	Included **$0**
Management	If running a scanning program, several headcount will be needed to manage the schedule, contracts, and infrastructure required.	20 manager hours **$4,000**
	TOTAL	**$73,000**

Notice that the bulk of the cost is in the "Remediation" stage. Even if all the other costs go to zero, Remediation is still extremely expensive. CAS focuses on minimizing all the costs listed above by enabling vulnerability discovery and remediation as early as possible in the software development lifecycle — often before code is even checked in, where the cost-to-fix is almost zero.

The 8 Continuous Application Security Activities

There are only eight activities required to create a Continuous Application Security Program. Each of these activities is outlined on the following pages.

ACTIVITY 01: CONTINUOUS THREAT INTELLIGENCE

WHY IT IS CRITICAL

Without continuously monitoring the attack surface, who is attacking applications, and what techniques they are using, companies will get blindsided by attacks and waste money on the wrong priorities.

METRICS

- **Are new threats, novel vulnerabilities, or new defense strategies identified by the team or the news?**
- **What is the time between vulnerability announcement and portfolio-wide protection? Is it getting longer or shorter?**

Application "Threat Intelligence" ensures the organization is continuously aware of application attackers and vulnerabilities, using both external sources of information and real data about what the organization is actually experiencing. Organizations should also be aware of their own attack surface.

Understanding Attackers:
Organizations also need intelligence about attackers – who is attacking and what techniques they are using. In CAS, organizations use instrumentation to continuously monitor actual attacks on their own systems. In addition, they stay abreast of industry sources, including threat reports, conferences, papers, blog posts, and vulnerability disclosures from organizations like OWASP, FS-ISAC, FIRST, SANS, WASC, and Verizon.

Understanding Novel Vulnerabilities: Most organizations are blissfully unaware of new classes of application security vulnerabilities. For example, it took many years for Cross-Site Request Forgery (CSRF) to be recognized as a real problem. This pattern has repeated for Clickjacking, XML External Entities, Expression Language Injection, Deserialization, library vulnerabilities, and so many other problems. In CAS, organizations monitor industry sources and use IAST and RASP to determine their actual vulnerability and deploy protections.

Understanding Attack Surface:
The application security attack surface is constantly changing. New applications, new third-party products, new code, new libraries, new frameworks, and new environments make attaining a basic understanding of the application portfolio extremely challenging. In CAS, security instrumentation reports inventory, detailed "bill of materials," and configuration information to a centralized location. This includes so-called "internal" and "external" applications, as well as code in development, integration, test, staging, and production.

When new attackers, vulnerabilities, and attack surfaces emerge, Threat Intelligence should investigate and possibly deploy new detection-oriented sensors to evaluate the relevance of the new problem. Security Research may be involved in deeper exploration. In some cases, an immediate response, further inquiry, and possibly a public response may be required. Threat Intelligence coordinates these efforts to protect the enterprise.

HOW CONTRAST PROVIDES CONTINUOUS THREAT INTELLIGENCE

Contrast Enterprise with RASP uses instrumentation to identify and block attacks with extreme accuracy. This data is a goldmine of threat intelligence. Organizations learn where attackers are coming from and what types of attacks they are trying. In addition, Contrast Enterprise shows exactly where in the codebase attackers are targeting, down to the specific line of code.

"There is no teacher but the enemy. No one but the enemy will tell you what the enemy is going to do. No one but the enemy will ever teach you how to destroy and conquer. Only the enemy shows you where you are weak. Only the enemy tells you where he is strong."

Orson Scott Card
Ender's Game

ACTIVITY 02: CONTINUOUS SECURITY ARCHITECTURE

WHY IT IS CRITICAL

Without a clear definition of what security actually means and a structure to organize security priorities, security cannot be measured and will never be achieved.

METRICS

- **Can you easily answer any "how do we protect against X" questions?**
- **Do you have a line-of-sight from every defense to the business reason for that defense?**

Application "Security Architecture" manages a unified set of primary and secondary security defense strategies. Without a structure to organize application security priorities, organizations will never be able to improve. This activity unifies security practices known as "business threat modeling," "application threat modeling," "security policy," "security requirements," "security test cases," and "security guidance."

Application Security Architecture is the art of organizing a structured set of defenses that effectively covers the threats to applications. Each defense should include the high-level business concern, attack techniques, defense strategy, defense implementation, and verification technique.

There are multiple ways to organize an application Security Architecture. The simplest approach is a list that captures each high-level business concern, attack techniques, defense strategy, defense implementation, and verification technique. To ensure coverage, it may be helpful to organize these defenses and all their details into categories or into a full matrix. A more sophisticated, long-term approach is to build a hierarchical security story, organized by business threat, that captures the complexity of the company's application Security Architecture.

Security Architecture can exist at different levels of maturity. Early on, only the most critical risks are addressed. At higher levels of architectural maturity, multiple layers of strong defenses will ensure that vulnerabilities are prevented, attacks are blocked, and incident response is prepared.

Level 1: No defense strategy

Level 2: Negative defense strategy – block dangerous behavior

Level 3: Positive defense strategy – allow only good behavior

Level 4: Verified defenses – defense usage is verified with automated sensors

Level 5: Defense in depth – primary and secondary defenses

Level 6: Automated defenses – built into frameworks, impossible to avoid

Making Security Architecture continuous means translating it into something that is useful at the speed of software development. In CAS, organizations use Contrast Enterprise to turn architectural decisions into automated sensors that operate during both development and operations. These sensors provide immediate feedback whenever an application shows any behavior that deviates from the expected behavior. These sensors turn a complex security issue into something that development and operations organizations can manage as part of their normal development process.

Security Architecture is an ongoing activity that is never complete. Over time, defenses must evolve in response to new risks, enhanced security, cost savings, and simplicity. As Threat Intelligence and Security Research continuously identify ways to improve the Security Architecture, Security Architecture must respond by organizing, standardizing, and simplifying the architecture as it evolves.

HOW CONTRAST SUPPORTS CONTINUOUS SECURITY ARCHITECTURE

Contrast Enterprise makes Security Architecture enforceable by instrumenting an application portfolio with sensors that automatically detect vulnerabilities and block attacks. Contrast Enterprise also builds a unified application inventory with details of every application and generates live architecture diagrams showing actual components and connections directly from the running applications.

"The traditional 'hack your way secure' approach doesn't work and engenders a false sense of security. Security investment must be made strategically over time, not as a knee-jerk response to the latest threat."

John Pavone
Aspect Security CEO

ACTIVITY 03: CONTINUOUS SECURITY RESEARCH

WHY IT IS CRITICAL

If defenses aren't tested like an adversary, it will never be known whether they are strong enough to withstand real attacks.

METRICS

- **Are researchers finding novel security issues and improvements?**
- **Are new sensors being created and added to Security Architecture?**

Improvements in security emerge from an evolutionary process where both the expected security architecture and the current implementation are continuously challenged and strengthened. In CAS, application "Security Research" actively challenges the organization's security in order to accelerate this evolution.

Security Research actively searches for novel risks and turns them into "security as code" – automated sensors that continuously monitor and protect against these risks. Security Research focuses on expanding application coverage, code coverage, and vulnerability coverage across the entire software supply chain, including third-party software and products.

The goal is to seek out gaps, weaknesses, simplifications, and optimizations and make improvements. But in CAS, Security Research isn't for managing "known" vulnerabilities. Those are already part of the Security Architecture, having been instrumented with sensors. Security Research uses techniques like the following to find "novel" security issues:

- *Penetration testing*
- *Red teams*
- *Security instrumentation*
- *Fuzzing*
- *Static analysis*
- *Vulnerability scanning*
- *Policy analysis*
- *Bug bounty programs*

To ensure discoveries and advances aren't lost, they are turned into code by creating new Contrast Enterprise with IAST and RASP rules or "sensors." They become part of the organization's security immune system, protecting against both vulnerabilities and attacks.

Some organizations may enhance their Security Research efforts by leveraging external resources, like consultants and "bug bounty" programs. In either case, the goal isn't to just find individual vulnerabilities, but to strengthen the organization's immune system to defeat all vulnerabilities of that type.

HOW CONTRAST SUPPORTS CONTINUOUS SECURITY RESEARCH

Contrast Enterprise helps security researchers quickly gather data from their entire application portfolio in both development and production. Researchers can deploy custom security sensors to test assertions and find potential weaknesses in applications and services.

"Security researchers are just so far ahead of what basic organizations can do. They're finding really exotic flaws, but if the company is not even doing basic patching, then it just doesn't matter. So there's a little frustration that researchers are still stuck doing very remedial type of work."

Nick Galbreath
Founder and CTO of Signal Sciences Corporation

ACTIVITY 04: CONTINUOUS SECURITY INTEGRATION

WHY IT IS CRITICAL

Only with continuous monitoring can vulnerabilities across the portfolio get fixed early, when they are significantly cheaper to fix.

METRICS

- **Are all of the applications in the portfolio continuously analyzed for vulnerabilities in code, libraries, architecture, and configuration?**
- **Are vulnerabilities discovered and fixed as a part of normal software development, and integrated with normal development tools without security expert involvement?**
- **Is the number of vulnerabilities being introduced more or less than the number of vulnerabilities being remediated each month?**

Application "Security Integration" ensures that software development projects are using security sensors and standard defenses to develop, deploy, and operate secure applications and APIs.

In CAS, there is no separate security step. Contrast Enterprise's vulnerability analysis is done automatically by instrumenting the running application during development, integration, or testing. The vulnerability analysis happens "in the background," and produces immediate notification of any discrepancies between the organization's security architecture and the actual implementation. When a new vulnerability is identified, Contrast automatically notifies developers using the tools they use to do their normal work. This could include a variety of channels:

- *Alert or notification via Slack, HipChat, or other messaging (ChatOps)*
- *Email notification and status messages*
- *Bug report filed with JIRA, TFS, and other platforms*
- *Live, always up-to-date application security dashboard*
- *Other dashboards via REST API*

Developers receive a detailed report on the vulnerability, exact line of code, full HTTP request, and detailed remediation guidance. With this type of instant notification, most vulnerabilities are fixed as part of the normal development process, before code is even checked in. This is a natural way to work for developers and avoids over 80% of the cost of application security compared to a traditional late-in-the-lifecycle approach.

With the explosion of components, frameworks, and other libraries, gaining assurance in the software supply chain has never been more important. The challenge is ensuring that all third-party software, and how it is used, is consistent with the expected security architecture. In CAS, all third-party code is assessed continuously using Contrast Enterprise with IAST along with the rest of the application. Contrast identifies both publicly known and previously undiscovered vulnerabilities in third-party code.

HOW CONTRAST SUPPORTS CONTINUOUS SECURITY INTEGRATION

Contrast Enterprise continuously assesses the security of the entire application, including custom code, libraries, frameworks, server, and runtime. Contrast's instrumentation-based approach makes assessment accurate, scalable, and easy.

"If you have code that's important enough to deploy, you have code that's important enough to instrument."

Gene Kim
Author, Researcher, DevOps Pioneer, and Founder of Tripwire

ACTIVITY 05: CONTINUOUS STANDARD DEFENSES

WHY IT IS CRITICAL

Complexity is the enemy of security, and establishing strong standard defenses is the best way to reduce the mind-bending complexity of application security.

METRICS

- **Is there an active project to build and maintain a set of standard enterprise security defenses?**
- **Do the standard defenses cover all the major security threats to the enterprise?**
- **Are the standard defenses thoroughly security tested and easy to use?**
- **What percentage of the application portfolio uses all the standard defenses?**

"Standard Defenses" means that defense mechanisms like authentication, session management, access control, encryption, input validation, output escaping, error handling, and logging are correctly implemented, easy to use, resilient against attack, and kept up-to-date. This activity provides a toolbox of strong defenses that accelerates software development and reduces the likelihood of introducing vulnerabilities.

Most organizations have recognized that they should not write their own encryption. Instead, vetted implementations are used that have been scrutinized and tested by experts. This same principle applies to all application security defenses – including controls as seemingly simple as input validation, encoding, and logging – as they are critically important to stopping attacks and can be quite difficult to get correct.

In CAS, organizations are encouraged to establish a complete set of trustworthy and easy to use security defenses. These defenses might be offered as shared libraries, web services, products, or other alternatives.

By centralizing these controls and externalizing them from the application, they can be tested, managed, and maintained at the high level of quality that they require.

Defenses must:

- *Be correctly implemented*
- *Be resistant to bypass or tampering*
- *Be easy for developers to find, configure, and invoke properly*
- *Be easy for end users to configure and use properly*
- *Come with sensors that verify their correct use*

The Standard Defenses should be built and managed as a normal software project, using the same people, infrastructure, and resources as other software projects. Security Research should perform careful manual testing and code review, as this is an area where bugs can be devastating. Every defense should include sensors that report where and how the defense is being used.

HOW CONTRAST SUPPORTS CONTINUOUS STANDARD DEFENSES

Contrast verifies that the organization's standard security controls are in place and properly configured. Contrast Enterprise may be configured to provide custom guidance that encourages developers to do the right thing in their code.

"Trying to build secure applications without a set of strong standard defenses is like trying to build a car with a bunch of stuff you found at the junkyard."

Jeff Williams
Contrast Security CTO

ACTIVITY 06: CONTINUOUS ATTACK PROTECTION

WHY IT IS CRITICAL

If applications can't detect and block attempted attacks, the likelihood of a successful breach increases significantly, and nobody will ever know it happened.

METRICS

- **Able to deploy defenses quickly to every application.**
- **Evidence of attempted and blocked attacks.**

"Attack Protection" means that the organization is continuously monitoring for attacks and has established the ability to block them. In addition, the organization has established the
ability to respond to breaches quickly and thoughtfully.

Applications are going to be continuously attacked. The vast majority of these attacks will be ineffective "doorknob rattling" in the hopes of uncovering an easily exploitable vulnerability. A small subset of these attacks will target a portion of a company's attack surface to reveal a real vulnerability. And a fraction of those attacks will be successful attacks that steal data, corrupt systems, deny service, or hijack proprietary technology. Although the application layer is the leading cause of breaches, most organizations do not have visibility into application-layer attacks.

Modern applications must have the ability to detect and block their own attacks. Attempts to detect attacks outside the application at the network perimeter simply cannot hope to understand enough of the protocols, data formats, and application logic to identify anything but the most trivial attacks. In CAS, Attack Protection ensures that all applications are equipped with Contrast Enterprise with RASP in order to detect and block attacks, including:

- *Bot traffic*
- *Ineffective attacks*
- *Attacks against custom code*
- *Attacks targeting vulnerabilities in third-party code (libraries, frameworks, servers, platforms)*

In addition, because the attack landscape can change very quickly, Attack Protection is charged with deploying new attack defenses quickly. Using information from Threat Intelligence and Security Research, new Contrast Enterprise rules are deployed to ensure that novel attacks against both custom and third-party code cannot be exploited.

Attack Protection informs existing alerting and logging infrastructure, such as syslog, SIEM, and alerting channels to ensure an appropriate response when a successful attack is detected. Attack Protection also must prepare a plan that covers forensics, evidence preservation, public relations, notification, meeting legal requirements, insurance, data recovery, and remediation. Being ready to respond quickly can minimize much or all of the reputation damage associated with a breach, while a poorly executed response can magnify the damage.

HOW CONTRAST SUPPORTS CONTINUOUS ATTACK PROTECTION

Contrast Enterprise with RASP enables every application to detect and block attacks accurately. In addition, Contrast also provides "CVE shields" that prevent attackers from exploiting known vulnerabilities in libraries — "virtual patches" that allow quick deployment of rules to block novel attacks — and "log enhancers" that add detailed security logging without having to recode and recompile applications.

"Attack-aware software applications provide attack detection and real-time defensive response with a very low false-positive rate. This technique allows an application to detect and neutralize a threat before the attacker exploits a known or unknown vulnerability."

Michael Coates
OWASP Chair

ACTIVITY 07: CONTINUOUS SECURITY ORCHESTRATION

WHY IT IS CRITICAL

Without coordinating application security across development, operations, management, and the Board of Directors, informed decisions can't be made about risks and investments may be wrongly prioritized.

METRICS

- **Is the Board able to understand and meaningfully contribute to security decisions?**
- **Is there clear justification for improvement projects?**

"Security Orchestration" is charged with making sure that the continuous application security program is running smoothly and producing great results. There are two main parts of Security Orchestration, ensuring that the Board of Directors is aware and involved with application security, and managing priorities for the other seven CAS activities.

The first priority is to create visibility into application security. There are numerous consumers of this information, including developers, testers, architects, security, audit, and senior management. The Contrast TeamServer gathers data from across development and operations, and generates dashboards showing:

- *Inventory: Detailed application portfolio and bill of materials for each application*
- *Application Coverage: Across applications, web services, APIs, interfaces, etc.*
- *Vulnerability Coverage: Across code, configuration, libraries, frameworks, and servers*
- *Defense Coverage: How well-defended are applications?*
- *Attack Coverage: Details of attacks and their impact*

These dashboards are always up to date because they are powered by continuous Contrast Enterprise agents. The data from the Contrast TeamServer can be integrated into almost any other dashboards already in use, such as tools like SonarQube and Splunk.

CAS makes essential application security data available for reporting to senior management and the Board of Directors. The Security Orchestration group is responsible for translating this data into recommendations and prioritized initiatives to improve security for the organization.

The second priority for Security Orchestration is to manage the budget, staffing, and scheduling the initiatives chosen by management. These projects should be executed and managed like any other initiative, with a clear plan, success criteria, and appropriate resources.

HOW CONTRAST SUPPORTS CONTINUOUS ORCHESTRATION

Contrast Enterprise provides visibility into application security across the entire application portfolio. In addition, Contrast makes it possible to take strategic guidance from upper management and implement it across both development and operations. In essence, CAS establishes an application security "control plane" that Security Orchestration can leverage to drive the program forward.

"Within development, especially iterative agile development, you really need to have security bleed into the ecosystem."

Justin Somaini
Chief Trust Officer at Box

ACTIVITY 08: CONTINUOUS SECURITY TRAINING

WHY IT IS CRITICAL

Security is often quite counter-intuitive. So it's unfair to expect development and operations teams to create and operate secure applications without training in how to do that.

METRICS

- **What percentage of the development and operations team has been trained in secure coding?**
- **What is the average score students receive on post-training tests?**
- **What is the average delta between pre- and post-training tests?**

"Security Training" ensures everyone understands why application security is important to the business, how the application security program actually works, and what everyone's individual responsibilities are. In addition, training is a good way to familiarize staff with the organization's Security Architecture and to ensure that they know when to use defenses and how to use them correctly.

In CAS, the first level of Security Training happens automatically as Contrast Enterprise with IAST provides instant feedback as developers do their normal work. This "microtraining" happens directly in people's work environments, such as a developer's IDE, QA environment, or bug tracker. Continuous analysis and instant notifications allow development and operations to adjust their behavior and correct security mistakes when they cost almost nothing to fix. Notification leverages the organization's existing infrastructure, such as email, Slack, HipChat, and PagerDuty.

In CAS, new additions are being made to the security architecture all the time. Getting an expert to do a briefing, video, or blog about the new threat, vulnerability, or defense is a great way to keep the staff up to date on the latest security information. Briefings should be tailored to the organization and must include defense details in addition to threat and vulnerability information. An occasional memo or speech emphasizing the importance of security should be published by a senior executive to keep everyone aligned.

Application security eLearning and instructor-led training are less immediate, but establish a strong foundation of application security knowledge, skills, and abilities. They are a great choice for new hires who need a lot of coverage quickly. Traditional training should include exercises, quizzes, and final tests. Use the analytics to measure exactly what percentage of development and operations has received training and how well they comprehended each of the instruction topics.

HOW CONTRAST SUPPORTS CONTINUOUS SECURITY TRAINING

Contrast Enterprise provides "microtraining" by instantly alerting the relevant departments and providing detailed remediation guidance when a new vulnerability or attack is detected. Contrast works within existing development and operational tools to provide this training instantly, exactly when it is needed.

AN EXAMPLE OF APPLYING CONTINUOUS APPLICATION SECURITY

CAS focuses on establishing a line-of-sight from a particular threat through the appropriate defenses and assuring that protection is correct and complete. The first step is to identify and understand the biggest threats to the business and prioritize them. SQL injection is a good example since most organizations should have it near the top of their list.

The next step is to identify and capture a "defense strategy" that everyone can understand. Essentially, what combination of secure coding, perimeter defense, training, penetration testing, logging, and luck is going to keep the enterprise safe?

In the past, a defense strategy might have been called: a policy, requirements, secure coding guideline, or maybe a test case of some kind. It might have been implicit in the tools or people employed to do security testing. Most likely it was a paper document that wasn't clear and wasn't understood or followed. But CAS makes choices explicit and codifies them into "sensors" that continuously verify that the defense strategy is implemented and working. Together, the collection of all your defense strategies and sensors is called "Security Architecture."

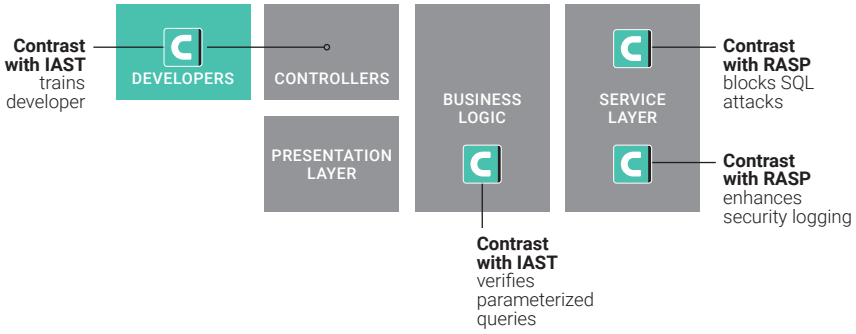
Let's say the decision is made on the following defense strategy for SQL injection. Note that the defense strategy for a different threat is likely to be very different.

Primary Defense: Use parameterized queries or persistence APIs

Secondary Defense: RASP for attack protection

Secondary Defense: RASP for enhanced logging of queries

Secondary Defense: Microtraining via instant IAST remediation guidance



When this protection is rolled out, legacy applications are analyzed by Contrast Enterprise with IAST in a distributed fashion, in parallel. Within a few hours (and from that point forward) there will be a portfolio-wide and always up-to-date picture of where SQL injection problems are located. With this visibility, strategic choices can be made about how to most cost-effectively remedy these issues. Developers working on new code will be instantly notified when new SQL vulnerabilities are introduced. These alerts provide "microtraining" on the exact SQL injection mistake the developer made, and exactly what they need to do to both fix it and prevent it in the future.

In operations, Contrast Enterprise with RASP protects applications, providing instant insight into attempted attacks across the entire software portfolio. Data is automatically collected into the SIEM to provide a rich picture of attacks across the enterprise. RASP provides the capability to instantly deploy defenses against SQL injection attacks across the entire application portfolio from a unified command-and-control dashboard.

By following this strategy, organizations are continuously protected against both SQL injection vulnerabilities **and** attacks. Across an organization, Contrast's RASP dashboard shows performance over time against defined security architecture, encouraging continuous improvement. The cost associated with new SQL injection threats has dropped to almost zero, as most new issues will be fixed by developers before the code has even been committed to a code repository.

Essentially, an application security "control plane" is established, allowing the organization to:

- *Profile all the applications in the organization and build a unified and detailed portfolio*
- *Measure how well all applications comply with the security architecture*
- *Protect applications against attacks and gain visibility into an attacker's activity*
- *Quickly respond to novel attacks and vulnerabilities across the entire portfolio*

So, what is Continuous Application Security? It's exactly what was described above:

The organization understood its threats, defined "secure" selected defenses, deployed vulnerability sensors, and established ongoing attack detection and blocking.

Everything is measured carefully and reported to senior management who can make informed decisions about application security. The bar has been permanently raised for application security.

> "Rugged is NOT the same as 'secure'. Secure is a possible state of affairs at a certain point in time. But 'rugged' describes staying ahead of the threat over time. Rugged organizations create secure code as a byproduct of their culture. You are rugged because you run the gauntlet, instrument your organization and your code, constantly experiment to see if anything breaks, and survive the process of hardening yourself through real-world experience. Rugged organizations produce rugged code – designed to withstand not just today's threat, but future challenges as well."
>
> **The Rugged Software Project**

SIMPLE CAS SELF-ASSESSMENT

Below is a worksheet to determine how well the goals of Continuous Application Security are achieved. See how many of the activities you may have completed in your CAS program.

Table 2. Quick CAS Assessment

CAS Activity	Matrix	Yes/No
1. Continuous Threat Intelligence	Are you aware of new threats, vulnerabilities, and defense techniques immediately when they are released?	
2. Continuous Security Architecture	Have you defined a defense strategy for the threats you face, and created sensors to let you know where you are vulnerable and attacked?	
3. Continuous Security Research	Do you have security experts constantly challenging the correctness, strength, completeness, and coverage of your security defenses?	
4. Continuous Security Integration	Are ordinary development and operations teams empowered with automation that enables them to find and fix their own vulnerabilities?	
5. Continuous Security Defenses	Do you have strong, simple, and easy-to-use security defenses available for people building and operating applications?	
6. Continuous Attack Protection	Do your applications have the capability to detect attacks and rapidly deploy protections?	
7. Continuous Security Orchestration	Have you established continuous visibility into application security across the entire portfolio for senior management and the Board of Directors?	
8. Continuous Security Training	Have you enabled development and operations with the knowledge, skills, and abilities they need in order to reliably produce and operate secure applications?	

KEY CAS CONCEPTS

Application security has been around for over 20 years, and many of the traditional concepts have some value at their core. But in their current incarnation, they often don't scale and don't work in real time, and are therefore incompatible with modern software development. In CAS, these key concepts are restructured to be continuous, real time, and scalable.

Continuous

In a world where applications are attacked every day, new threats are frequently unleashed, and as new vulnerabilities are discovered often, enterprises need continuous visibility and control of security across their entire application portfolio. In CAS, vulnerabilities and attacks are instantly detected and reported, not waiting undetected for an annual scan. Most critically, organizations must be able to deploy new defenses immediately in case of an attack, without having to rewrite and redeploy code.

Instrumentation

Instrumentation is a safe and proven way of adding missing capabilities to applications without having to recode, retest, and redeploy them. Many popular logging and application performance management products have relied on instrumentation for over a decade. Security instrumentation adds real-time capabilities to identify vulnerabilities, block attacks, analyze libraries, provide detailed application inventory, and even enable centralized policy command and control.

Interactive Application Security Testing (IAST)

IAST is an assessment technology that uses instrumentation to detect vulnerabilities by watching applications as they run. IAST is simple to deploy and has significantly more context about the application than SAST or DAST tools, yielding far better coverage and accuracy. Contrast Enterprise provides both IAST and RASP in a single agent that works across the entire software lifecycle.

Real-Time Security Feedback

Security costs increase dramatically the longer a vulnerability exists, a threat goes unaddressed, or an attack goes undiscovered. Providing security feedback in real time means that vulnerabilities can be eliminated as part of normal software development, and that attacks can be neutralized before they get started. That eliminates the costs of triaging, documenting, tracking, scoring, and retesting risks.

Runtime Application Self-Protection (RASP)

RASP is a defensive technology that uses instrumentation to add the capability of detecting and blocking attacks to applications at runtime. RASP is simple to deploy and has significantly more context about the application than a WAF or other external protection, and therefore is much more accurate. Contrast Enterprise provides both IAST and RASP in a single agent that works across the entire software lifecycle.

Security

Security isn't some mystical property that can only be glimpsed by wizards. CAS makes it concrete – something to build, test, and measure. In CAS, security is knowing that strong defenses are assigned to the threats that matter most to the business; that those defenses are correct, configured properly, and deployed in all the right places; and that they can detect and block both known and novel vulnerabilities and attacks.

Sensor

In CAS, a "sensor" is a set of security instrumentations designed to analyze code for a particular vulnerability, detect and block a particular attack, or create visibility into some aspect of an application, such as components, frameworks, architecture, or backend connections. These sensors are the basis of both IAST and RASP technologies, and delivered in a single agent in Contrast Enterprise.

240 3rd Street | Los Altos, CA 94022
Phone: 888-371-1333

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