Product Data Sheet

Azul Vulnerability Detection: Continuously Detect Known Vulnerabilities in Your Java Applications

Detect Vulnerabilities with No Performance Penalty and Eliminate False Positives

Today, Java is ubiquitous across the enterprise, the ideal choice for development, DevOps, and operations teams worldwide.

Given Java's enterprise prevalence, securing the applications and software running on top of the platform is a must for maintaining customer trust. Business and technical leaders are faced with securing enterprise systems and customer data against an explosive growth of known vulnerabilities in 3rd party software components and applications. In Q1 2022 there were 200+ known Common Vulnerabilities and Exposures (CVEs) in 3rd party Java applications and components, many with the highest risk score, cutting across thousands of contributors.

Failure to detect and patch known vulnerabilities in Java application estates can expose organizations to significant impact and cost, including financial penalties running into the hundreds of millions of dollars, compromise of customer data, and turnover in executive staff.

Azul Vulnerability Detection at-a-glance

Continuous Detection at Point-of-Use in Production.
Accurately assesses custom and 3rd party applications' exposure to vulnerabilities in production. Compares code run to a Java-specific CVE database.

Eliminates False Positives
Eliminates false positives by monitoring the code loaded by the JVM.

NoOps with Transparent Performance
Leverages existing runtime information from Azul JVMs. Agentless approach means no performance impact and no maintenance of a separate agent in production.

Detects without Source Code
Recognizes components using unique Hash-based identifiers.

Java CVE Knowledge Base
Updated daily with the latest CVEs filtered to focus on Java-specific vulnerabilities

Azul Vulnerability Detection, an agentless cloud service, provides observability of your Java applications to continuously detect known vulnerabilities in production. By leveraging Azul JVMs, it produces more accurate results with no performance penalty and eliminates false positives.

Azul Vulnerability Detection monitors all your Java software and applications to accurately identify components loaded and in use in production. Azul Vulnerability Detection uniquely identifies each component using bytecode-aware hashing techniques. It maps these components accurately to vulnerabilities in a knowledge base updated daily with the latest CVEs from external databases, publicly available information, and more.

This enables accurate, continuous assessment of custom and vendor applications exposure to vulnerabilities in production without the need for source code. Azul Vulnerability "just works" to detect vulnerabilities in all Java applications – whether you built it or not, haven't built it in years, or are introducing a regression with a recent change.

Azul Vulnerability Detection focuses scarce human effort by eliminating false positives. It does this by monitoring the code loaded and in use by applications running on the JVM vs looking at static file listings and source code.

Since Azul Vulnerability Detection leverages runtime information from Azul JVMs, there is no additional agent or software to deploy or manage, resulting in minimal operational burden and performance overhead.

Inside Azul Vulnerability Detection

Azul Vulnerability Detection comprises runtime information from Azul JVMs sent to a backend cloud service for detection of vulnerabilities.

- The Azul JVM is used to run Java applications. Leveraging runtime information the JVM already produces saves a separate management/installation step associated with legacy agent based solutions and removes any performance impact in production environments.

Azul Vulnerability Detection Advantages

- Continuous detection of vulnerabilities in production
- Checks all Java software – custom and vendor applications, 3rd party libraries
- Easy to turn on, easy to use, ships as part of Azul JDKs
- Focuses human effort by eliminating false positives
- Agentless means no performance penalty for production detection
- Leverages Azul JVMs for existing runtime information
- Generates complete results – works on all major packaging structures including shaded jars, fat jars

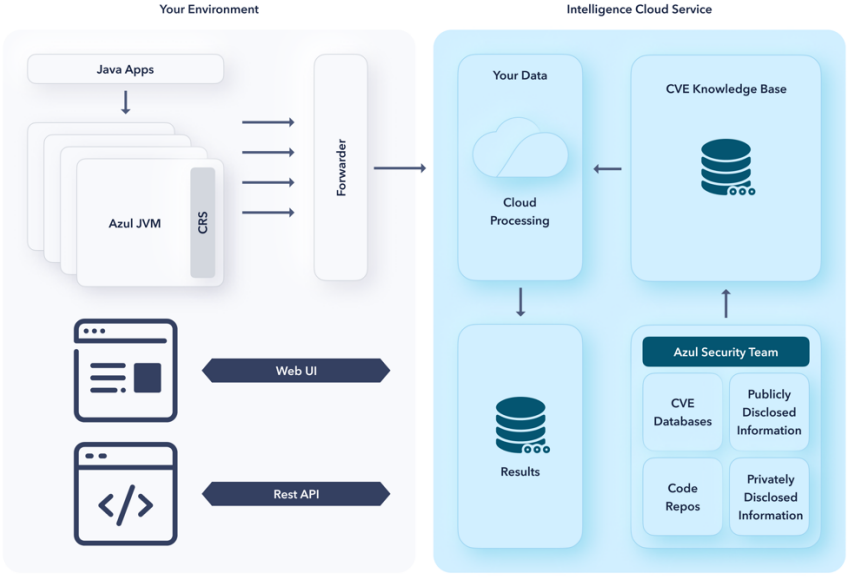
Product Data Sheet

Azul Vulnerability Detection: Continuously Detect Known Vulnerabilities in Your Java Applications

- Forwarder: Azul JVMs connect to the Intelligence Cloud Service through a Forwarder. The Forwarder is a piece of software Azul provides that acts as an application-specific proxy so that Azul JVMs can reach the Intelligence Cloud Service without needing special firewall rules.
- CVE Knowledge Base contains information about known CVEs. Azul's security team filters these down to understand which relate to Java and which customers should pay attention to. As a result of the CVE knowledge base, customers can focus on risks unique to Java.
- Component Knowledge Base is a recognition set that Azul Vulnerability Detection uses to identify many open-source components. Once recognized, Azul Vulnerability Detection checks the known component against the CVE Knowledge Base to see if it is vulnerable.
- Azul Vulnerability Detection is not another dashboard for customers to look at. Users can access data on which components are in use, and vulnerable, using either the product's API or an intuitive UI. The role of the web UI is to show the information we have and guide customers to the REST API.

Azul Vulnerability Detection Features

- Ships as part of a full Azul JDK -- easy start, easy to use
- Supports Java SE 17, 15, 13, 11, or 8
- Supports Azul Zulu and Azul Zulu Prime Builds of OpenJDK
- Detection API to access present, used, vulnerable component level information
- Web UI for looking at data quickly and as a guide to the data available in the Detection API.
- Forwarder component that facilitates communication between Azul JVMs on an internal network and Azul Vulnerability Detection.
- Component Knowledge Base for component recognition using byte-code aware hashing and unique component identifiers
- CVE Knowledge Base updated daily with Java related CVEs in accordance with National Vulnerability Database



Azul Vulnerability Detection Advantages Cont.

- Catches regressions and reintroductions of vulnerabilities
- More precise component recognition using bytecode-aware hashing techniques
- Focus on Java related vulnerabilities using Knowledge Base updated daily with latest CVEs from National Vulnerability Database
- Creates an accurate SBOM of components loaded and in use in production

Contact Azul

385 Moffett Park Drive, Suite 115

Sunnyvale, CA 94089 USA

+1.650.230.6500

www.azul.com