SYNOPSYS®

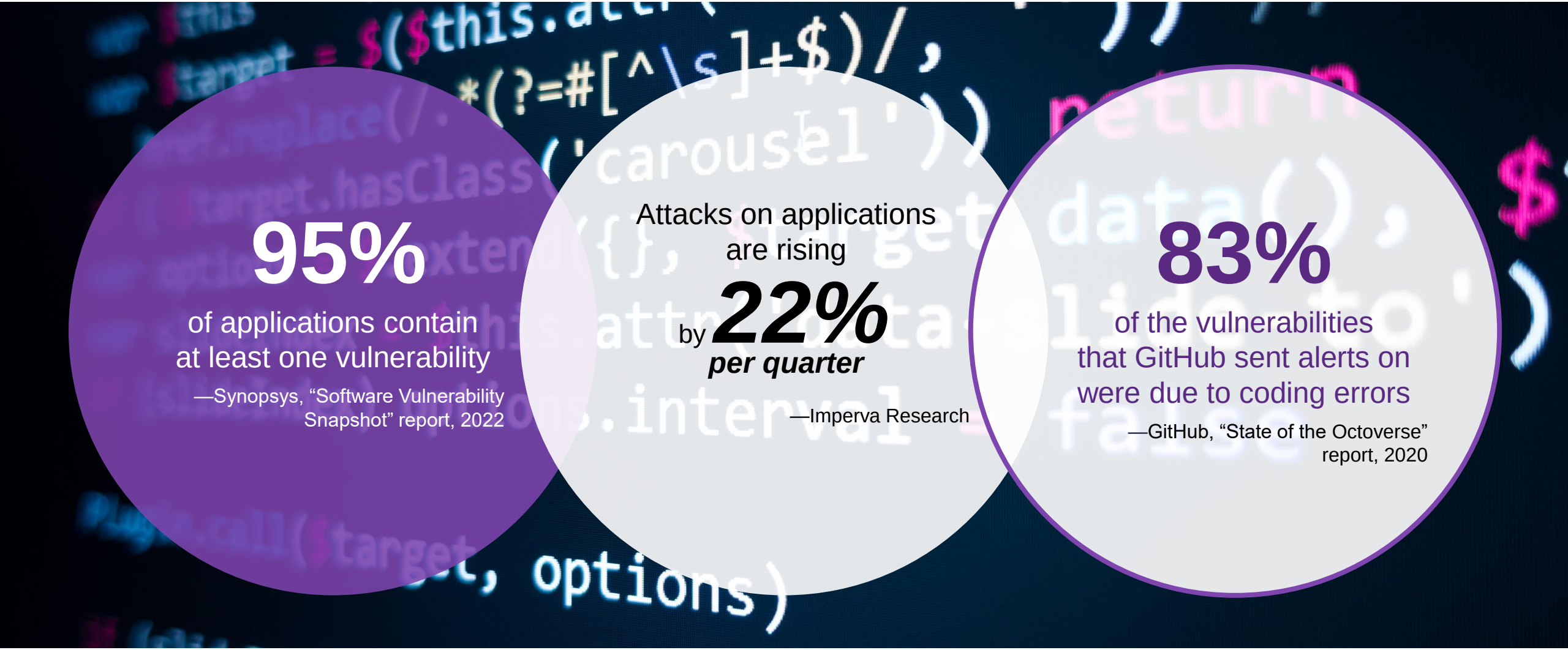
Synopsys Static Analysis

Improve code security

Pavan Kumar Boggarapu, Sales Engineer, Sr Staff

June 2024

Application code is the source of many attacks



95%

of applications contain at least one vulnerability

—Synopsys, “Software Vulnerability Snapshot” report, 2022

Attacks on applications are rising

by ***22%***

per quarter

—Imperva Research

83%

of the vulnerabilities that GitHub sent alerts on were due to coding errors

—GitHub, “State of the Octoverse” report, 2020

It’s about enabling teams to secure software at every stage

Developers

Help me build secure, high-quality code faster.

DevOps Teams

Help me automate testing without compromising velocity.

Security Teams

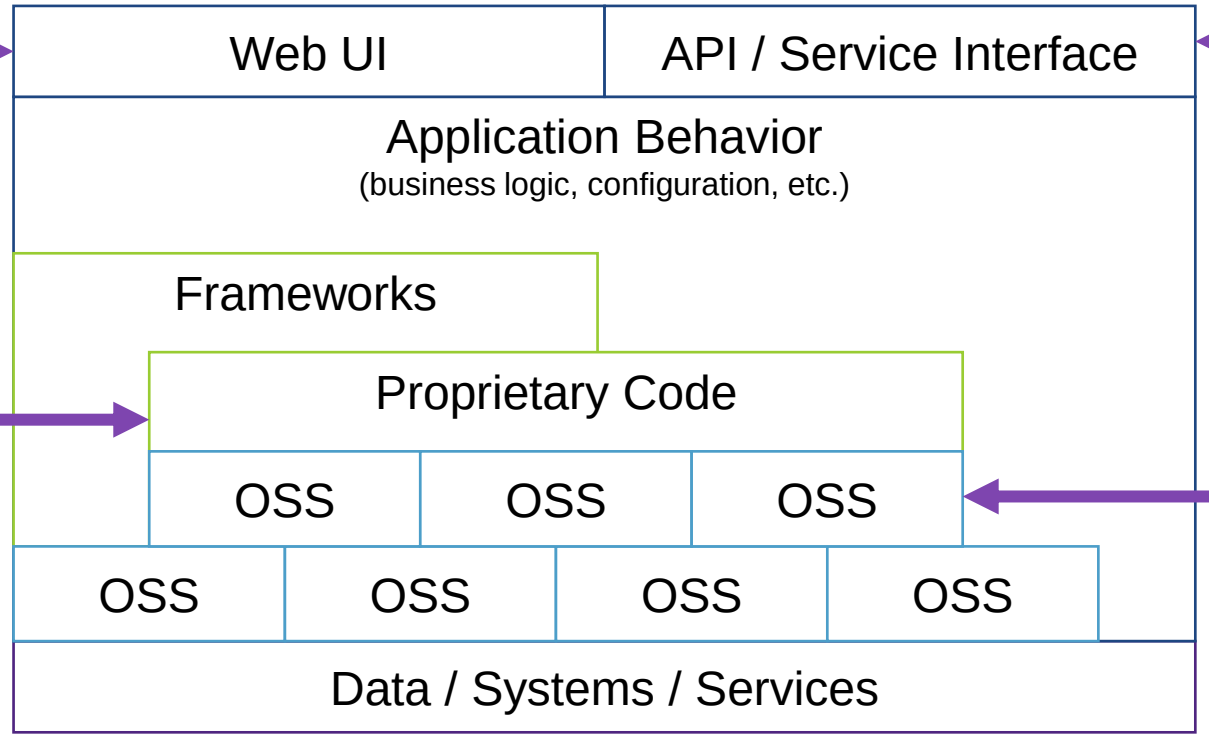
Help me manage risk proactively and focus on what matters most.

To simplify things, let's think about how apps are built

How do we integrate and automate all of this?

How do we know we've addressed runtime vulnerabilities and data protection issues before we deploy?

How do we ensure the protocols & APIs our software exposes aren't vulnerable to common hacks?



How can our developers produce code with fewer defects and security weaknesses (CWEs) without slowing down?

How do we track and manage open source use and the known security(CVE) and license compliance risks that come with it?

Synopsys solutions cover the entire application

Seeker & Whitehat/ fAST Dynamic

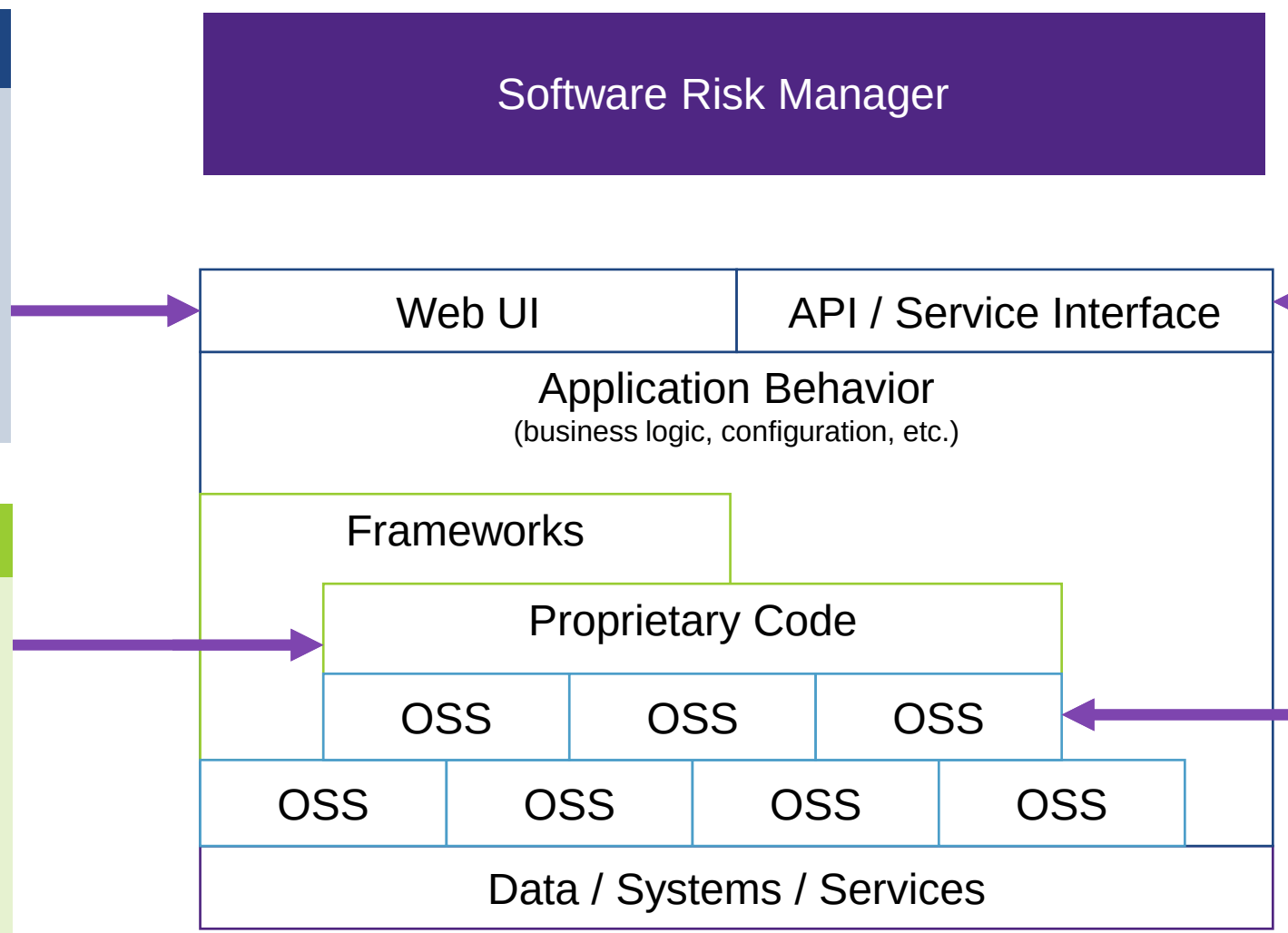
Dynamic & Interactive Analysis

Pinpoint exploitable vulnerabilities and data protection issues in web applications

Coverity/fAST Static

Static Analysis

Find and fix security vulnerabilities and quality issues in code as it is being developed



Seeker & Defensics

Protocol & API Fuzzing

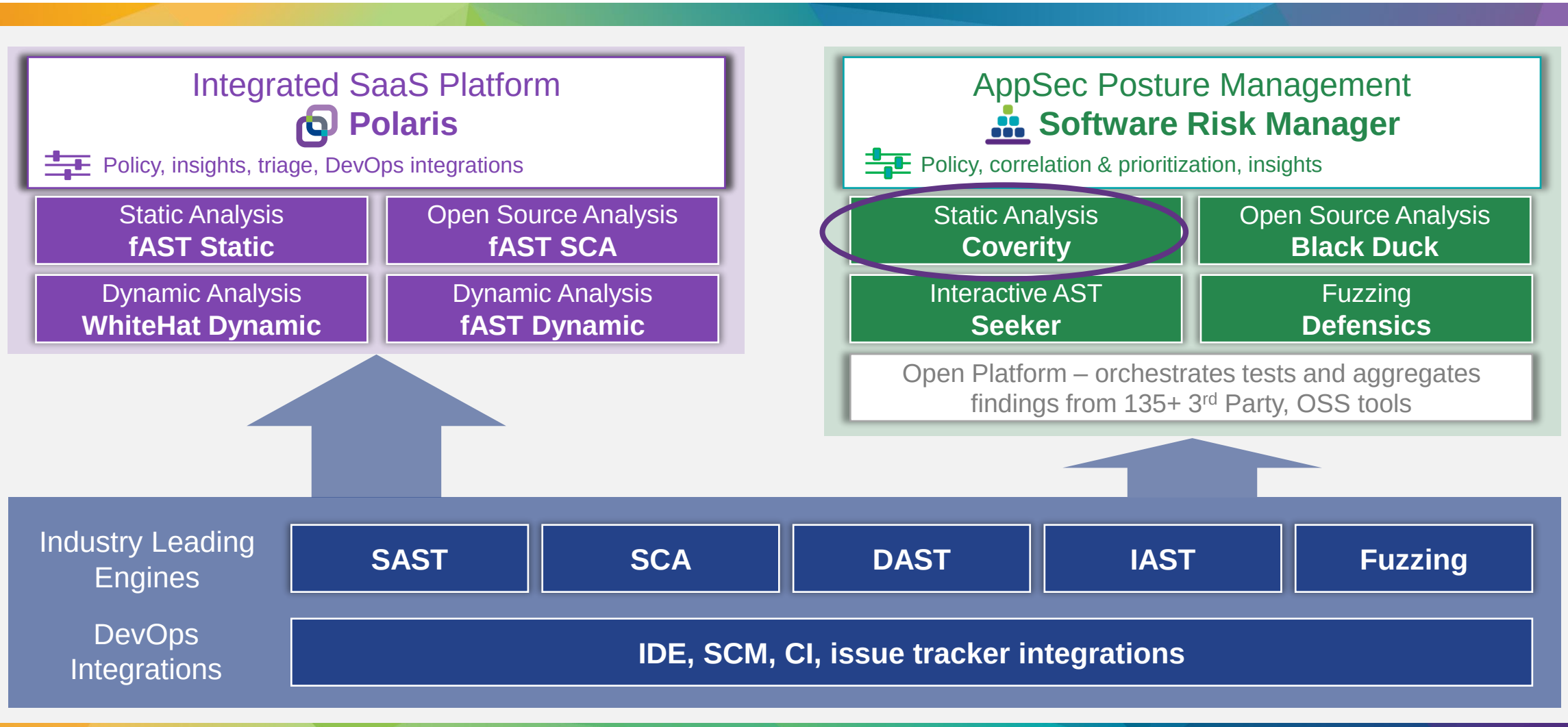
Detect vulnerabilities exposed through protocols and APIs

Black Duck/fAST SCA

Software Composition Analysis

Detect and manage open source and third-party component risks in development and production

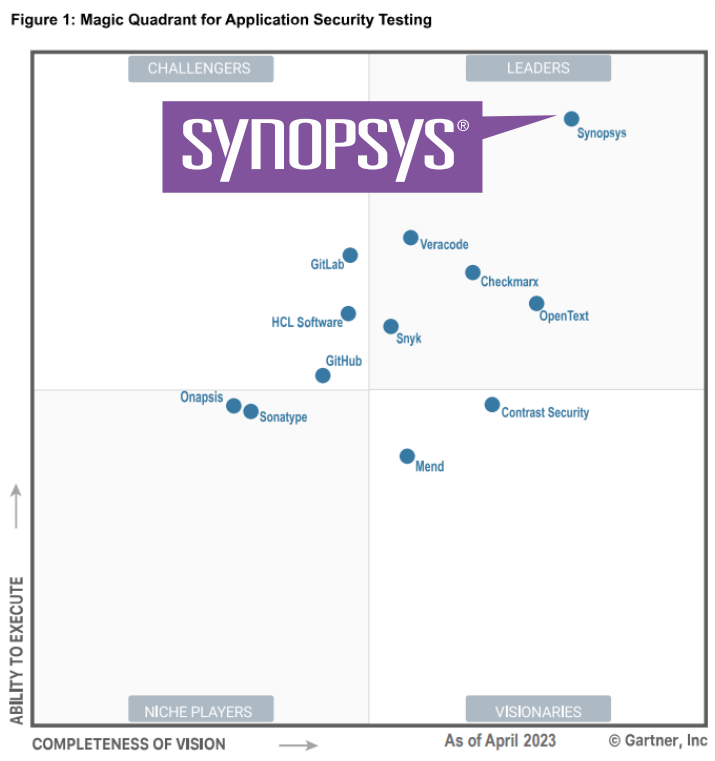
The Synopsys AppSec Portfolio



We’re the recognized leader in Application Security Testing

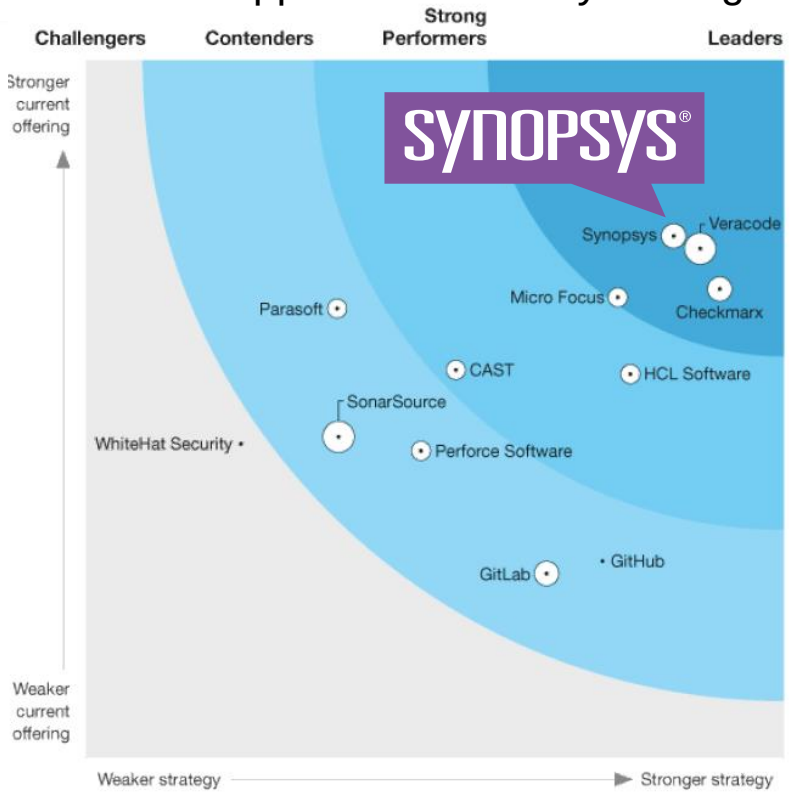
Gartner®

Magic Quadrant for
Application Security Testing

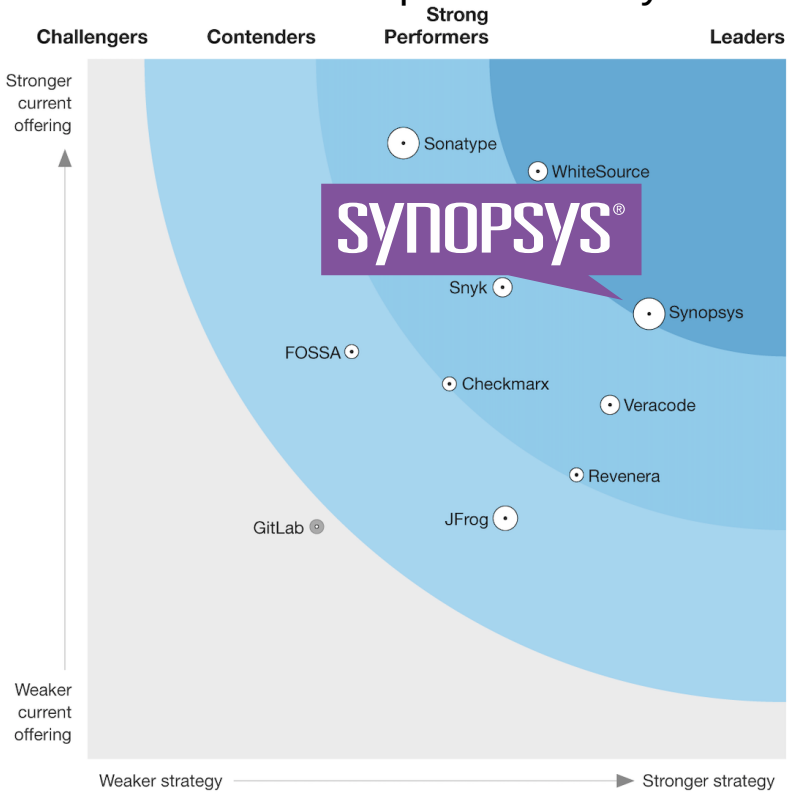


FORRESTER

Forrester Wave™:
Static Application Security Testing



Forrester Wave™:
Software Composition Analysis



Source: Gartner (May 2023)

What is SAST?

What is SAST ?

- It stands for Static Application Security Testing.
- Analyze an application’s source code to identify vulnerabilities without executing the program.
- Analyze various architectures and technologies including web applications , mobile applications etc.
- Analysis requires availability of code that can be compiled.

Static vs Dynamic Analysis

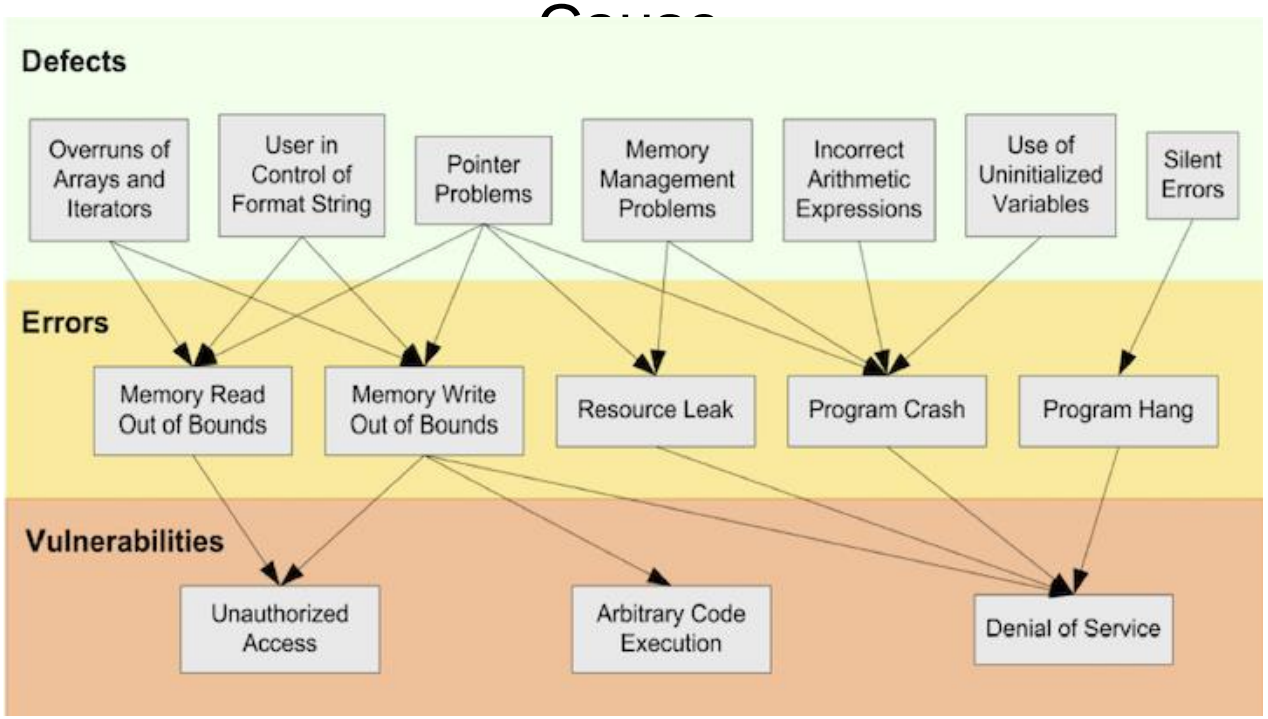
Static and Dynamic Analysis are complementary techniques which each excel in areas that are challenging to the other.

- **Static Analysis**
  - A method of examining software at compile time, reporting potential defects
- **Dynamic Analysis**
  - A method of examining software at runtime, by executing the code over multiple test cases

Managing Software Integrity

- Software defects are expensive to find and repair, especially when they get shipped to customers
- Defects are caused by code complexity, code size, delivery pressure, rapid changes, etc.
- Many defects are simply *patterns* in source code that can be identified and removed during development

Security Issues v Root Causes



What does Coverity do?

Coverity analyzes your source code, and pinpoints:

- coding errors
- security flaws
- Lines, files and functions, insufficiently tested

8. **alloc_fn:** Storage is returned from allocation function "fopen(signed char const *, signed char const *)".
9. **var_assign:** Assigning: "newfp" = storage returned from "fopen(stderrPath, "w")".

```
        newfp = fopen(stderrPath, TEXT("w"));
```

10. Condition "newfp", taking true branch

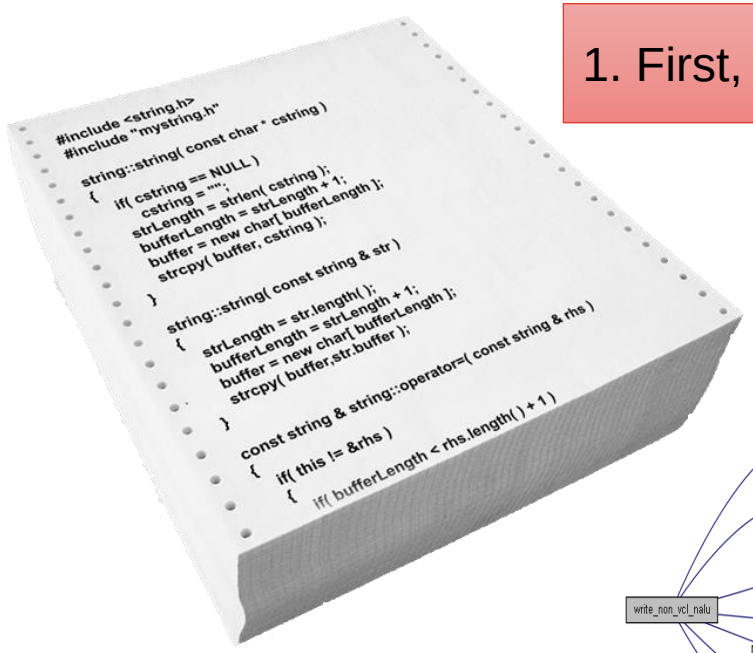
```
        if ( newfp ) {
            *stderr = *newfp;
        }

    setvbuf(stdout, NULL, _IOLBF, BUFSIZ);  /* Line buffered */
    setbuf(stderr, NULL);                   /* No buffering */
    stdioRedirectEnabled = 1;
```

CID 27154 (#3 of 6): Resource leak (RESOURCE_LEAK)

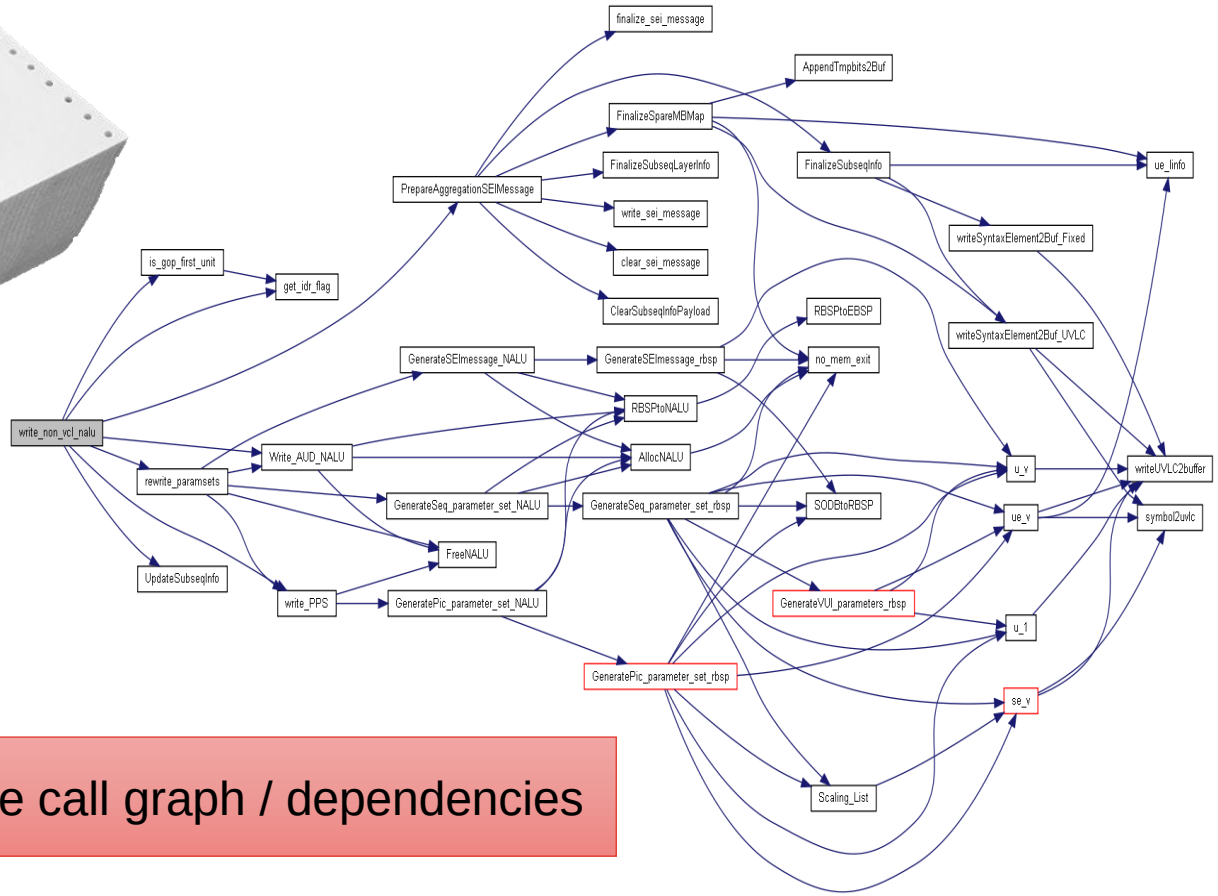
11. **leaked_storage:** Variable "newfp" going out of scope leaks the storage it points to.

```
}
```

How does Coverity do this?



1. First, scans your code



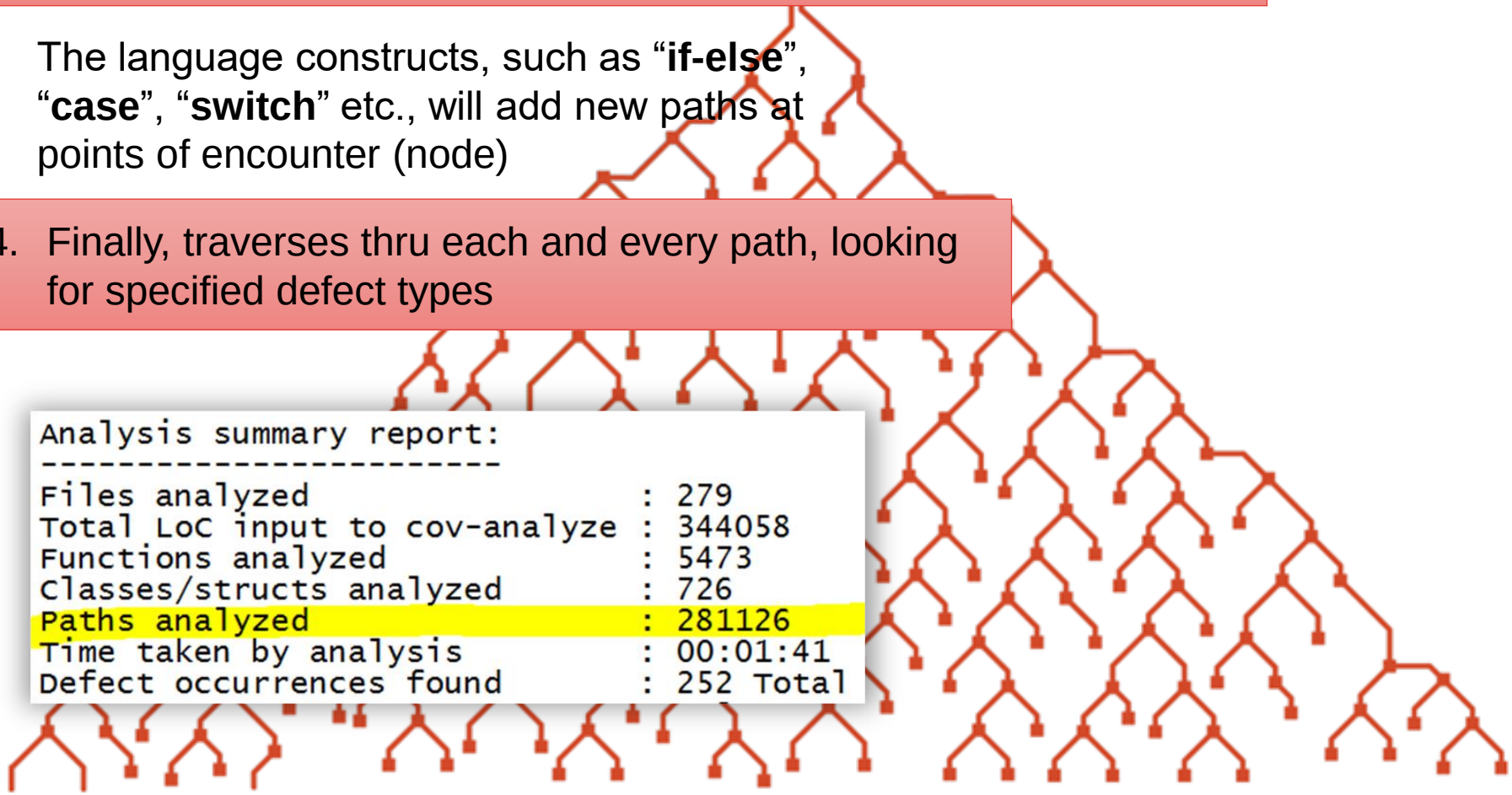
2. Then, calculates the call graph / dependencies

All Possible Paths

3. Next computes the fan-out for all the possible paths thru your code

- The language constructs, such as “**if-else**”, “**case**”, “**switch**” etc., will add new paths at points of encounter (node)

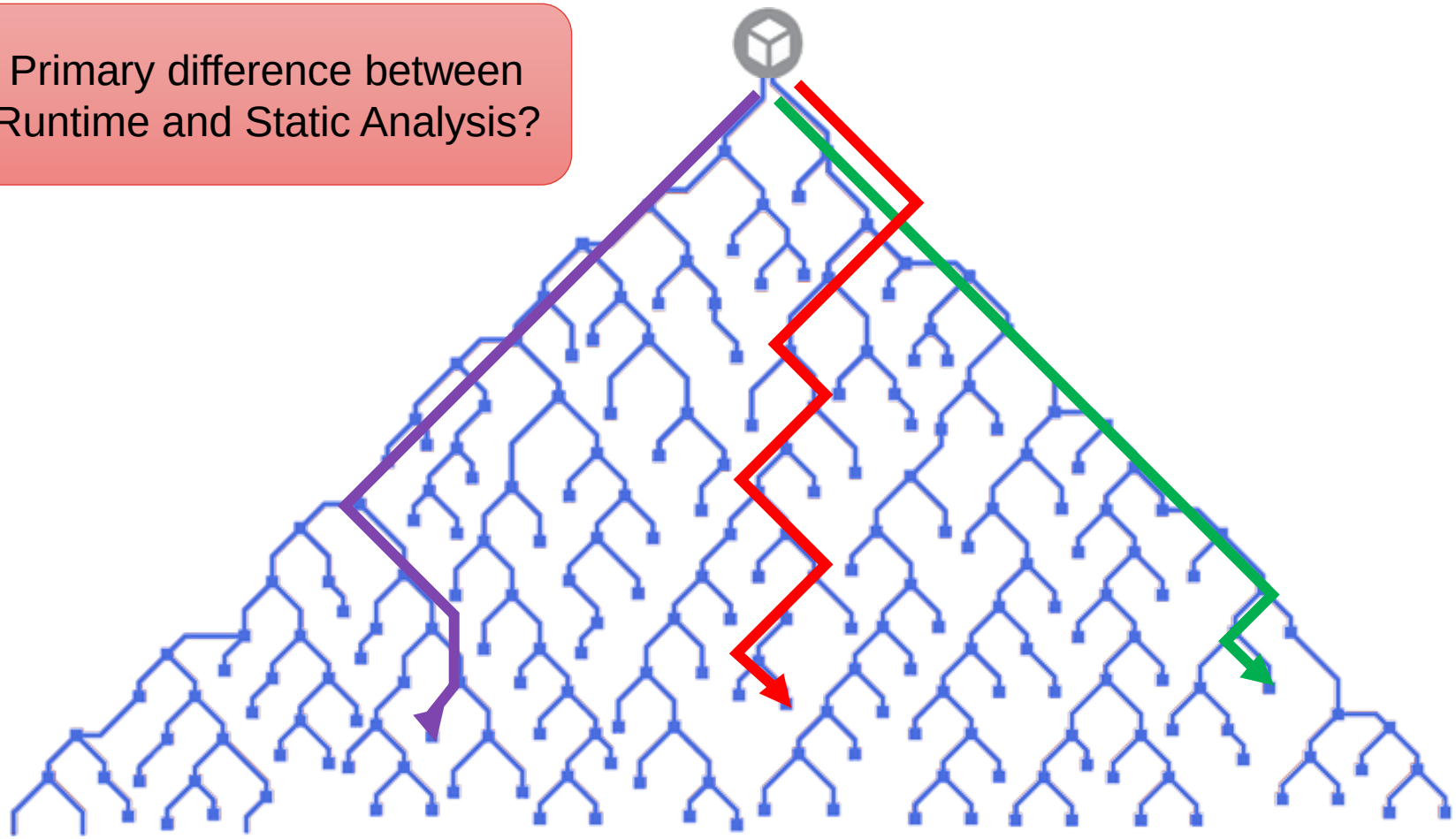
4. Finally, traverses thru each and every path, looking for specified defect types



```
Analysis summary report:
------------------------
Files analyzed                  : 279
Total LoC input to cov-analyze  : 344058
Functions analyzed              : 5473
Classes/structs analyzed        : 726
Paths analyzed                  : 281126
Time taken by analysis          : 00:01:41
Defect occurrences found        : 252 Total
```

Runtime vs. Static

Primary difference between Runtime and Static Analysis?



Static Analysis Strengths

- Early detection
  - Can find defects before a working executable is ready
- No test cases
  - Test cases not required – all paths automatically analyzed
- Fast analysis of very large codebases
  - Analyze millions of lines of code in a few hours
- Deterministic
  - Analysis of the same codebase = same results

What types of issues are detected?

- Coverity Analysis pinpoints defects in your code in myriad of categories, such as:
  - Memory corruption, resource leaks
  - NULL object/pointer dereferences
  - Thread Concurrency (Deadlock, Race Condition, etc.)
  - Security
  - Logic errors
  - Incorrect program behavior
  - Web Application security flaws
  - Lines, files and functions, insufficiently tested (TA.de)

Introducing Synopsys static analysis

Synopsys static analysis solutions give you fast, scalable, and comprehensive detection of security and quality defects on premises, in the cloud, and at the developer desktop

Coverity

In-depth static analysis for secure code that’s free of defects

Comprehensive Analysis

- Identifies both code quality and security issues
- Uncovers complex issues that span files and libraries
- Broad and deep language and framework support
- Standard Compliance

Developer Velocity

- Accurate results let developers focus on business value
- Fast scans on PR/MR to find defects early in the SDLC
- Easy triage and prioritization of results

Enterprise Scale

- Thousands of developers
- Thousands of projects
- Apps with millions of lines of code

Broad standards compliance and SDLC integrations

Broad standards compliance and SDLC integrations

Uncover vulnerabilities across all your languages

Broad and deep support for more than 20 languages

Language Coverage

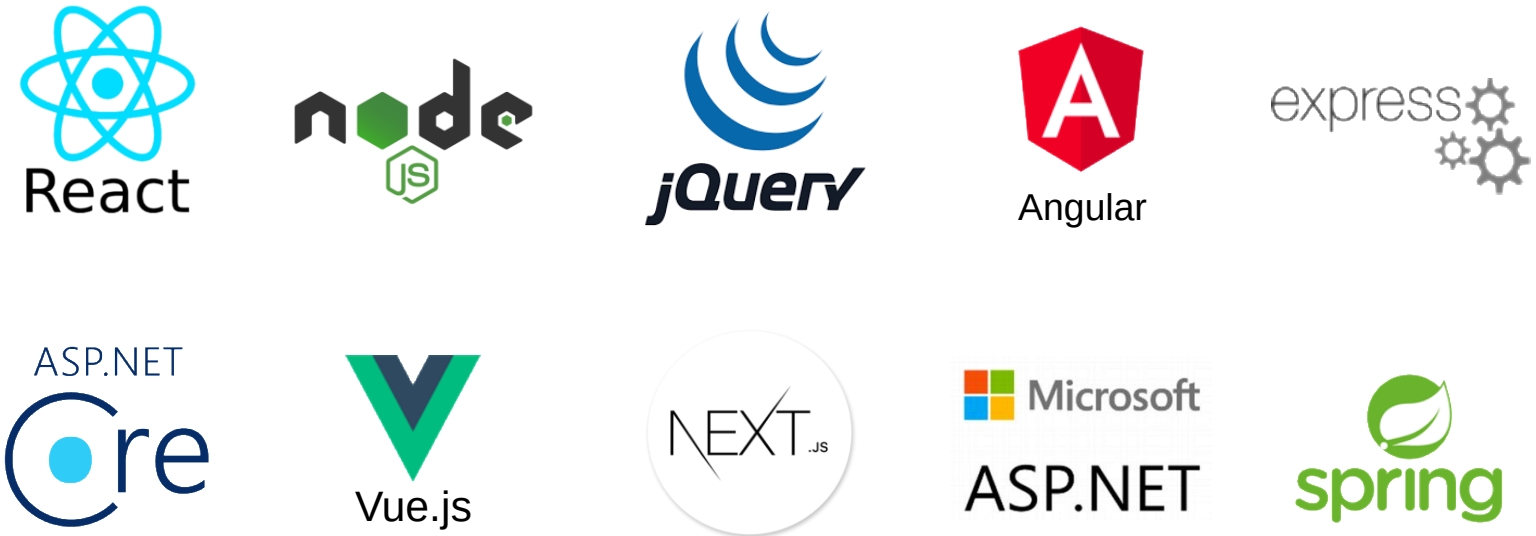
Support for Popular Compilers

- ARM C/C++
- Clang
- GNU GCC/G++
- Intel C++ for Windows
- JDK for Mac OS X
- Microsoft Visual C++
- Nvidia CUDA (NVCC)
- Renesas C/C++
- SONY PS4 SDK
- Sun (Oracle) CC
- Sun/Oracle JDK
- Wind River C/C++

Analyze the context of code usage and frameworks

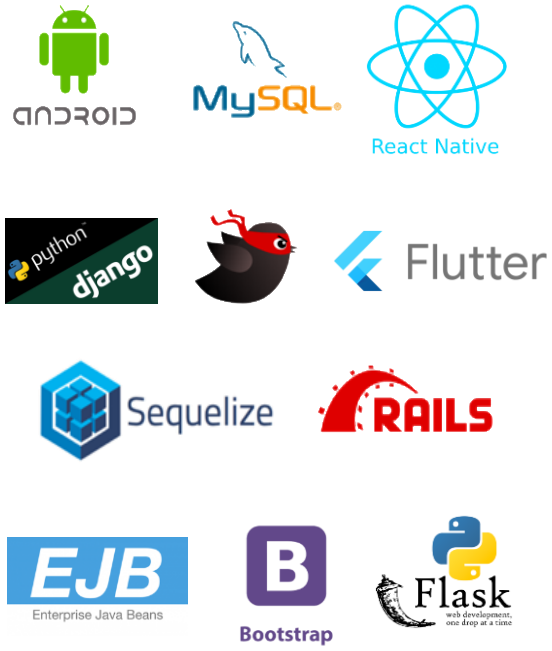
Poorly understood frameworks and APIs lead to false positives and missed vulnerabilities

10 Most Popular Web Application Frameworks



Source: *Stack Overflow, [2023 Developer Survey] (90,000+ responses)*

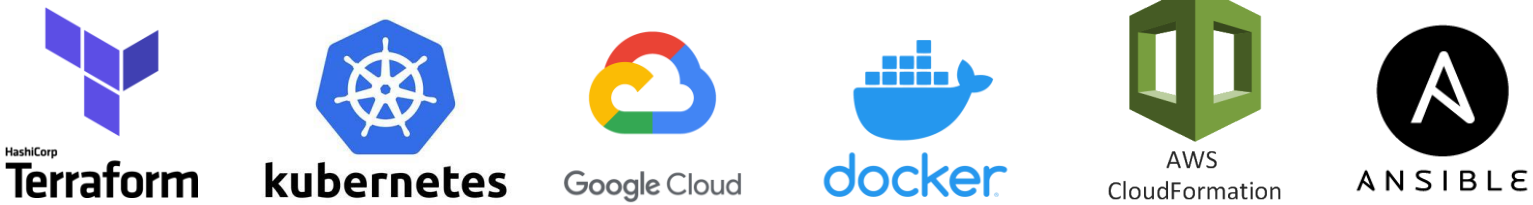
Support for 200+ Frameworks in All



Deploy cloud environments with confidence

IaC scans identify misconfigurations to prevent vulnerabilities in cloud environments

Infrastructure-as-Code



File formats:

JSON	YAML	HCL	HTML	XML	plist
TOML	Properties	Vue template	JSX	TSX	

Supports CIS Benchmarks for Cloud Environments

- Access controls
- Data protection
- Hardcoded secrets
- Network settings
- Infrastructure configuration
- Software settings

CIS Center for Internet Security®
Creating Confidence in the Connected World.

The Coverity advantage

Finds meaningful, actionable defects with a low false-positive rate

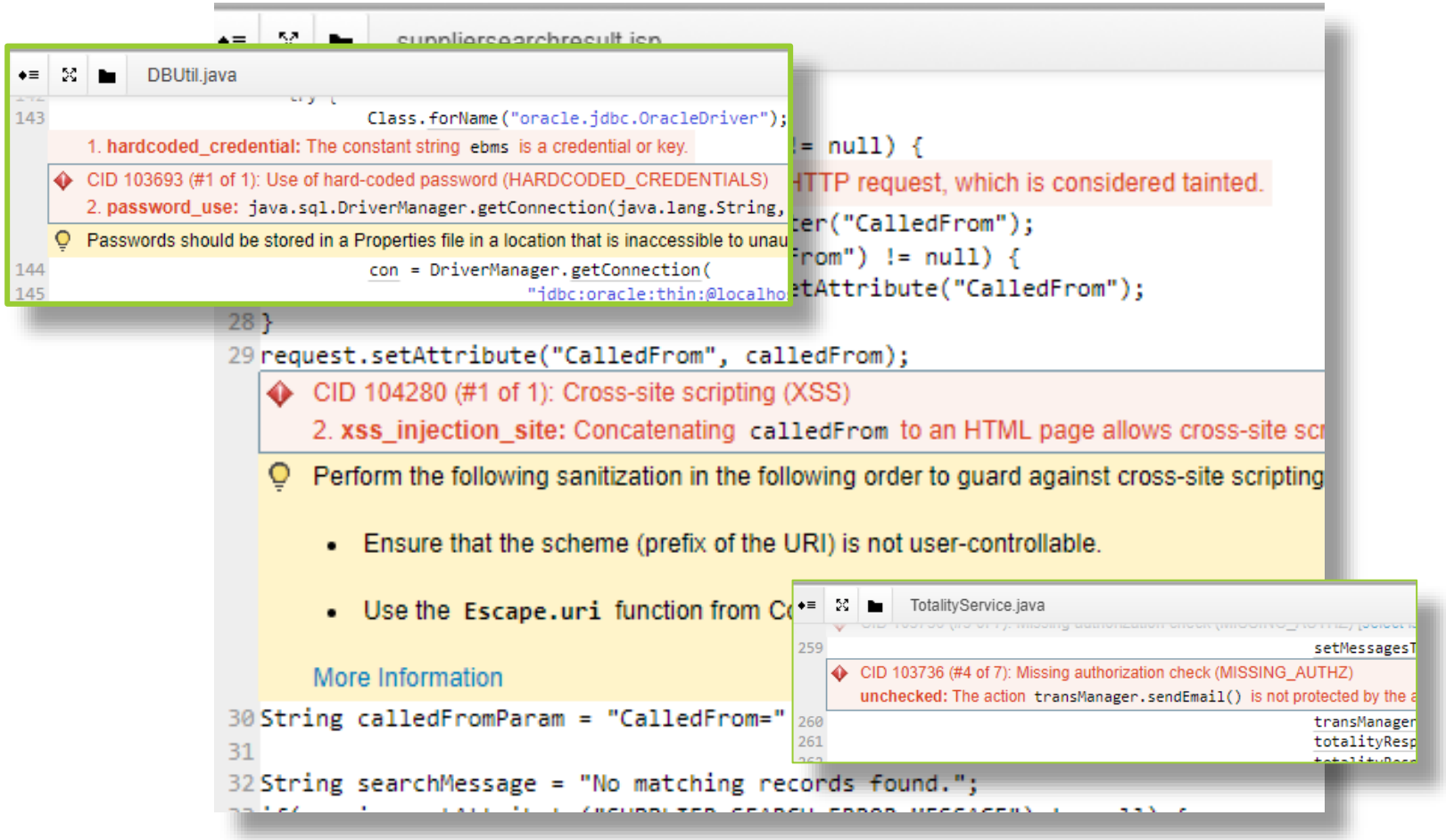
Comprehensive (security) coverage of all common security standards:

Web and Enterprise applications

- OWASP Top-10
- CWE Top-25
- PCI DSS

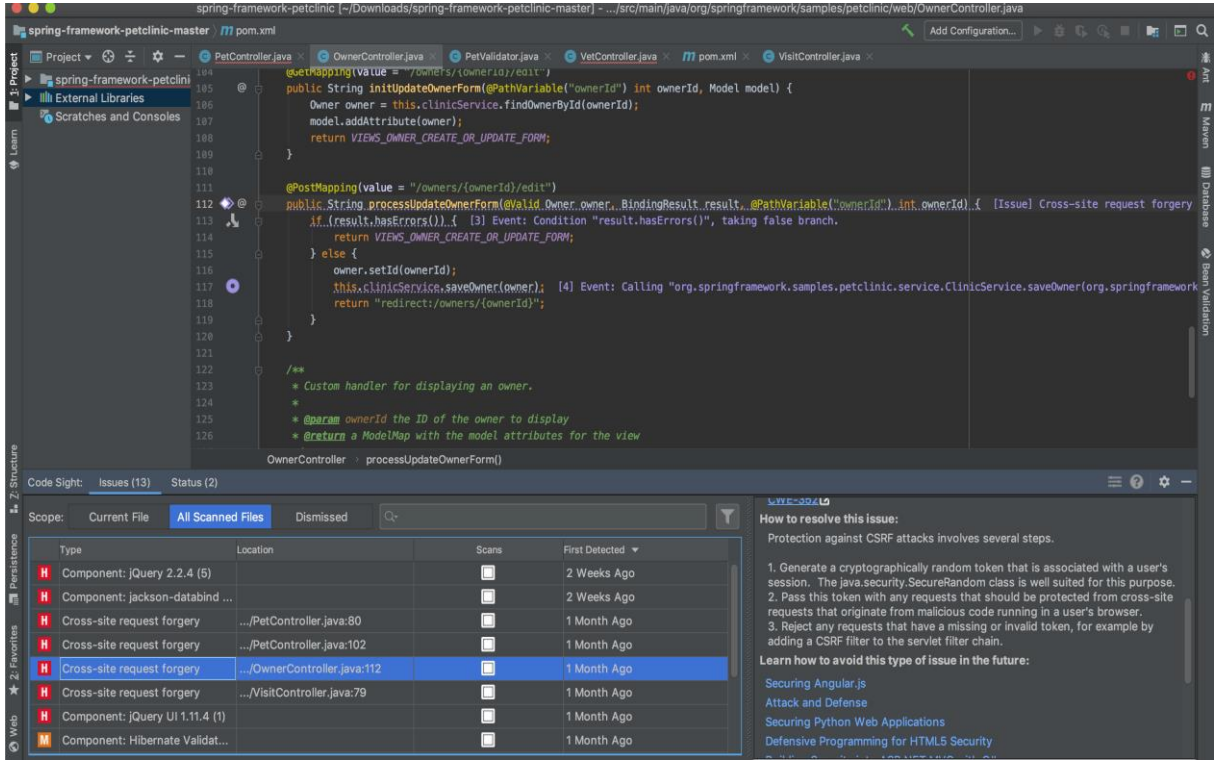
Mobile applications

- OWASP Mobile Top-10
- Mobile platforms
  - iOS (Swift)
  - Android (Java, Kotlin)



Code Sight

Secure code as it’s being written



Visual Studio VS Code IntelliJ eclipse

16 Total IDE Integrations

Catch defects before code commit

- Combines static analysis and SCA to detect security, quality, and compliance issues in <u>both</u> proprietary code and open source dependencies

Improve developer productivity

- Integrated into IDE. No need to switch tools.
- Detailed remediation guidance minimizes time to fix.
- Reduces costly context switching to address issues found later in the build/test pipeline

Help developers write better code

- Built-in context-based eLearning delivers the right training at the right time so developers learn how to write more secure, higher-quality code

Live Demo

Thank You