CONTRAST SECURITY

WHITEPAPER

CONTRAST ASSESS

MARKET-DEFINING INTERACTIVE APPLICATION SECURITY TESTING FOR MODERN AGILE AND DEVOPS METHODOLOGY

CONTENTS

INTRODUCTION

Contrast Assess is a revolutionary application security testing solution that transforms an organization's ability to secure their software by making applications self-protecting.

Contrast Assess infuses software with vulnerability assessment capabilities so that security flaws are quickly and automatically identified. Organizations can use Contrast Assess to secure their applications without changing the application software stack, or how they build, test, or deploy code. The result is accurate, continuous vulnerability assessment that integrates seamlessly with existing software development and security processes, scales across the software development lifecycle and the entire application portfolio, and easily outpaces traditional solutions.

This whitepaper will cover how Contrast Assess' unique Interactive Application Security Testing (IAST) architecture makes software capable of assessing itself continuously for vulnerabilities, while providing the highest accuracy, efficiency, and coverage.

WHAT IS INTERACTIVE APPLICATION SECURITY TESTING?

Analyst firm Gartner has defined the IAST category as follows:

"Interactive application security testing (IAST) uses instrumentation that combines dynamic application security testing (DAST) and static analysis security testing (SAST) techniques to increase the accuracy of application security testing. Instrumentation allows DAST-like confirmation of exploit success and SAST-like coverage of the application code, and in some cases, **allows security self-testing during general application testing**. IAST can be run stand-alone, or as part of a larger AST suite, typically DAST."[1]

How is the Contrast Assess approach to IAST unique?

Gartner's definition is relatively broad, allowing for a variety of solutions to be classified as an IAST product. In practical terms, the difference between IAST products is significant.

Specifically, only Contrast Assess addresses the phrase above in bold (emphasis added), which represents a complete shift in how application security is performed: a product that enables security self-testing during general testing and eliminates the need for a separate security testing phase. Contrast replaces the point-in-time vulnerability assessment "snapshot" of SAST, DAST and other IAST solutions with a continuous flow of telemetry about vulnerabilities.

Competing IAST solutions conform to the Gartner definition, but find vulnerabilities using DAST or DAST-like techniques to simulate attacks against a running application. Organizations using those solutions must still wait for a separate security testing scan to complete, to receive a snapshot of their application security status from that scan. This is not a continuous view.

Contrast Assess neither scans nor attacks applications, but uses patented state-of-the-art deep security instrumentation technology to combine the most effective elements of static and dynamic testing, software composition and configuration analysis technologies, and deliver them directly into applications.

Contrast Assess performs static analysis **before** the code starts running – including custom code as well as all code found in libraries, frameworks, application servers, and the runtime platform – and adds instrumentation to observe and report on the running code as it executes.

How is Contrast Assess Different than SAST?

Traditional **SAST** solutions attempt to build a model of an application and pseudo-execute it from known entry points – but SAST is blind to how all the pieces of an application work together and operate at runtime, and can generate extensive false negatives and false positives. Contrast Assess observes real data and control flow activity from within a running application, and identifies a much broader range of vulnerabilities – with greater accuracy – than traditional SAST solutions. Contrast Assess is fully distributed and infuses each application with a "self-assessment" capability that performs analysis continuously, in parallel, across an entire portfolio of applications. SAST solutions cannot operate in a distributed manner because they rely on experts to analyze and triage results, which creates a significant bottleneck.

How is Contrast Assess Different than DAST?

Traditional **DAST** tools try to exploit the running application with attacks, and detect vulnerabilities by analyzing HTTP responses – but DAST is blind to what occurs within the application, and provides only limited coverage of an application. Contrast Assess performs a complete static analysis of all the code, as described above, and analyzes HTTP traffic and HTTP responses from inside the application. Because Contrast Assess works from within the application, it also provides more accurate analysis than traditional **Penetration (Pen) Testing** tools.

And, unlike either SAST or DAST products, Contrast Assess uses techniques found in **Software Composition Analysis (SCA)** tools to build an inventory of all the libraries, frameworks, and microservices used by the application to identify vulnerabilities across all those components.

CONTRAST ARCHITECTURE

Contrast Assess includes an agent that installs on each application server and a centralized management server to collect, analyze and report on vulnerabilities identified by the agents, and to control the deployment. It requires no changes to the build-test-deploy cycle, to the software stack or the runtime environment.

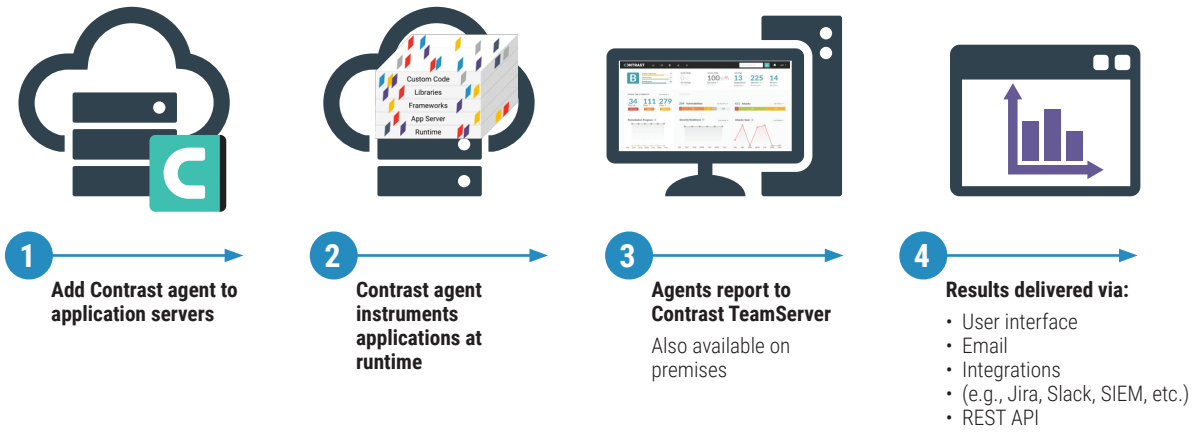
Contrast Agent

Contrast Assess uses an **agent** on each application server to instrument applications with passive sensors to discover and report vulnerability data to the Contrast TeamServer.

Contrast TeamServer

Contrast Assess operates under a command and control architecture. Contrast **TeamServer** is a centralized server with a modern user interface for managing the deployment and reporting on vulnerabilities. Contrast TeamServer includes a number of native integrations with popular software development tools and supports a variety of notification, reporting, and API access methods, including a comprehensive RESTful API that can be used for custom integrations.

Figure 1. Contrast Assess Agents and TeamServer



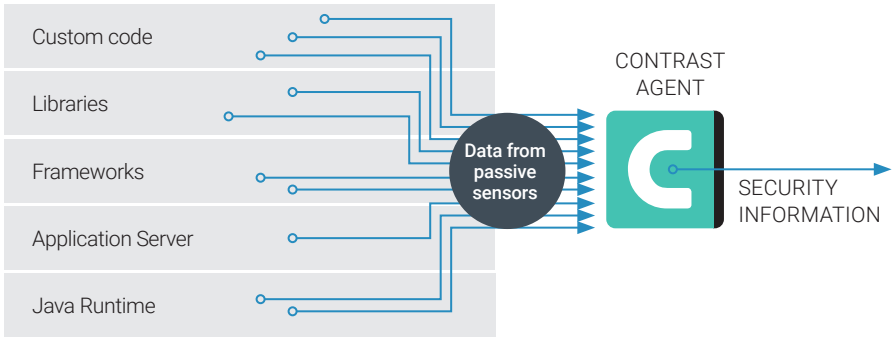
DEEP SECURITY INSTRUMENTATION

Deep security instrumentation is a technique performed at runtime, as program code loads into memory.

Software instrumentation has been used for many years for application performance and logging. Companies like New Relic and AppDynamics instrument running applications to improve performance. They send findings to a server, and use that data to create useful reports and dashboards.

Similarly, when the Contrast agent is installed, it automatically and invisibly instruments applications with a rules engine and simple passive sensors that gather information about the application's security, architecture, libraries, and frameworks, and report that information to the central Contrast TeamServer. Since the evidence needed to accurately identify and describe a vulnerability is usually scattered in several places within an application, Contrast's **deep security instrumentation** has numerous advantages over traditional static and dynamic analysis tools, which have no inside-the-running-application "context" to inform their analysis. Contrast's patented methodology provides developers with instant security feedback as soon as they write and run their code, instead of weeks or months later.

Figure 2. Deep Security Instrumentation



Instrumenting an application with passive sensors provides more access to information about the application, and delivers unprecedented levels of speed and accuracy in identifying vulnerabilities.

By instrumenting the entire application, Contrast Assess can trace data and execution through custom code, libraries, frameworks, and even the runtime platform.

For example, Contrast's instrumented analysis can identify:

- Credit card numbers extracted from a database and flag when they end up in a log file
- A weak encryption algorithm specified in a properties file
- Data flowing from within an encoded cookie, through a data bean, into a session store, into a JSF component, and finally into a browser – indicating cross-site scripting (XSS) weakness.

Contrast Assess instrumentation monitors and reports on the following information with an application:

HTTP Requests contain useful information about authentication, sessions, use of SSL, certain headers, CSRF tokens, and sitemap information. Contrast passively monitors HTTP requests with sensors, enabling testing in just about any environment. Other techniques like DAST use HTTP requests to carry out security tests, and require an expert to configure and run them.

HTTP Responses provide a wealth of security information. Security headers, CSRF tokens, SSL, autocomplete, access control, and session management are all easily revealed by analyzing responses.

Data Flow is the holy grail of security analysis. Most of the interesting injection problems can only be revealed with a strong data flow engine. Static analysis tools used to be the only way to analyze data flow, but they were difficult to use and suffered from false alarms. Newer instrumentation-based approaches can perform highly accurate "whole application" data flow analysis including frameworks, libraries, dynamic class loading, and more.

Control Flow shows whether the code executes the right steps in the right order, and analysis can reveal several types of vulnerabilities. For example, it will check if the proper hashing occurred during the authentication process, or if there is an access control check in every Struts action.

Libraries and Components are the building blocks of modern applications. Analyzing these components for known vulnerabilities is of critical importance. It is also important to know if the library has unknown vulnerabilities or hazards that an unsuspecting developer may have used improperly.

Frameworks are used by all modern web applications and web services, and many frameworks have a set of security controls built in. Instrumentation monitors and reports on things like whether these controls were used properly, or if the developers invoked them in all the right places. Understanding the framework pays off immensely in providing useful context-sensitive guidance to developers.

Application Server is like a framework, in that it also has numerous security controls. Contrast's instrumentation checks if the application is using security controls. Tools can do better analysis when they understand the application server in use and what protection it provides.

Configuration Data can be stored in XML files, property files, databases, and the like. Most security controls can be configured on or off. Many of them have detailed configuration that allows their behavior to be tailored. This is a direct and very accurate way to measure application protection.

Backend Connections provide information about where sensitive data is stored, which is key to being able to perform great security analysis. Tools that can understand backend connections are much more likely to be able to see critical vulnerabilities and describe them to developers.

CONTRAST ASSESS DIFFERENTIATORS

1. Ideal for Agile and DevOps

Today's high velocity Agile software development models need application security testing that is efficient, non-intrusive, and integrated into development environments. Contrast Assess is designed to work interactively with developers as they write and test code. A restful API provides seamless integration with tools like Slack, HipChat, JIRA, or email to notify developers when they've introduced a vulnerability.

DevOps teams can add the Contrast agent into their standard server application build and test automation tools like Jenkins, Maven, Gradle, and Bamboo to discover and report any vulnerabilities within an application. Contrast Assess automatically enables security analysis while test cases and automated test scripts are executed, and fails any build that has excessive vulnerabilities.

2. Highest Accuracy

Unlike legacy application security testing solutions, Contrast Assess produces accurate results without dependence on application security experts.

Contrast Assess combines the most effective elements of Interactive (IAST), Static (SAST), and Dynamic (DAST) application security testing technology, software composition analysis (SCA), and configuration analysis, and delivers them directly into applications.

This combination of techniques produces the telemetry necessary to detect vulnerabilities with virtually no false positives and no false negatives.[2]

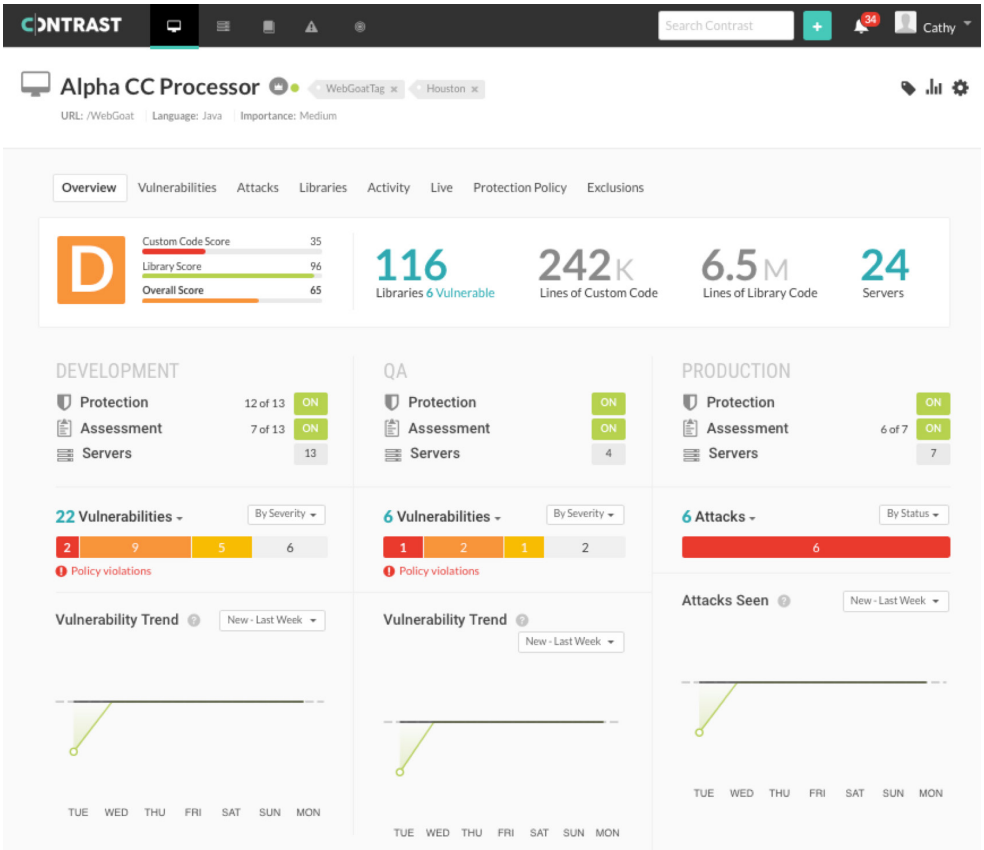
3. Continuous Analysis

Unlike legacy approaches that require time-consuming, late-stage scans that disrupt the development process, there is no separate security testing phase with Contrast Assess. Contrast Assess uses its deep security instrumentation to produce a continuous stream of accurate vulnerability analysis whenever and wherever software is run. Development, QA, DevOps, and security teams get results as they develop and test software, enabling them to find and fix security flaws early in the software lifecycle, when they are easiest and cheapest to remediate.

Contrast Assess displays security analysis results in a real-time dashboard of critical security information, vulnerabilities, and remediation advice across all applications. Each application receives an easy-to-read and understand letter grade for security based on both security and analysis coverage.

The results in Figure 3 were captured by Contrast after only a few minutes of browsing a deliberately flawed, vulnerable open source software application that was created to assist developers with application security.

Figure 3. Continuous Analysis



All applications are presented in a clear, understandable dashboard in the Contrast TeamServer user interface. Each application also has its own dashboard with a letter grade for both security and coverage.

Contrast Assess has no separate testing phase. It produces results continuously so development teams can release software as fast as they want, knowing it is secure.

4. Scalable Architecture

Contrast Assess scales because it instruments application security into each application, delivering and distributing vulnerability assessment across an entire application portfolio. Every running application continuously produces results in parallel.

Contrast differs from legacy approaches that require application security experts – a human element that does not scale.

Even with hundreds or thousands of applications, Contrast Assess provides an always up-to-date dashboard for each application with vulnerability, library, architecture, and other security details.

KEY FEATURES OF CONTRAST ASSESS

Extensive Vulnerability Coverage

Contrast provides extensive coverage over most common vulnerabilities, including the OWASP Top Ten[3]. Note that there are no rules in Contrast Assess for Insufficient Transport Layer Protection (A6) and Security Misconfiguration (A5), since these are typically enforced outside of the Java environment.

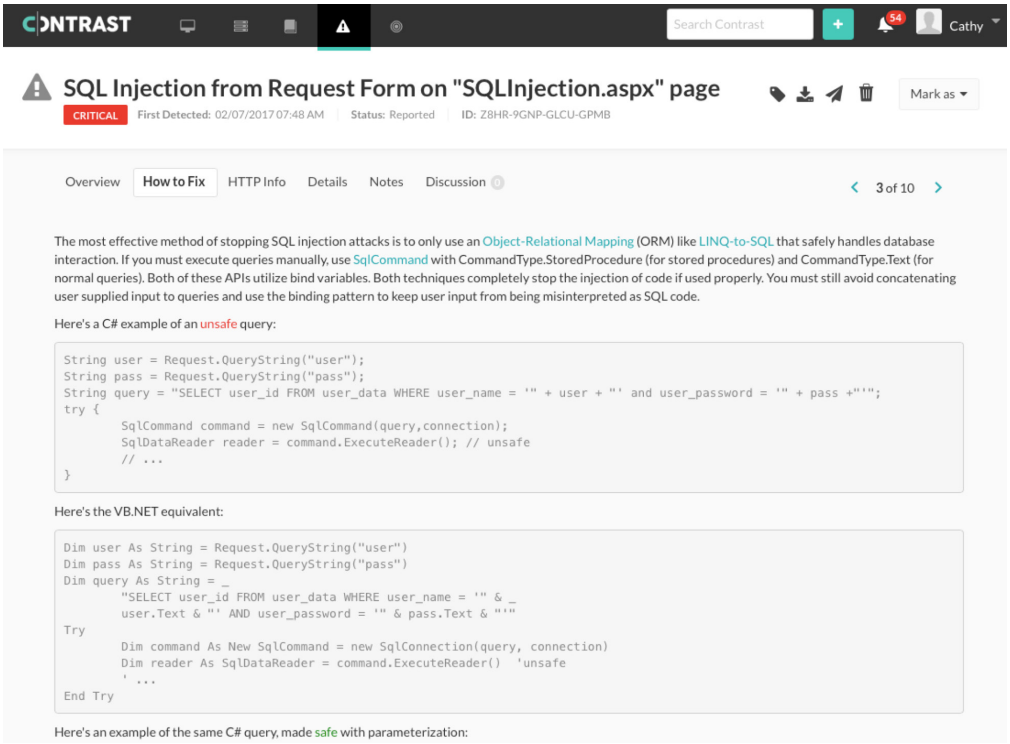
- SQL injection (A1)
- Blind SQL injection (A1)
- Command injection (A1)
- Reflected XSS (A3)
- Stored XSS (A3)
- Session ID disclosure (A2)
- Path traversal (A4)
- Insecure direct object reference (A4)
- Authorization missing (A4)
- Weak hash algorithm (A6)
- Weak encryption algorithm (A6)
- Underprotected APIS (A10)
- Arbitrary forward
- Unchecked redirect
- No size limit on data read
- File download injection
- HTTP header injection
- And more…

Code-Level Remediation Advice

The Contrast user interface also explains vulnerabilities to those that need to understand and fix them. Contrast's innovative Security Trace format pinpoints exactly where a vulnerability appears in the code, and how it works.

The SQL Injection example illustrated in Figure 4 explains to the developer exactly how untrusted data flows through the application and gets embedded in SQL query without either validation or parameterization. Contrast "speaks the developer's language," and provides remediation guidance that is easy to understand and implement.

Figure 4. SQL Injection Remediation

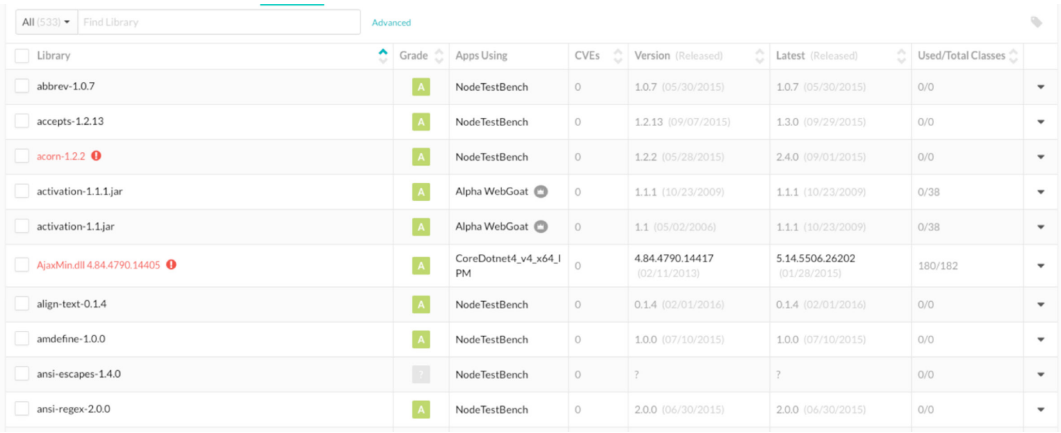


Contrast vulnerability reports include all the details needed to understand the problem, find it in the code, and how to fix it correctly.

Third-Party Code Analysis

Like icebergs, 80 percent of the code in modern applications is "beneath the surface," lurking in libraries, frameworks, and other components. Applications often have 50 or more of these libraries, comprising millions of lines of potentially vulnerable code. Contrast Assess automatically analyzes these libraries and provides a detailed dashboard that lists the actual libraries, which library versions are in use, and how many classes of each library are in use for each application, as well as the number of CVEs associated with each library.

Figure 5. Library Analysis

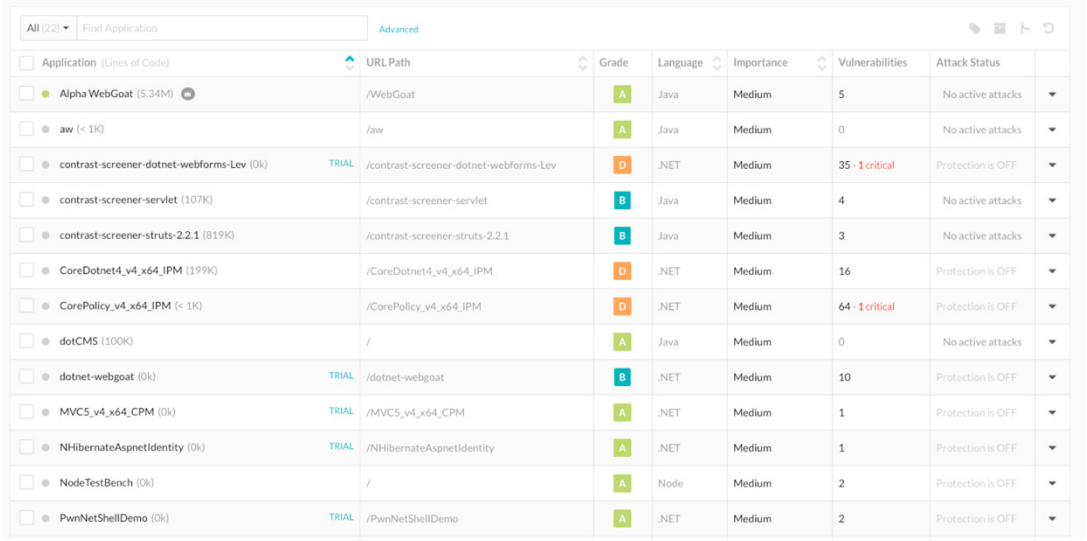


Contrast automatically discovers third-party libraries, alerts to the known (and unknown) risks they may bring with them, and provides critical versioning and usage information that helps remediate risks.

Application Inventory

Though it may seem simple, application inventory may be the hardest problem to solve for application security teams. Organizations may have hundreds or thousands of applications, Microservices, APIs – each with multiple instances of different versions installed across development and QA – and they're all constantly changing. Contrast tracks and continuously feeds that information about internal and external web services, and their relationships across an application into a unified security inventory and bill of materials that's always up-to-date.

Figure 6. Application Inventory

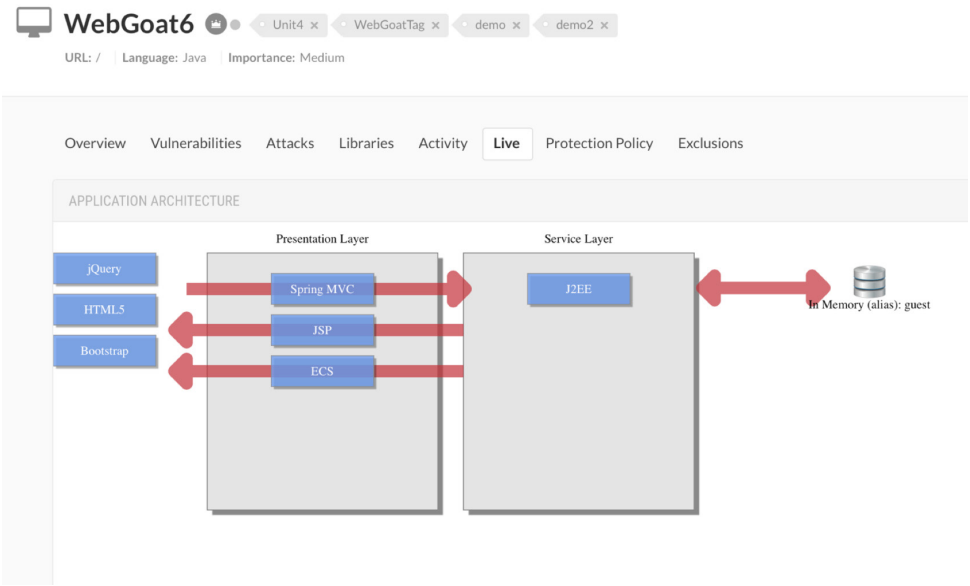


Contrast automatically displays up-to-date intelligence on which applications are in use, and application metadata including lines of code, libraries in use, component technologies, architecture, and backend connections.

Live Application Architecture

Understanding an application's architecture is extremely helpful when performing security analysis. Contrast automatically generates simple diagrams that illustrate the application's major architectural components. This information helps the developer quickly identify the meaning of a vulnerability that Contrast pinpoints. In Figure 7, Contrast has correctly identified that the WebGoat6 application has the following backend connections: web services, frameworks being used within the application (Spring, JSP and ECS), a Java virtual machine, and a database. Imagine the benefit of having up-to-date architectural information available, on demand, for every application across the entire portfolio.

Figure 7. Live Application Architecture



Contrast Assess displays the full application architecture under the Live tab.

CONCLUSION

Organizations considering interactive application self-testing technology should look beyond the Gartner definition of IAST and evaluate solutions based how well they provide application security testing that is accurate, continuous, integrated, scalable, and works in modern agile and DevOps environments.

Contrast Assess' implementation of IAST uses a highly scalable architecture – employing distributed agents with a central dashboard – to empower every application to analyze, enforce and communicate about application security across the software development lifecycle. Contrast Assess analyzes every line of code – including third-party components – for vulnerabilities, and provides instant and accurate feedback to developers. This built-in and continuous analysis means that developers can find and fix vulnerabilities as part of their development process. Application security experts can remove themselves from the critical path of software development, and spend more time on strategic security initiatives.

Contrast Assess enables organizations to prioritize their development and operations teams to remedy application security risks immediately, and reduce friction throughout the entire software lifecycle.

If your organization is ready to make a change, then sign up for a personalized demo of Contrast with one of our technical experts. Visit *www.contrastsecurity.com* and click "*get a demo*" button on the top of every page. Need more information? Go to the *products section* to read more about Contrast.

1 Source: Tirosh, Ayal. Gartner, "Hype Cycle for Application Security, 2016"
2 For information on Contrast Assess accuracy, read the Technical Brief, "Accurately Assessing AppSec with the OWASP Benchmark" at *https://www.contrastsecurity.com/owaspbenchtechbrief*
3 *https://www.owasp.org/index.php/Top_10*

CONTRAST SECURITY
240 3rd Street
Los Altos, CA 94022
888.371.1333