Build Security In With Coverity

Give Developers Early Feedback to Identify Security Issues

Ashutosh Kumar, Product Marketing Manager

James Croall, Director, Technical Product Management

11/5/2020

Topics Covered

Evolving AppSec paradigm

- Importance of onboarding developers

Feedback mechanism in the CI pipeline

- Feedback in the IDE
- Feedback in a pull request

Evolving AppSec Paradigm

The Tussle

“The only thing more dangerous than a developer is a developer conspiring with security. The two working together gives us means, motive, and opportunity.”

The Phoenix Project

“Compliance is not optional. It’s the law. My job is to keep them out of orange jumpsuits, and so I did what I had to do.”

The Phoenix Project

From the authors of The Visible Ops Handbook
The Phoenix Project
A Novel About IT, DevOps, and Helping Your Business Win
Gene Kim, Kevin Behr, and George Spafford

The Perception Around Security

Governances, rules, and gates

Optimize for risk

Waterfall inspired, out-of-band

Highly centralized

"In any value stream, there is always a direction of flow, and there is always one and only constraint; any improvement not made at that constraint is an illusion."

The Phoenix Project

The Evolution in Security

Engineering-led

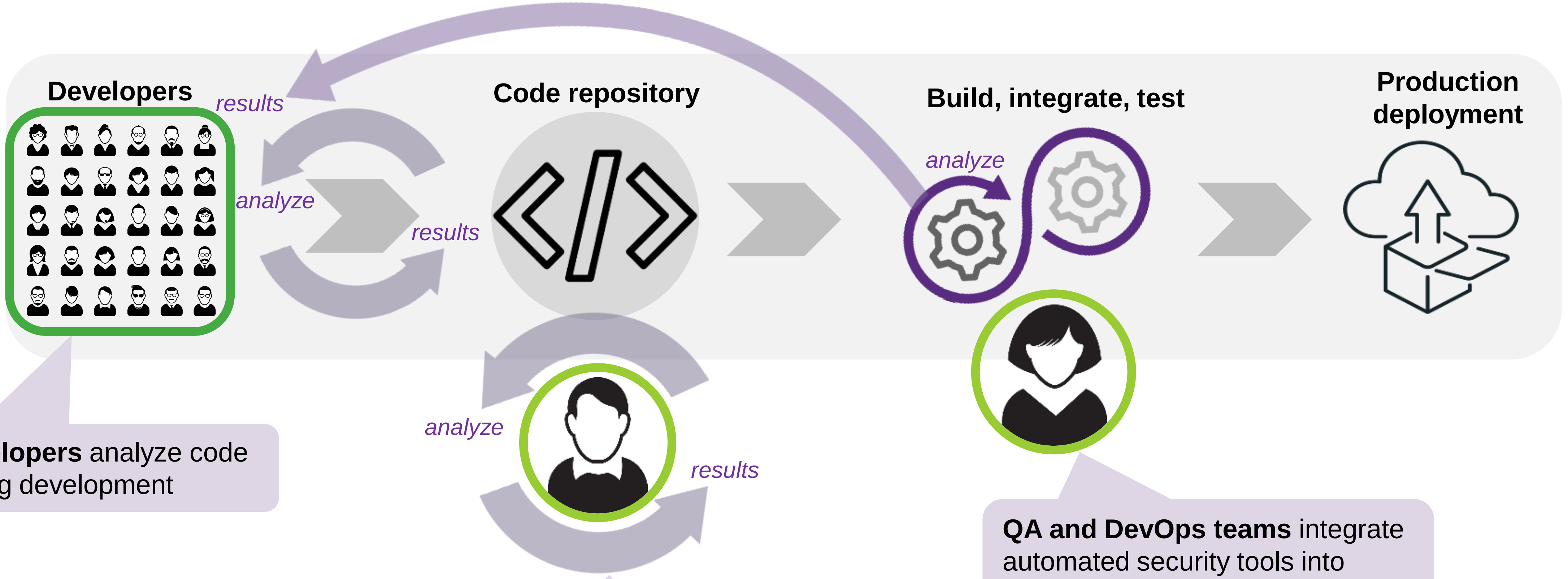
Feature velocity–focused

Optimized for business and risk

Metric-driven, fast, in-line; everyone is responsible

Decentralized security

The Security Practices Today



Developers analyze code during development

Security teams empowered to independently scan projects for security dashboards / audits

QA and DevOps teams integrate automated security tools into CI/CD pipelines

Who Needs Static Analysis?

Security executive

Development executive

DevOps manager

Developer

Coverity: SAST Solution for Your Needs

Comprehensive

Accurate

Integrated

Fast

Intelligent

Coverity Capabilities

Language support

Fortran
.NET Core
Microsoft ASP.net
node JS
Objective-C
JS
php
python
C/C++
Java
C# .NET develop
Scala
GO
Microsoft VB.net
Kotlin
RAILS
TS

Security guidelines

OWASP Mobile Security Project
CWE
CWE Top 25 Most Dangerous Software Errors
OWASP Open Web Application Security Project
CISQ Consortium for Information & Software Quality™

Standards compliance

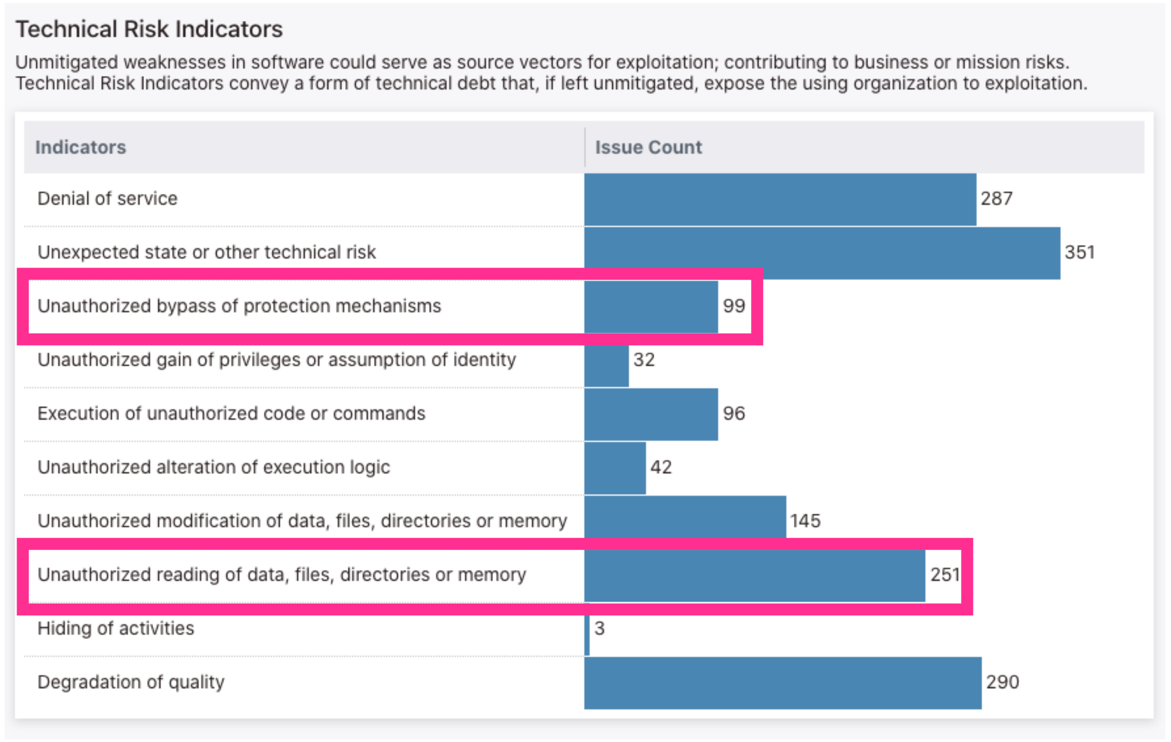
CERT
ISO 26262
DISA
PCi
AUTOSAR
MISRA
NVIDIA CUDA
CISQ Consortium for Information & Software Quality™

SDLC workflow

Android Studio
IntelliJIDEA
JIRA
Bugzilla
Jenkins
eclipse
Visual Studio
git

CISQ Consortium for Information & Software Quality™ & SYNOPSYS® Silicon to Software™

- Coverity is compliant with CISQ's new Automated Source Code Data Protection Measure
  - Coverity covers all of the 36 parent weaknesses in the ASCDP Measure
  - The Synopsys Polaris platform aggregates weaknesses by Technical Risk Indicators, including those representing non-compliance with data protection guidelines



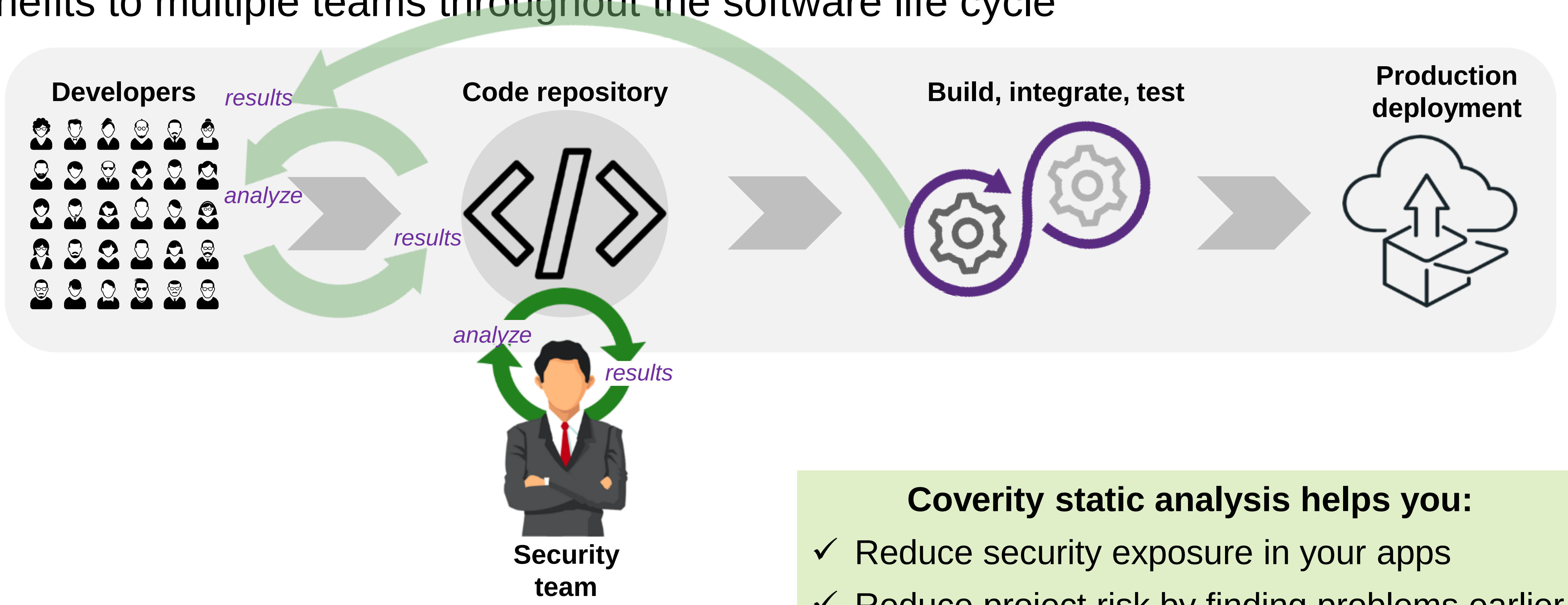
Technical Risk Indicators

Unmitigated weaknesses in software could serve as source vectors for exploitation; contributing to business or mission risks. Technical Risk Indicators convey a form of technical debt that, if left unmitigated, expose the using organization to exploitation.

Indicators	Issue Count
Denial of service	287
Unexpected state or other technical risk	351
Unauthorized bypass of protection mechanisms	99
Unauthorized gain of privileges or assumption of identity	32
Execution of unauthorized code or commands	96
Unauthorized alteration of execution logic	42
Unauthorized modification of data, files, directories or memory	145
Unauthorized reading of data, files, directories or memory	251
Hiding of activities	3
Degradation of quality	290

Static Analysis for Security Teams

Benefits to multiple teams throughout the software life cycle



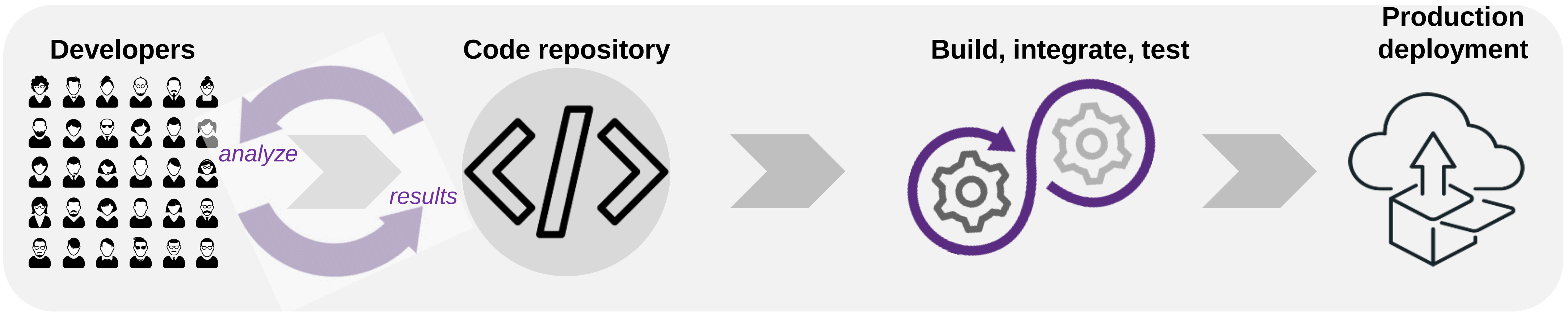
Coverity static analysis helps you:

- ✓ Reduce security exposure in your apps
- ✓ Reduce project risk by finding problems earlier
- ✓ Lower your software development costs

$ Cost of fixing vulnerabilities **$$** **$$$**

Static Analysis Helping Developers

Benefits to multiple teams throughout the software life cycle



Developers value:

- Accurate, actionable findings—bugs and vulnerabilities
- Easily integrated, unobtrusive tools
- Focus on fixing defects, instead of triaging and managing tool integrations

Coverity strengths:

✓ Large library of checkers > depth of analysis

✓ High accuracy findings (low FP rate)

✓ Fast, comprehensive analysis during development

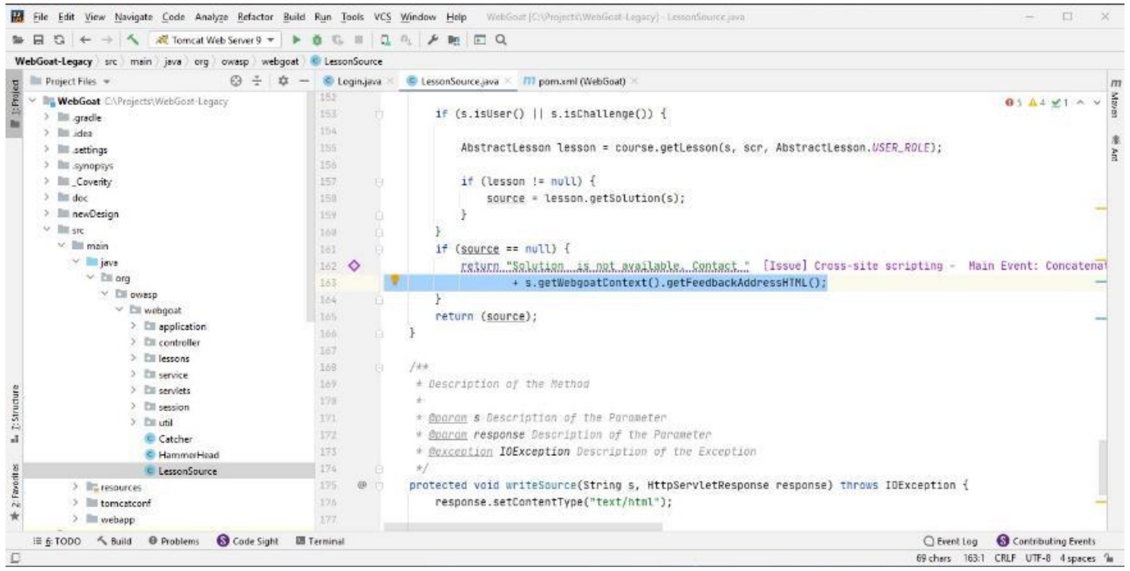
✓ Integrated into popular IDEs

Build Feedback Mechanism in CI Pipeline

Advantages of Shifting Left

- Reduce security issues going into production code
- Lower cost of fixing issues = Better ROI
- Improve application robustness and process agility
- Improved application security = Preserved company reputation

Feedback to Developers



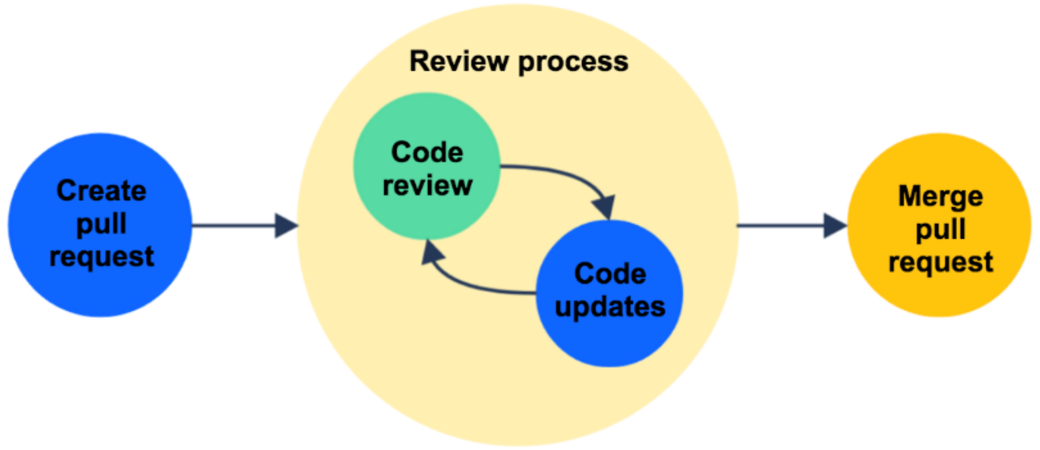
```
    if (s.isUser() || s.isChallenge()) {

        AbstractLesson lesson = course.getLesson(s, scr, AbstractLesson.USER_ROLE);

        if (lesson != null) {
            source = lesson.getSolution(s);
        }
    }
    if (source == null) {
        return "Solution  is not available. Contact "  [Issue] Cross-site scripting -  Main Event: Concatenat
                + s.getWebgoatContext().getFeedbackAddressHTML();
    }
    return (source);
}

/**
 * Description of the Method
 *
 * @param s Description of the Parameter
 * @param response Description of the Parameter
 * @exception IOException Description of the Exception
 */
protected void writeSource(String s, HttpServletResponse response) throws IOException {
    response.setContentType("text/html");
```

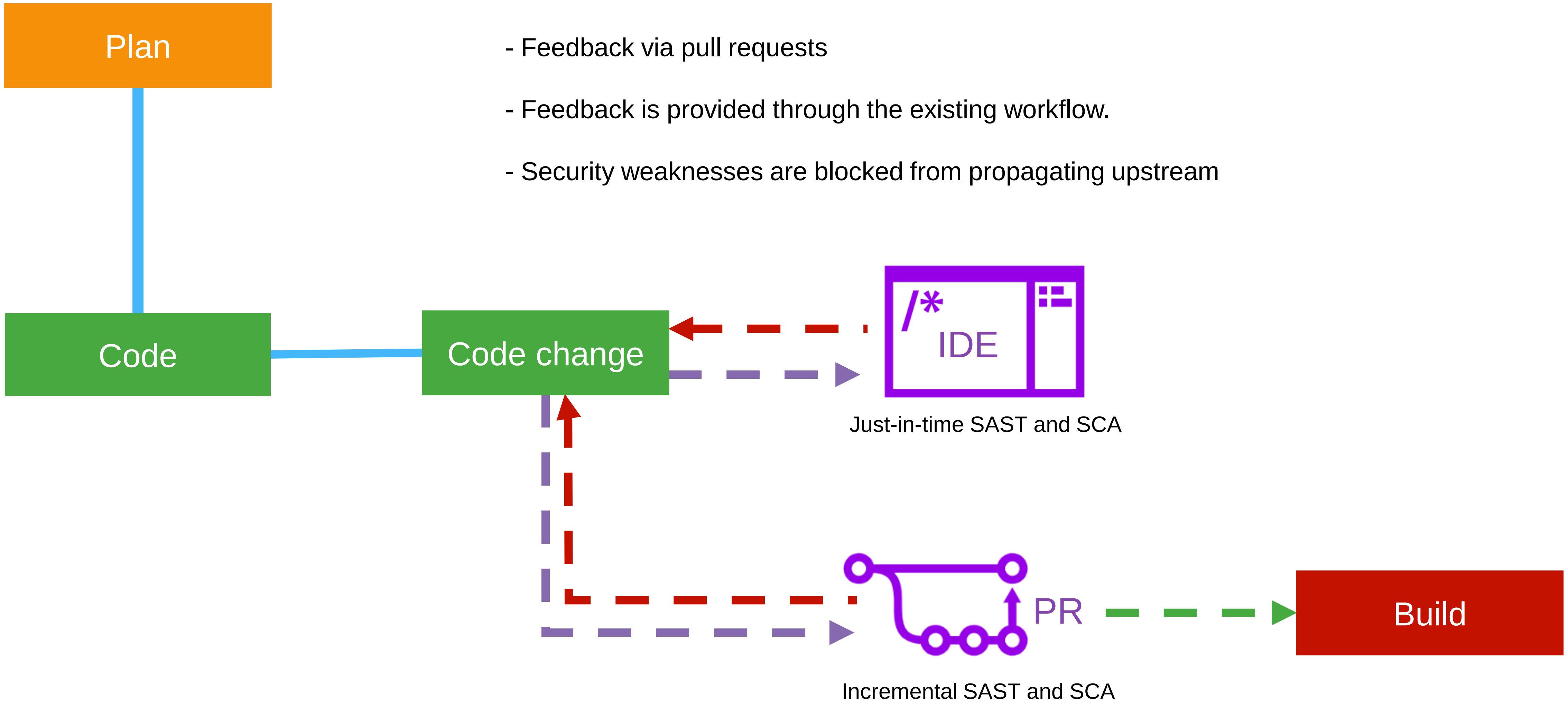
Feedback in the IDE as the code is being written



Feedback on pull requests as the code is being merged

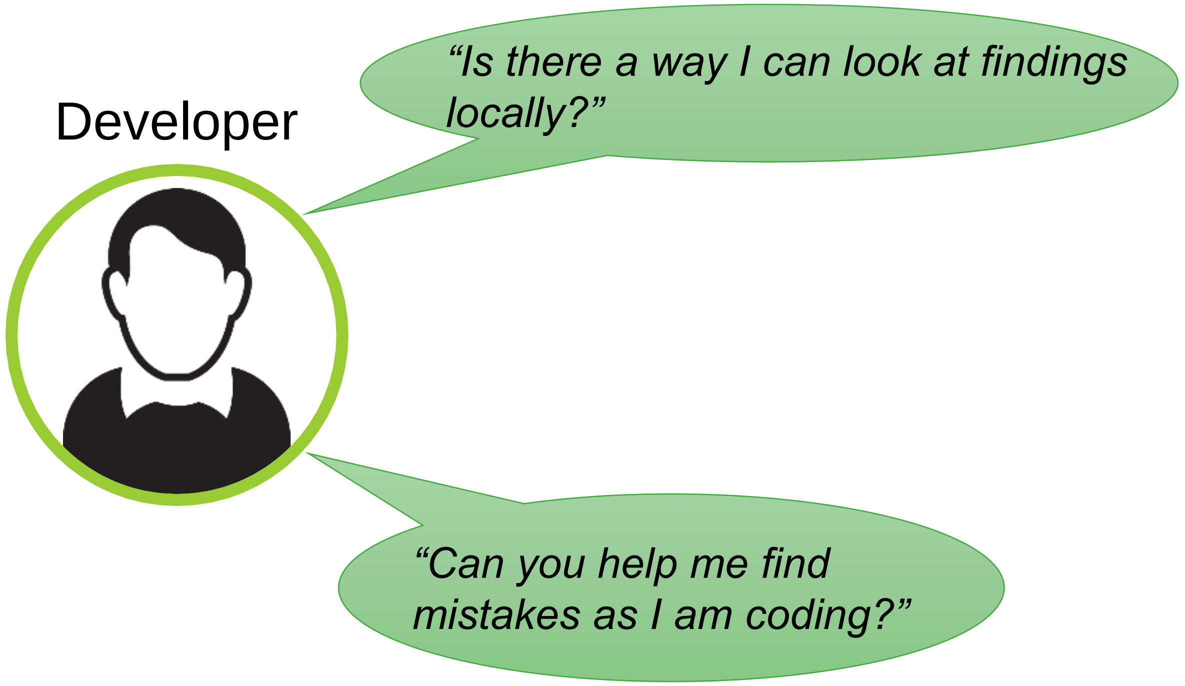
Code Data Flow

- Just-in-time feedback to developers
- Feedback via pull requests
- Feedback is provided through the existing workflow.
- Security weaknesses are blocked from propagating upstream



Feedback in the IDE

Synopsys Code Sight™


Developer
"Is there a way I can look at findings locally?"
"Can you help me find mistakes as I am coding?"

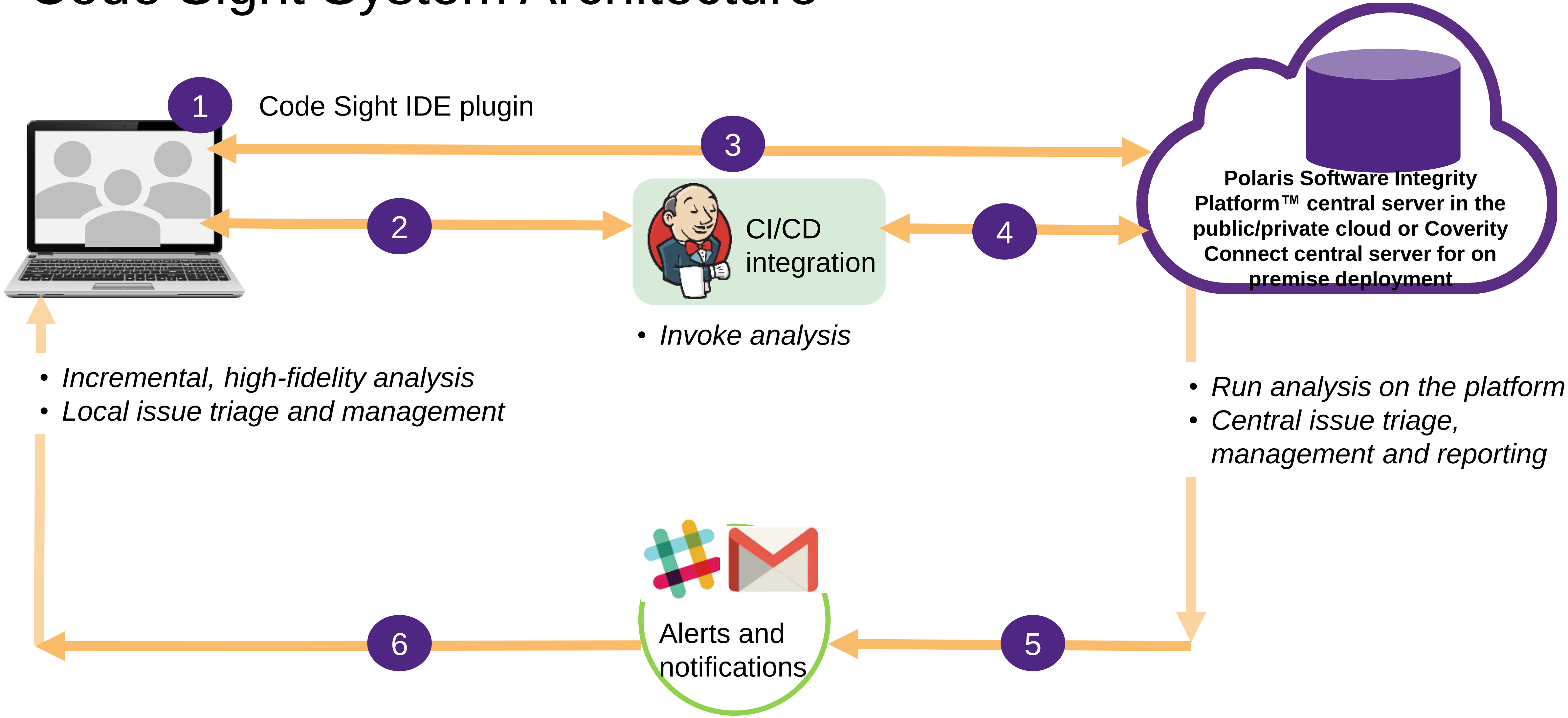
The Code Sight IDE plug-in

- Helps development teams shift left
- Helps developers find and fix issues before checking in code
- Simplifies—Just open a file
- Scans for thousands of different potential problems as you code

Feedback in the IDE

- Code Sight system architecture
- Code Sight supported IDEs
- Coverity® with Code Sight
- Live demo

Code Sight System Architecture

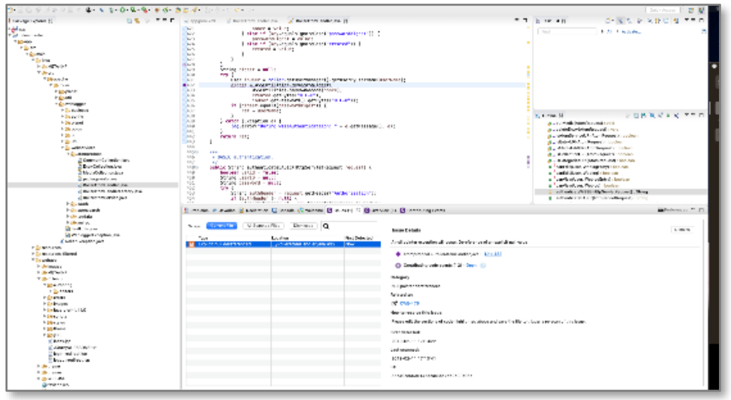

1
Code Sight IDE plugin
3
2
CI/CD integration
4
Polaris Software Integrity Platform™ central server in the public/private cloud or Coverity Connect central server for on premise deployment
Alerts and notifications
6
5

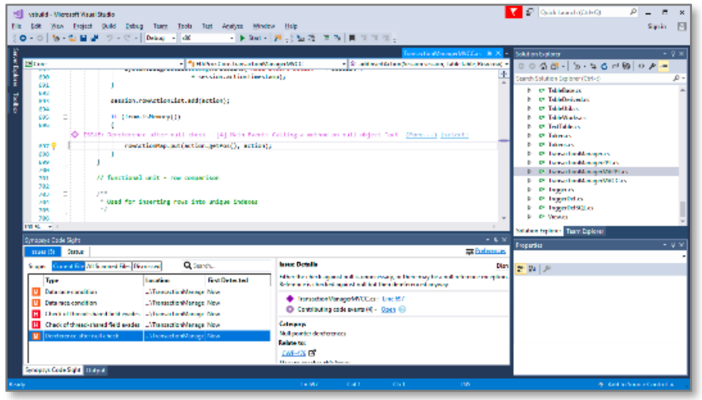
- *Invoke analysis*

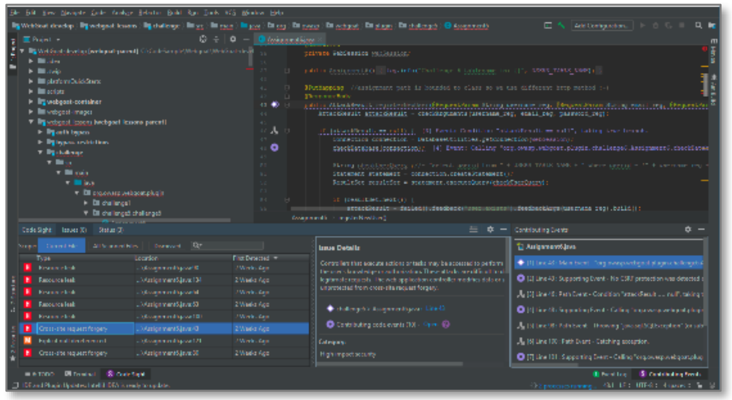
- *Incremental, high-fidelity analysis*
- *Local issue triage and management*

- *Run analysis on the platform*
- *Central issue triage, management and reporting*

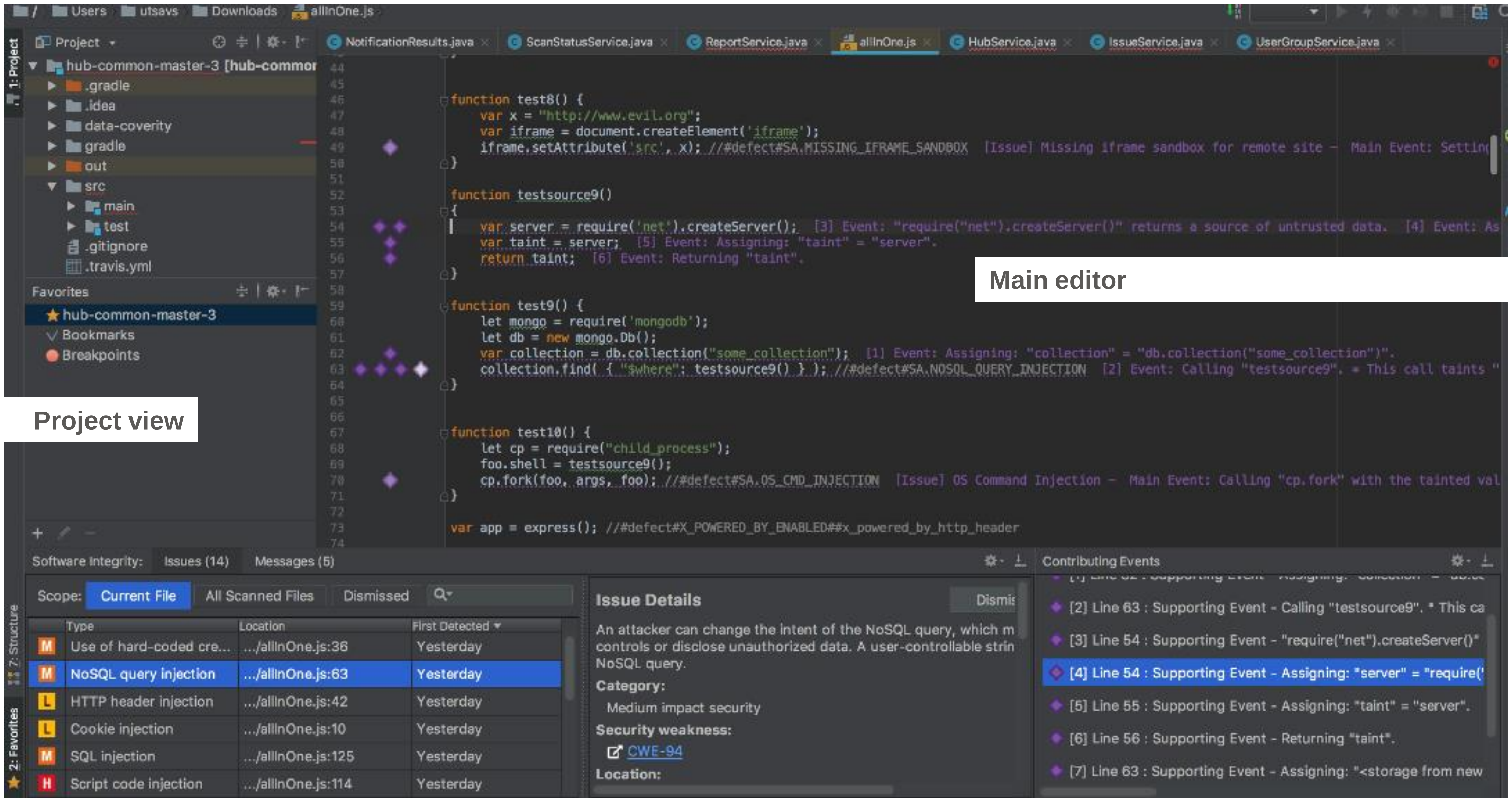
Code Sight Supported IDEs


Eclipse and compatible IDEs


Microsoft Visual Studio


JetBrains IDEs such as IntelliJ

Coverity With Code Sight



```
function test8() {
    var x = "http://www.evil.org";
    var iframe = document.createElement('iframe');
    iframe.setAttribute('src', x); //#defect#SA.MISSING_IFRAME_SANDBOX  [Issue] Missing iframe sandbox for remote site -  Main Event: Setting
}

function testsource9()
{
    var server = require('net').createServer();  [3] Event: "require("net").createServer()" returns a source of untrusted data.  [4] Event: As
    var taint = server;  [5] Event: Assigning: "taint" = "server".
    return taint;  [6] Event: Returning "taint".
}

function test9() {
    let mongo = require('mongodb');
    let db = new mongo.Db();
    var collection = db.collection("some_collection");  [1] Event: Assigning: "collection" = "db.collection("some_collection")".
    collection.find( { "$where": testsource9() } ); //#defect#SA.NOSQL_QUERY_INJECTION  [2] Event: Calling "testsource9". * This call taints "
}


function test10() {
    let cp = require("child_process");
    foo.shell = testsource9();
    cp.fork(foo, args, foo); //#defect#SA.OS_CMD_INJECTION  [Issue] OS Command Injection -  Main Event: Calling "cp.fork" with the tainted val
}

var app = express(); //#defect#X_POWERED_BY_ENABLED##x_powered_by_http_header
```

Software Integrity: Issues (14) Messages (5)

Scope: Current File | All Scanned Files | Dismissed

	Type	Location	First Detected
M	Use of hard-coded cre...	.../allInOne.js:36	Yesterday
M	NoSQL query injection	.../allInOne.js:63	Yesterday
L	HTTP header injection	.../allInOne.js:42	Yesterday
L	Cookie injection	.../allInOne.js:10	Yesterday
M	SQL injection	.../allInOne.js:125	Yesterday
H	Script code injection	.../allInOne.js:114	Yesterday

Issue Details Dismis

An attacker can change the intent of the NoSQL query, which m
controls or disclose unauthorized data. A user-controllable strin
NoSQL query.

Category:
Medium impact security

Security weakness:
CWE-94

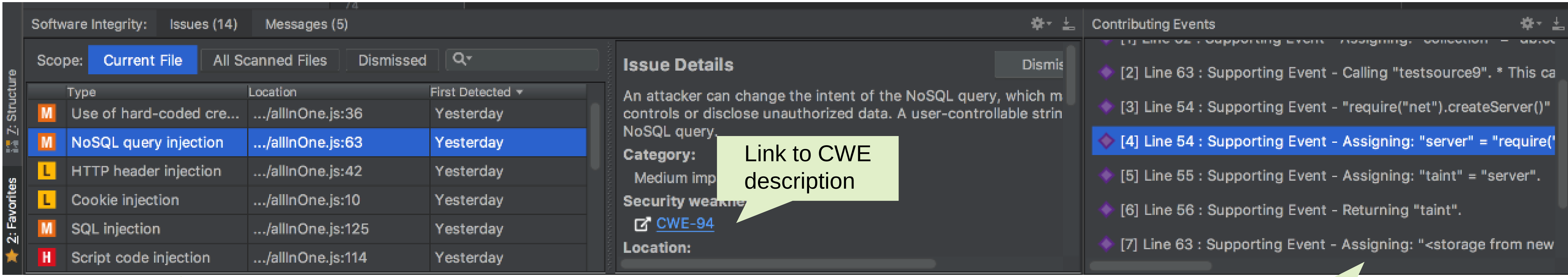
Location:

Contributing Events

- [2] Line 63 : Supporting Event - Calling "testsource9". * This ca
- [3] Line 54 : Supporting Event - "require("net").createServer()"
- [4] Line 54 : Supporting Event - Assigning: "server" = "require('
- [5] Line 55 : Supporting Event - Assigning: "taint" = "server".
- [6] Line 56 : Supporting Event - Returning "taint".
- [7] Line 63 : Supporting Event - Assigning: "<storage from new

Coverity With Code Sight

Crisp issues view



Software Integrity: Issues (14) Messages (5)

Scope: Current File | All Scanned Files | Dismissed

	Type	Location	First Detected
M	Use of hard-coded cre...	.../allInOne.js:36	Yesterday
M	NoSQL query injection	.../allInOne.js:63	Yesterday
L	HTTP header injection	.../allInOne.js:42	Yesterday
L	Cookie injection	.../allInOne.js:10	Yesterday
M	SQL injection	.../allInOne.js:125	Yesterday
H	Script code injection	.../allInOne.js:114	Yesterday

Issue Details

An attacker can change the intent of the NoSQL query, which m
controls or disclose unauthorized data. A user-controllable strin
NoSQL query.

Category:
Medium imp

Security weakne

CWE-94

Location:

Contributing Events

[2] Line 63 : Supporting Event - Calling "testsource9". * This ca
[3] Line 54 : Supporting Event - "require("net").createServer()"
[4] Line 54 : Supporting Event - Assigning: "server" = "require("
[5] Line 55 : Supporting Event - Assigning: "taint" = "server".
[6] Line 56 : Supporting Event - Returning "taint".
[7] Line 63 : Supporting Event - Assigning: "<storage from new

Link to CWE description

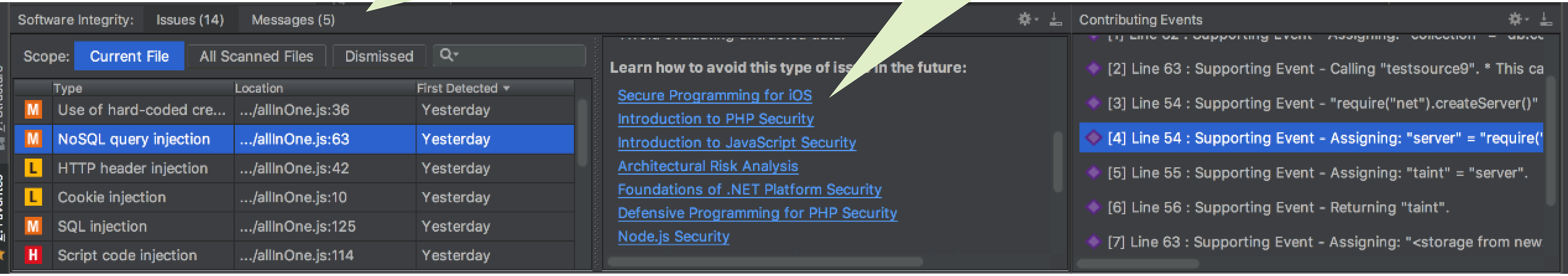
Prioritized vulnerabilities by category

Triage and dismiss vulnerabilities

Scroll down

Related eLearning courses

Dataflow view: main and supporting events



Software Integrity: Issues (14) Messages (5)

Scope: Current File | All Scanned Files | Dismissed

	Type	Location	First Detected
M	Use of hard-coded cre...	.../allInOne.js:36	Yesterday
M	NoSQL query injection	.../allInOne.js:63	Yesterday
L	HTTP header injection	.../allInOne.js:42	Yesterday
L	Cookie injection	.../allInOne.js:10	Yesterday
M	SQL injection	.../allInOne.js:125	Yesterday
H	Script code injection	.../allInOne.js:114	Yesterday

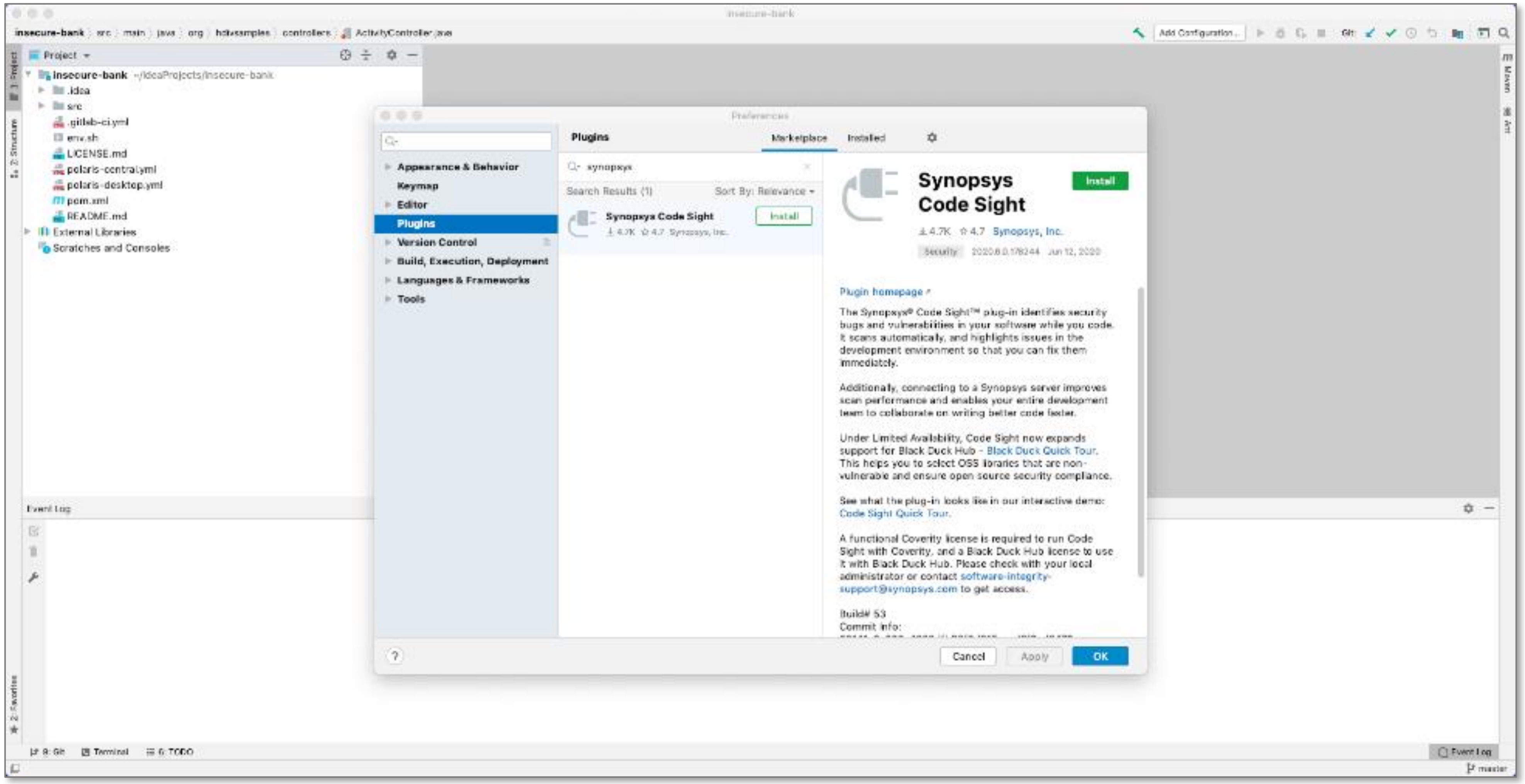
Learn how to avoid this type of issue in the future:

- Secure Programming for iOS
- Introduction to PHP Security
- Introduction to JavaScript Security
- Architectural Risk Analysis
- Foundations of .NET Platform Security
- Defensive Programming for PHP Security
- Node.js Security

Contributing Events

[2] Line 63 : Supporting Event - Calling "testsource9". * This ca
[3] Line 54 : Supporting Event - "require("net").createServer()"
[4] Line 54 : Supporting Event - Assigning: "server" = "require("
[5] Line 55 : Supporting Event - Assigning: "taint" = "server".
[6] Line 56 : Supporting Event - Returning "taint".
[7] Line 63 : Supporting Event - Assigning: "<storage from new

LIVE Demonstration: Feedback With Code Sight



Synopsys Code Sight

4.7K ☆ 4.7 Synopsys, Inc.

Security 2020.8.0.178244 Jun 12, 2020

Plugin homepage

The Synopsys® Code Sight™ plug-in identifies security bugs and vulnerabilities in your software while you code. It scans automatically, and highlights issues in the development environment so that you can fix them immediately.

Additionally, connecting to a Synopsys server improves scan performance and enables your entire development team to collaborate on writing better code faster.

Under Limited Availability, Code Sight now expands support for Black Duck Hub - Black Duck Quick Tour. This helps you to select OSS libraries that are non-vulnerable and ensure open source security compliance.

See what the plug-in looks like in our interactive demo: Code Sight Quick Tour.

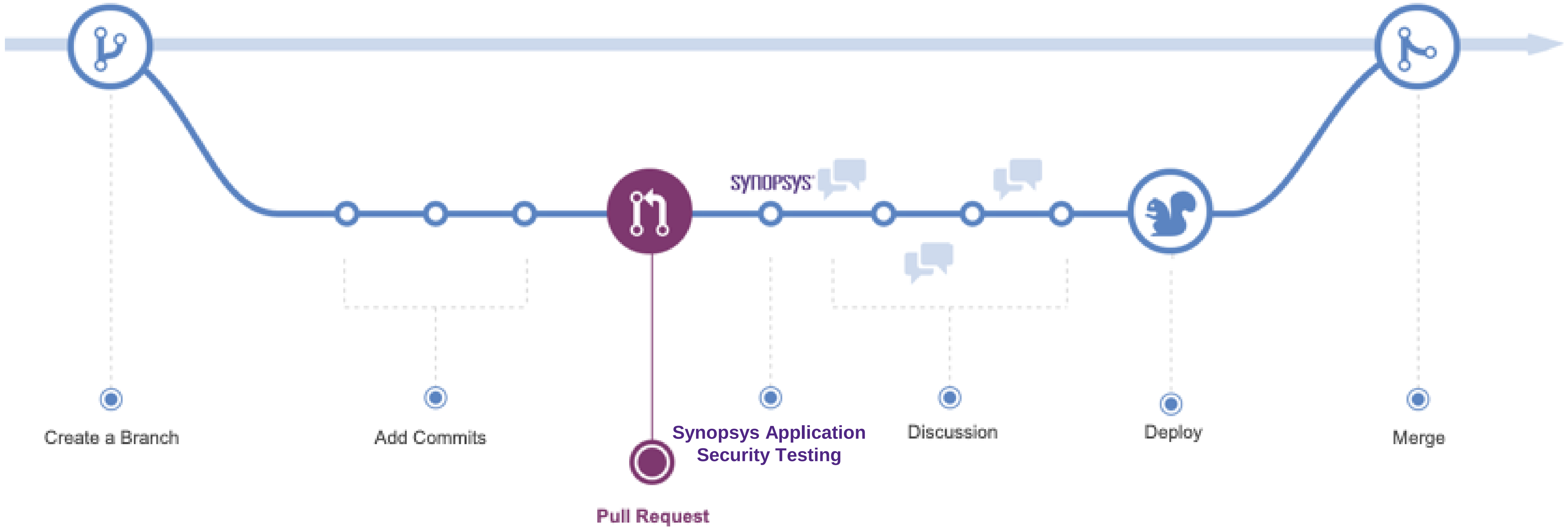
A functional Coverity license is required to run Code Sight with Coverity, and a Black Duck Hub license to use it with Black Duck Hub. Please check with your local administrator or contact software-integrity-support@synopsys.com to get access.

Build# 53
Commit info:

Feedback in a Pull Request

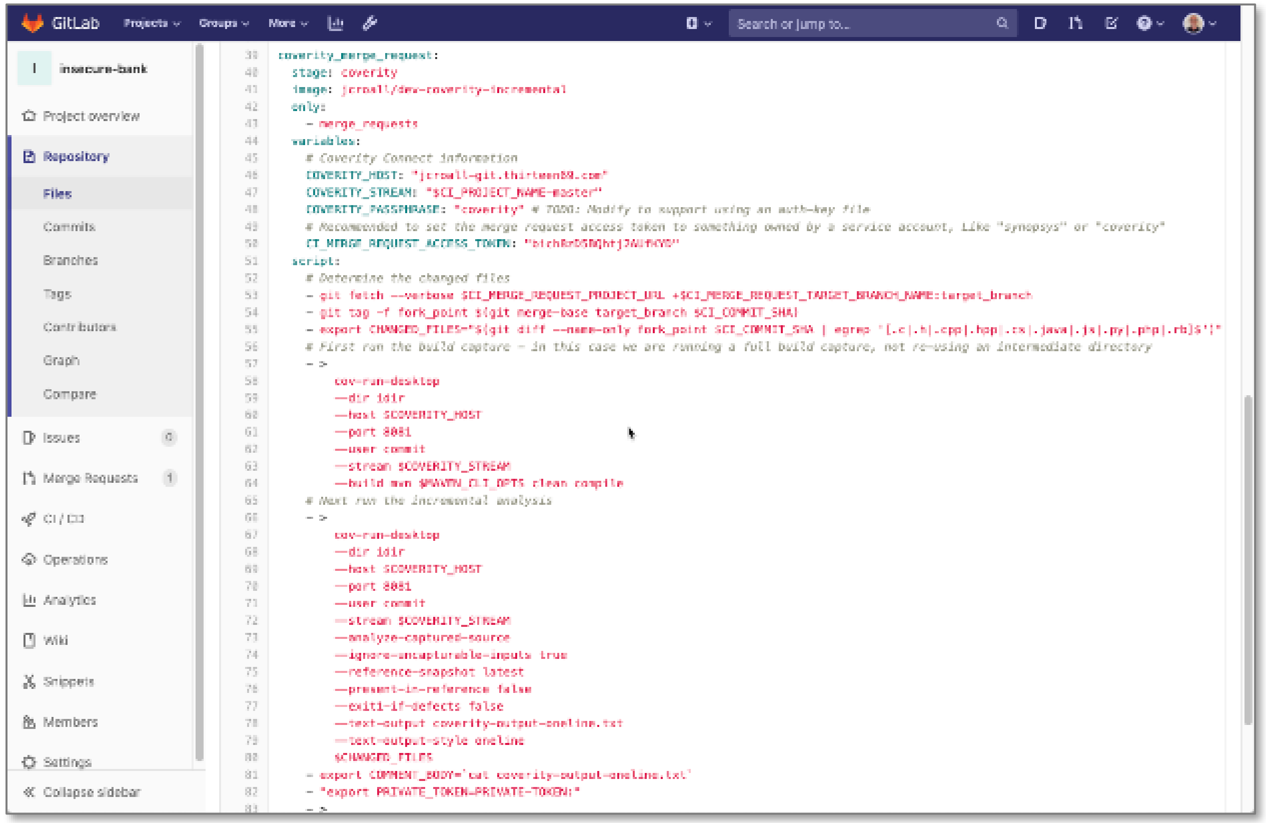
- Pull request analysis: Key factors
- Live demo

Feedback in a Pull Request

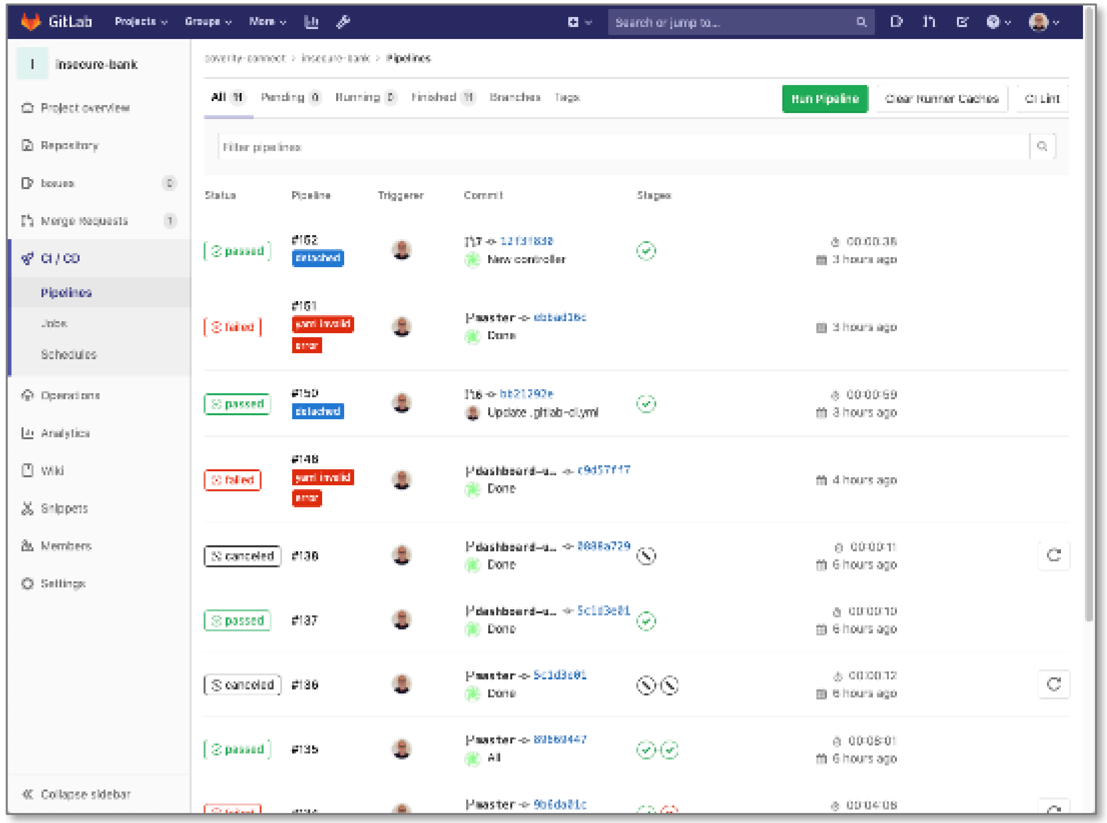


- Incremental scan
- Result summary
- Ability to create issue tickets

LIVE Demonstration: Setting Up Incremental Analysis in GitLab CI



```yaml
coverity_merge_request:
  stage: coverity
  image: jcroall/dew-coverity-incremental
  only:
    - merge_requests
  variables:
    # Coverity Connect information
    COVERITY_HOST: "jcroall-git.thirteen88.com"
    COVERITY_STREAM: "$CI_PROJECT_NAME-master"
    COVERITY_PASSPHRASE: "coverity" # TODO: Modify to support using an auth-key file
    # Recommended to set the merge request access token to something owned by a service account, Like "synopsys" or "coverity"
    CI_MERGE_REQUEST_ACCESS_TOKEN: "bichRrDSBQhfj2AUfKYD"
  script:
    # Determine the changed files
    - git fetch --verbose $CI_MERGE_REQUEST_PROJECT_URL +$CI_MERGE_REQUEST_TARGET_BRANCH_NAME:target_branch
    - git tag -f fork_point $(git merge-base target_branch $CI_COMMIT_SHA)
    - export CHANGED_FILES="$(git diff --name-only fork_point $CI_COMMIT_SHA | egrep '[.c|.h|.cpp|.hpp|.cs|.java|.js|.py|.php|.rb]$')"
    # First run the build capture - in this case we are running a full build capture, not re-using an intermediate directory
    - >
        cov-run-desktop
        --dir idir
        --host $COVERITY_HOST
        --port 8081
        --user commit
        --stream $COVERITY_STREAM
        --build mvn $MAVEN_CLI_OPTS clean compile
    # Next run the incremental analysis
    - >
        cov-run-desktop
        --dir idir
        --host $COVERITY_HOST
        --port 8081
        --user commit
        --stream $COVERITY_STREAM
        --analyze-captured-source
        --ignore-uncapturable-inputs true
        --reference-snapshot latest
        --present-in-reference false
        --exit1-if-defects false
        --text-output coverity-output-oneline.txt
        --text-output-style oneline
        $CHANGED_FILES
    - export COMMENT_BODY=`cat coverity-output-oneline.txt`
    - "export PRIVATE_TOKEN=PRIVATE-TOKEN:"
    - >
```



Status	Pipeline	Triggerer	Commit	Stages	
passed	#162 detached		!17 -o 12f31838 New controller	✓	00:00:36 3 hours ago
failed	#161 yaml invalid error		master -o ebbad16c Done		3 hours ago
passed	#150 detached		!16 -o bb21292e Update .gitlab-ci.yml	✓	00:00:59 3 hours ago
failed	#148 yaml invalid error		dashboard-u... -o c9d57f17 Done		4 hours ago
canceled	#138		dashboard-u... -o 8886a729 Done	⊘	00:00:11 6 hours ago
passed	#137		dashboard-u... -o 5c1d3e81 Done	✓	00:00:10 6 hours ago
canceled	#136		master -o 5c1d3e81 Done	⊘ ⊘	00:00:12 6 hours ago
passed	#135		master -o 83869447 All	✓ ✓	00:08:01 6 hours ago
			master -o 9b6da81c		00:04:08

More Questions?

- Visit the Synopsys Software Integrity Community page:
    - https://community.synopsys.com/s/article/Prevent-Security-Weaknesses-from-Escaping-a-GitLab-Merge-Request

Questions

Synopsys®
Silicon to Software™

Thank You