Vantage Inspect

Powered by ShiftLeft

Organizations need a way to test application code for security vulnerabilities that is easy, repeatable and part of an automated process

Vantage Inspect powered by ShiftLeft combines Intelligent Software Composition Analysis (I-SCA), Static Analysis Security Testing (SAST) and Infrastructure-as-code technologies into one powerful solution that comprehensively inspects an application's source code, open source libraries and infrastructure.

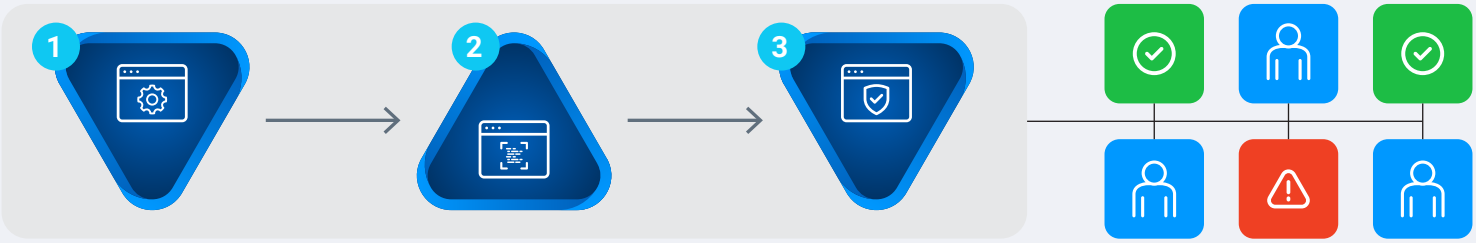
Vantage Inspect has been designed specifically to insert security checks effortlessly into developer workflows. Inspect's speed and accuracy enables security automation with every pull request, which provides the right developer with the right vulnerability information at the right time. Hence, vulnerabilities get fixed faster and earlier, which drives down mean-time-to-remediation (MTTR), reduces attack surfaces, and minimizes technical debt accrual. Furthermore, Vantage Inspect goes beyond technical vulnerabilities (e.g., The OWASP Top Ten) to identify cloud-centric vulnerabilities that traditional static analysis tools can't find, such as business logic flaws, data leakage, hard-coded literals, and insider threats.

Key Features & Functionalities

- ✓ Intelligent Source Code Analysis (I-SCA)
- ✓ Static Application Security Testing (SAST)
- ✓ Infrastructure-as-Code (IaC) functionality for cloud-native application development
- ✓ Test beyond source code — identify business logic flaws, data leakage and insider threats
- ✓ Improves compliance and risk management
- ✓ Incremental scans give developers security feedback in minutes — not hours
- ✓ Real-time and contextual security analysis delivered natively

This all-in-one approach uniquely preserves ease-of-use, accuracy, and flexibility within the SDLC.

How Does Vantage Inspect Work?



STEP 1

Configure App or Pull Repository

Start by importing one of your Github repositories or analyzing one of our demo apps. Don't have GitHub? Use our CLI.

STEP 2

Safely Analyze your Source Code

Your source code is never sent to our servers. Depending on the size of your app, a scan may only take a few minutes. So scan early, scan often.

STEP 3

Identify and Fix Vulnerabilities

Vantage Inspect offers both Data Flow intel and eLearning to provide contextual insight and mitigation suggestions

Integrate the Results Into your Workflow.

Vantage Inspect allows you to assign issues to different developers as well as track fixes and regressions between scans.

Technical Features

SUPPORTED VULNERABILITIES

+ Authentication Bypass
+ Cookie Injection
+ Cross-Site Scripting (XSS)
+ Crypto AES with ECB
+ Deserialization
+ Directory Traversal
+ Eval
+ File Write
+ Hard-Coded Password
+ Header Injection
+ Improper Certificate Validation
+ Insecure Cookie
+ Insecure JWT Algorithm
+ Insecure Puppeteer Settings
+ Insecure TLS Configuration
+ JWT Exposed Creds
+ JWT Unsafe Parsing
+ LDAP Injection
+ Mail Injection
+ Mass Assignment
+ Missing Certificate Check
+ NoSQL Injection
+ Open Redirect
+ OSS Express Body Parser
+ Path Traversal
+ Prototype Pollution
+ Remote Code Execution
+ Sensitive Data Exposure
+ Sensitive Data Leak
+ Session Injection
+ SQL Injection
+ SSRF
+ TLS Checks are Disabled
+ Trust Boundary Violation
+ Unvalidated Input Used in Cookie
+ Usage of Insecure API
+ Weak Encryption
+ Weak Hash
+ **Weak Random:** The application uses the weak random number generator crypto.pseudoRandomBytes()/Math.random()
+ XML External Entity
+ XML Injection/XXE (External Entity) Injection
+ XPath Injection

Also supports: OWASP 10 | Secrets | SCA including Attackability, etc.

SUPPORTED LANGUAGES

C# | GO* | JS | python* | Java | Terraform* | Scala | TS*

*I-SCA not supported

INTEGRATIONS WITH CI / CD

ZERONORTH | GitHub | GitLab | Trello | Bitbucket | Microsoft Azure | Travis CI | circleci | Basecamp | Jenkins | CloudBees. | CHEF | splunk> | Jira Software | puppet | asana

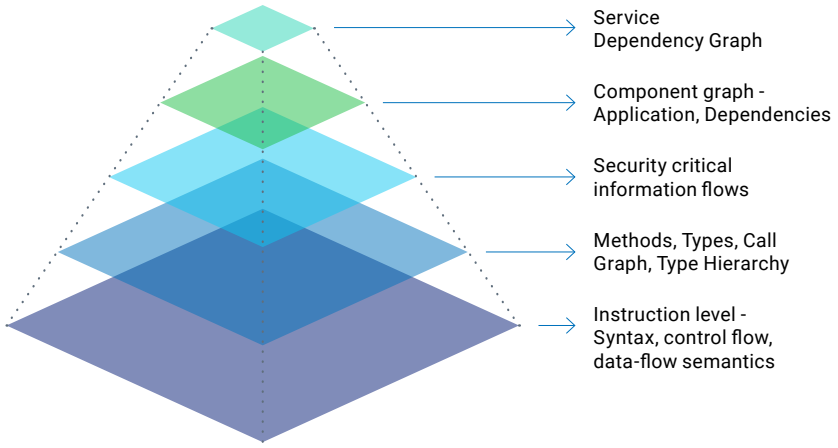
Our Technology

Code Property Graph

The Code Property Graph (CPG) is a break-through innovation in static code analysis that powers Vantage Inspect. The CPG merges multiple compilations of code into a layered graph database that can be queried to obtain security relevant information about the code. This allows for scans to simulate an application's data flow to discover complex vulnerabilities.

For each statement/expression - whether in bytecode form or source code form - a syntax tree is included in the CPG that decomposes it into its language elements. Taking into account known semantics of library functions, a data flow representation is constructed that enables intraprocedural data flow analysis. This allows for scans to be executed in a language-specific context without requiring a unique language engine for all dataflows. In essence, this process is what allows for the detection of vulnerabilities.

THE CPG PYRAMID



Throughout the analysis, additional nodes and edges are created that represent analysis results. The complete representation is then packaged into a binary that is stored in encrypted cloud storage and cannot be accessed or decrypted. The CPG becomes a unique artifact that can be queried against for subsequent scanning. It allows small changes in code to be scanned incrementally without the need to regenerate an entire dataflow model once again. It also allows for the source code to remain on-premises with no way to reverse engineer the output.

Extensible Code Analysis

The CPG is an intermediate representation (IR) independent of the programming language. This has the added benefit of making queries independent of the programming language. When a programming language is supported, i.e., its translation into CPG is complete, any query written for any other programming language can also apply to the new language.

Outcome

Apps are able to be data-flow analyzed beyond the context of a single language; dataflow can be traced between a javascript frontend and a C# backend. Additionally, this means adding new languages is significantly faster and more efficient. Only requiring a language to CPG process rather than a new language analysis engine.

A Unique Data Flow Tracker

The workhorse of the CPG is a state-of-the-art data flow tracker: it is interprocedural, flow-sensitive, context-sensitive, field-sensitive, and operates on an intermediate code representation. The data flow engine provides a configurable set of heuristics to allow reporting of findings in a desired time frame.

Outcome

This tool fundamentally understands the code and highlights the security risks that might have gone unnoticed.

Precise Flow Analysis

High-level information flows are core to the precision of our analysis. The CPG marks parameters of sources, sinks, and transformations that provide information on where data comes from, where it goes, or how it is transformed. By combining the primary data flow with its descriptor flow, we derive high-level data flows and formulate rules for their classification.

Outcome

This enables the tool to be more thorough without sacrificing performance and speed.

Intelligent Software Composition Analysis (I-SCA) identifies and prioritizes open-source vulnerabilities based on application risk

Open-Source code has gained popularity as it provides increased efficiency and streamlined delivery of application code. However, these components can introduce unknown vulnerabilities and risk. I-SCA provides security insights into 3rd party and open-source code identifying vulnerable components being utilized by development teams.

I-SCA utilizes a four-step process to identify and prioritize vulnerabiliites. It also utilizes the concept of "**Attackability**" to prioritize only a subset of OSS vulnerabilities for mitigation.

- ✓ Reduced false positives
- ✓ Prioritized vulnerability results
- ✓ Reduced overall impact on the software delivery timeline

It's highly possible that despite utilizing a vulnerable component in a web application, the deployed version may not be exploitable in the application's present state.

I-SCA provides greater insight into a "real" threat vs. a "potential" threat, thereby helping teams remediate the most relevant vulnerabilities first.

We consider a finding to be reachable if an attacker-controlled path connects application inputs to the CVE. The concept of Attackability is crucial because it tells you if someone can exploit a vulnerability in your application; if not, then you can consider this vulnerability a low priority for mitigation.

Why Organizations Select Vantage Inspect

Vantage Inspect combines the functionalities of industry leading I-SCA, SAST, Infrastructure-as-Code to deliver real-time security feedback directly to Developers and within their native code repository. Vantage Inspect helps ensure source code vulnerabilities are remediated before integrating into DevOps' CI/CD pipelines while also educating developers to produce higher quality code.

Intelligent Software Composition Analysis

Vantage Inspect's Intelligent I-SCA uses the concept of "Attacker Reachability" to prioritize only a subset of OSS vulnerabilities for mitigation. It can trace code paths that can potentially lead attackers from insecure inputs directly to open-source vulnerabilities, using the power of Code Property Graph.

Secret Detection

Vantage Inspect detects Secrets, or hard-coded values (e.g., client Secrets, username/password combinations) and sensitive information (e.g., phone numbers and addresses). Unlike "grepping" for these patterns that lead to false positives, the use of Code Property Graph identifies when secrets are being leaked without proper transformation or obfuscation.

Security Insights

Vantage Inspect provides security insights into potential code vulnerabilities as well as remediation best practices. These are conditions that can lead to OWASP Top 10 or CWE Top 25 vulnerabilities.

Vantage Inspect Contextual Information

Each vulnerability includes the data flow of a vulnerability from source to sink, a brief description of the vulnerability class and additional information on best remediation practices.

Unlimited Scans	✓
Unlimited Applications	✓
Attack Selection	✓
Remediation Advice	✓
Testing Local and Remote Applications	Local Only

NTT APPLICATION SECURITY 1741 Technology Dr. #300, San Jose, CA 95110 • 1.408.343.8300 • www.whitehatsec.com

NTT Application Security