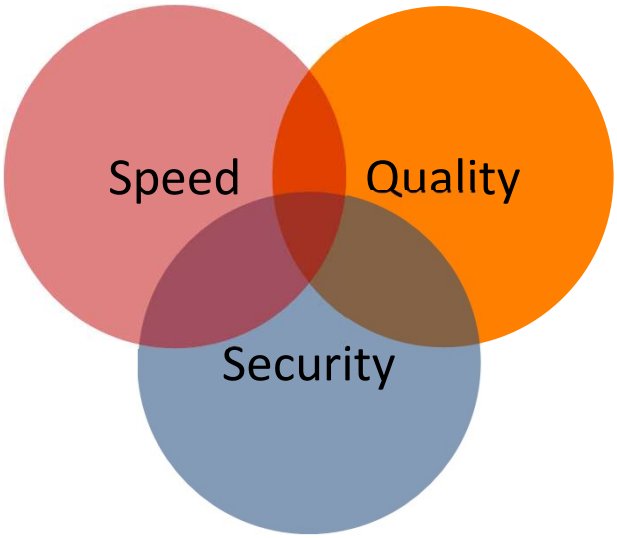
DevSecOps – Detecting 0-day and N-day vulnerabilities, everyday.

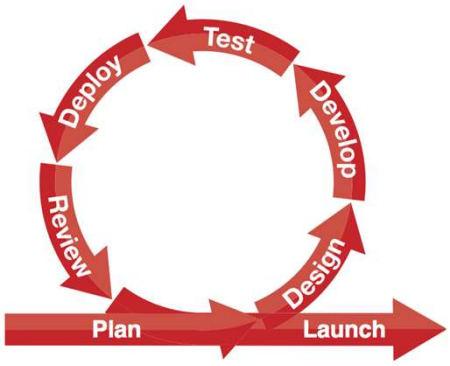
Walter Capitani | May 2021

How do you like your software?

How do you like your software?

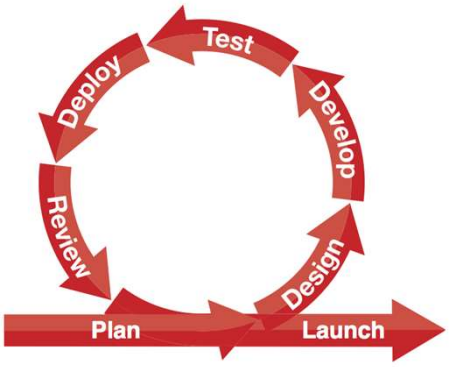
Why not all three?





The DevOps movement

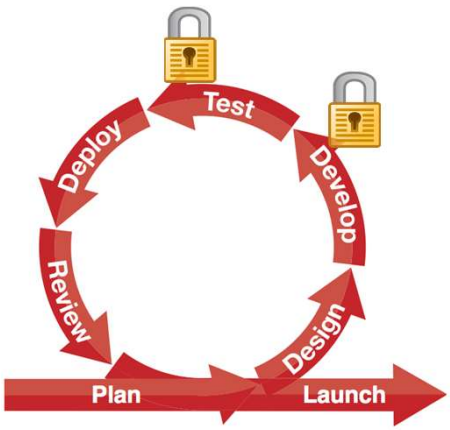
- The DevOps movement integrated testing into every code change
  - unit tests
  - functional tests
  - deployment tests



GRAMMATECH

What about security?

- Security traditionally viewed as a post-development or QA task
- The DevSecOps movement brings security testing into the development cycle
  - Find (and fix) vulnerabilities earlier
  - Reduce uncertainty
    - Does our software have any vulnerabilities?
    - How long will it take to resolve them?



GRAMMATECH

What kind of vulnerabilities are we looking for?

- 0-day
  - Can be exploited right now
  - No patch available
  - Exploits may not be widespread
- Zero Day Initiative currently tracking over 400 undisclosed vulnerabilities discovered in 2021[1]
- N-day
  - Can be exploited right now
  - Patch *is* available
  - Exploits *likely* to be widespread
- A 2020 study[2] of 60 vulnerabilities showed 30% were exploited more than 1 week after the patch was issued

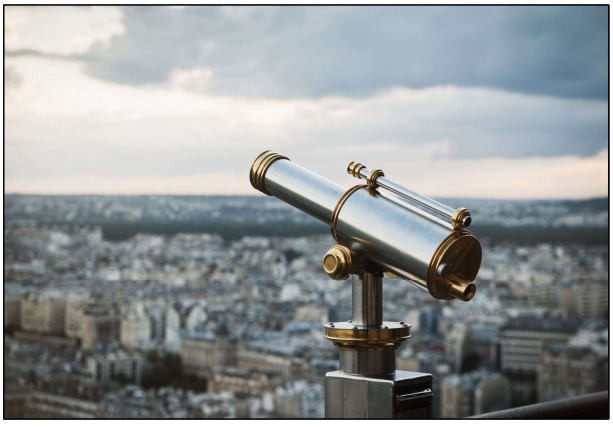
[1] https://www.zerodayinitiative.com/advisories/upcoming/

[2] https://www.fireeye.com/blog/threat-research/2020/04/time-between-disclosure-patch-release-and-vulnerability-exploitation.html

GRAMMATECH

Where can these vulnerabilities be found?

- Source code and Binaries
  - Internal
  - Open Source
  - 3rd Party



GRAMMATECH

What are the challenges?

- Source code can be expensive to fix
  - Open Source
  - 3rd Party
- No source code available
  - Binaries

Comprehensive Security Strategy

- Internal Source Code
  - Eliminate vulnerabilities
  - Your team knows the codebase
  - DevSecOps reduces the cost of remediation

- Binaries
  - Eliminate 0-Day and N-Day vulnerabilities
  - Remediation through package / binary / vendor updates

CODESonar® CODESentry™

DevSecOps for Source Code

CODESonar®

GRAMMATECH

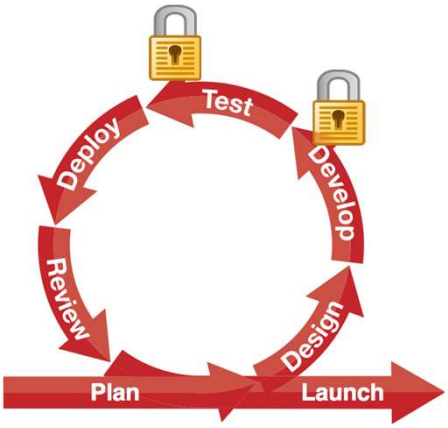
CodeSonar – Static Application Security Testing

CODESonar®

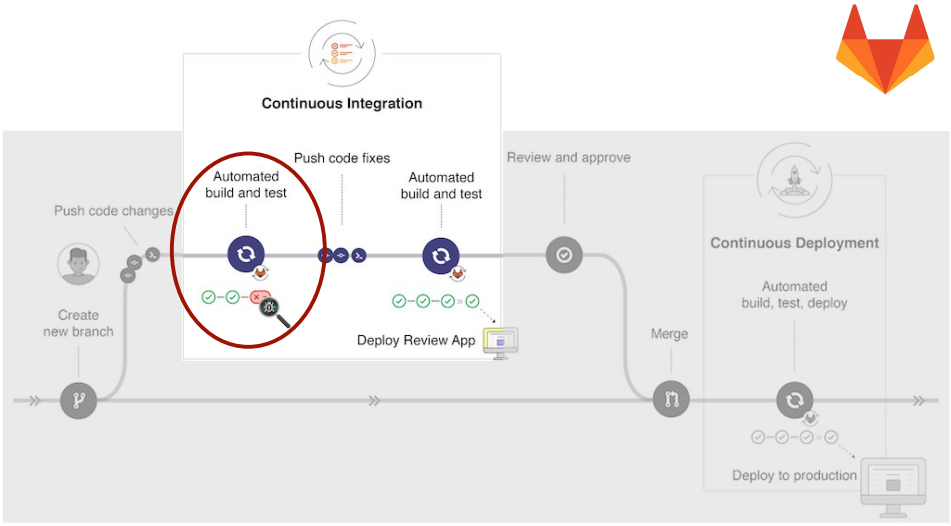
- Multi-language static analysis platform with abstract execution
  - C/C++, Java, Android, C#, Intel-32/64, ARMv7, PPC
- Warning tracking with suppression
- Developer-friendly interface
  - Clear explanations with path information
  - Whole program navigation and visualization
- Integrated with popular CI/CD tools such as GitLab and Jenkins
- Identifies Tainted Data Injection, Buffer Overflow, Hardcoded Passwords and many other security vulnerabilities

GRAMMATECH

DevSecOps for Source Code

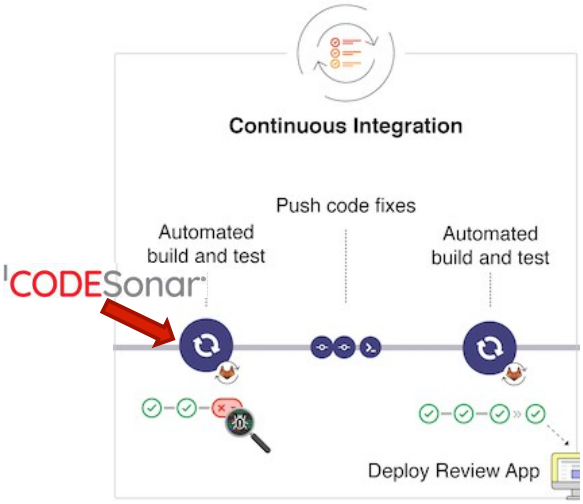


Example: GitLab Integration



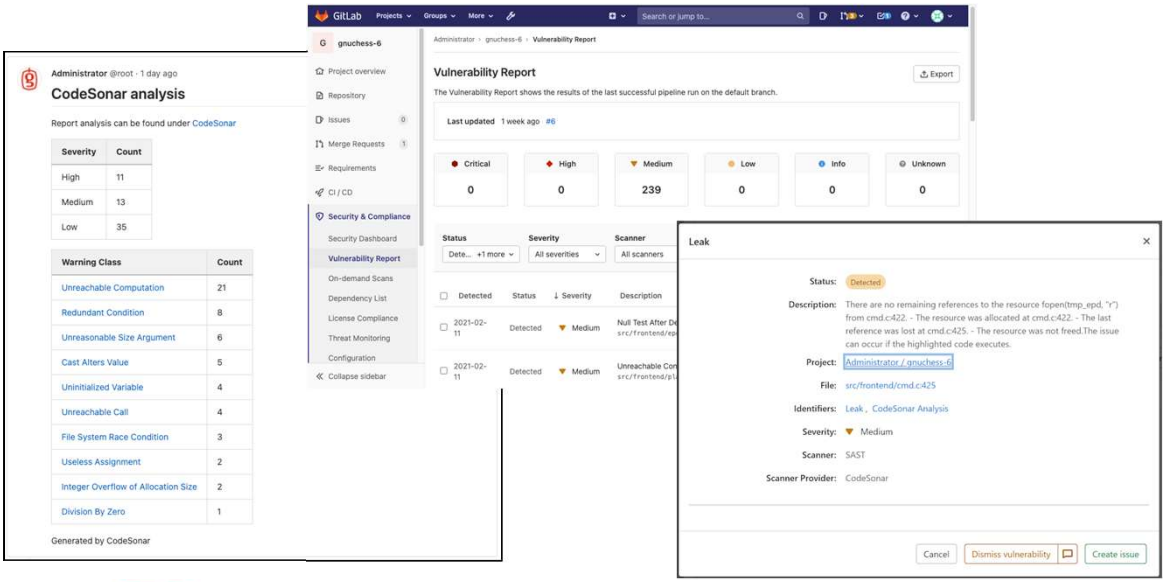
SAST in the CI/CD Pipeline

- CodeSonar is run as part of GitLab pipeline merge request
- Shift Left from QA into development
- 0-Day vulnerabilities



GRAMMATECH

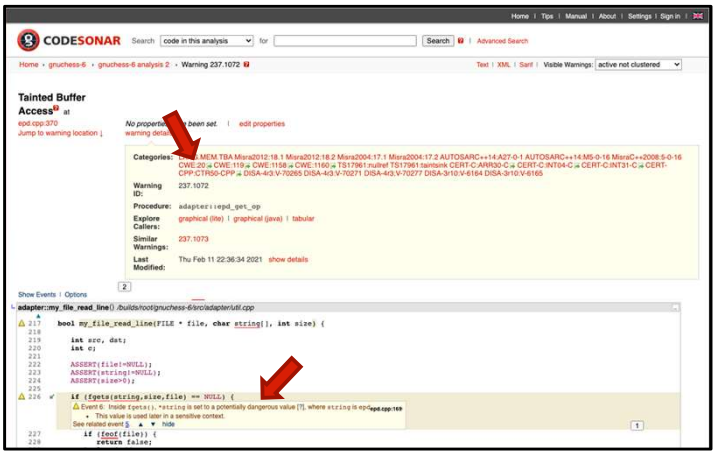
SAST in the CI/CD Pipeline



GRAMMATECH

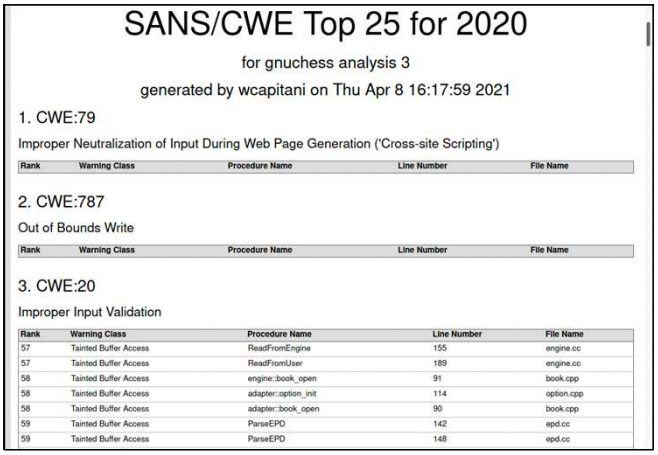
SAST in the CI/CD Pipeline

- What if we need access to the CodeSonar warning?
- Just click through to access the Hub



SAST in the CI/CD Pipeline

- Use built-in reports to track the security status of your project



SANS/CWE Top 25 for 2020

for gnuchess analysis 3

generated by wcapitani on Thu Apr 8 16:17:59 2021

1. CWE:79

Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

Rank	Warning Class	Procedure Name	Line Number	File Name

2. CWE:787

Out of Bounds Write

Rank	Warning Class	Procedure Name	Line Number	File Name

3. CWE:20

Improper Input Validation

Rank	Warning Class	Procedure Name	Line Number	File Name
57	Tainted Buffer Access	ReadFromEngine	155	engine.cc
57	Tainted Buffer Access	ReadFromUser	189	engine.cc
58	Tainted Buffer Access	engine::book_open	91	book.cpp
58	Tainted Buffer Access	adapter::option_init	114	option.cpp
58	Tainted Buffer Access	adapter::book_open	90	book.cpp
59	Tainted Buffer Access	ParseEPD	142	epd.cc
59	Tainted Buffer Access	ParseEPD	148	epd.cc

DevSecOps for Source Code

- Security of source code moved into the development cycle
- Early detection of vulnerabilities reduces:
  - the cost of remediation
  - the risk of missed release dates
- Security is not just a QA activity, but an everyday activity

DevSecOps for Binaries

CODESentry™

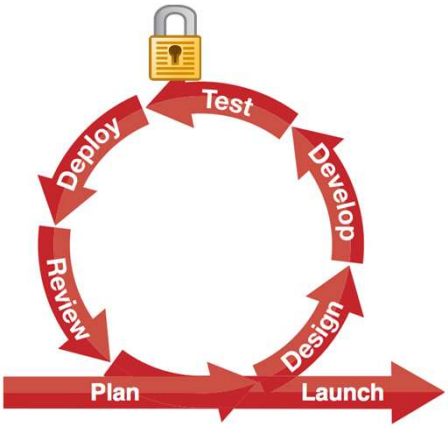
CodeSentry – Binary SCA

CODESentry™

- Binary Software Composition Analysis
  - Identifies OSS and 3rd Party components with known vulnerabilities
  - Known vulnerabilities (N-days) are a major threat
  - Continuously identifies critically exposed security flaws
  - Variable levels of vulnerability discovery
  - Creates SBOM
  - Maintains component history
- OnPrem or SaaS deployment

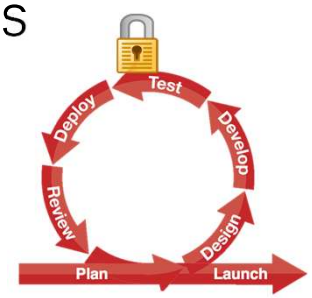
DevSecOps for Binaries

CODESentry™



DevSecOps for Binaries

- On each merge request, software package is built for testing purposes
- Submit Binary Artifacts for Analysis
- Retrieve Security Scores and KPIs
- Pass or Fail the build
- Create tasks for developers to investigate findings



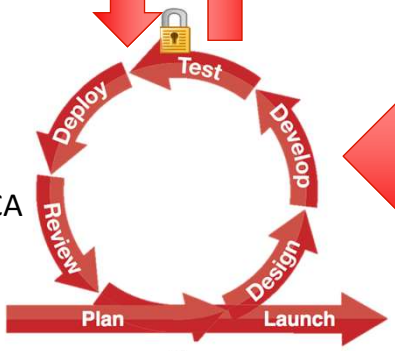
GRAMMATECH

CodeSentry API Integration

CODESentry™

Binary SCA Findings

Binary Artifacts

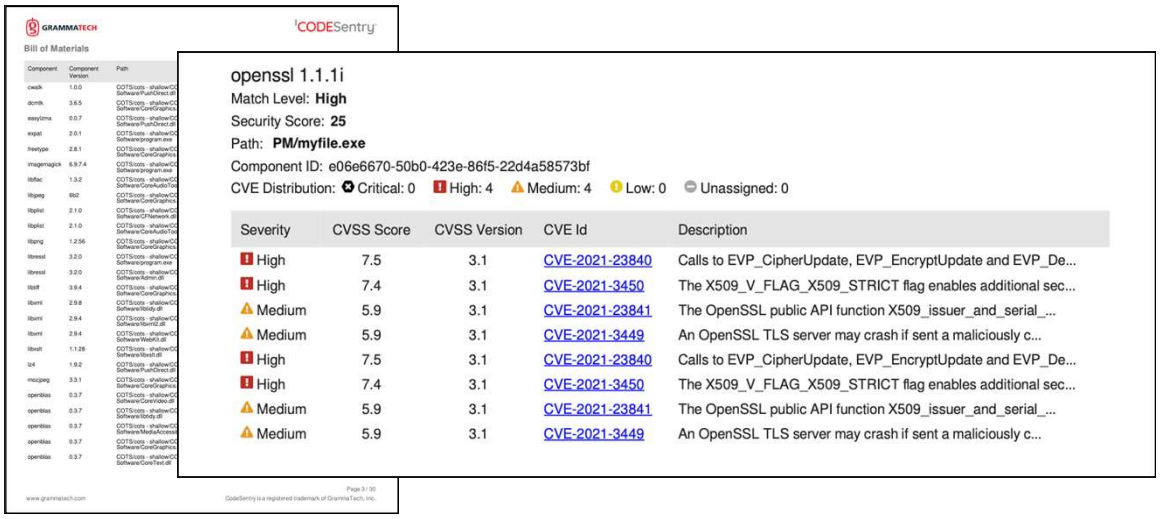


Source Code
Open Source Components
3rd Party Binaries

Review the results of Binary SCA
Pass or fail the build

GRAMMATECH

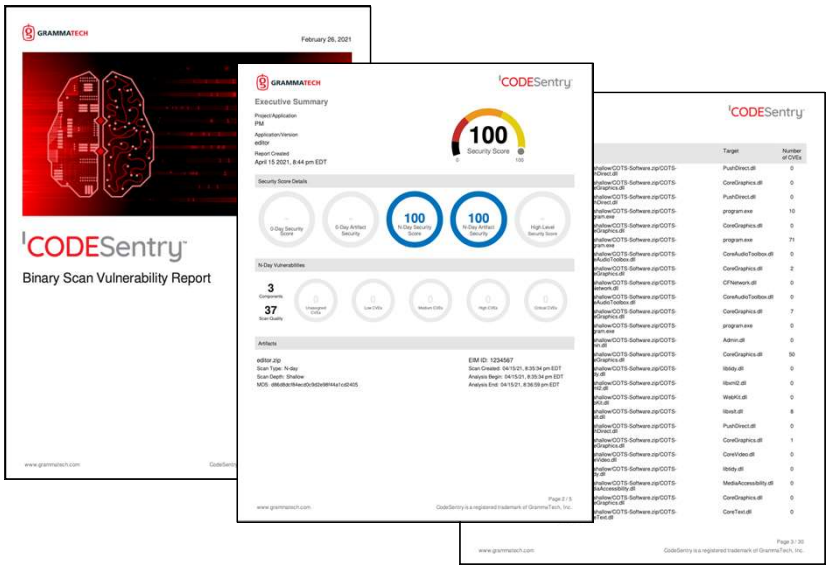
Identify vulnerable components



Information From Detected Components

- Component and version information
- Vulnerability information
  - List of CVEs and CVSS score
- List of functions included
  - Helps refine vulnerability assessment

Reports for process compliance



- Security Score
- Artifact Security
- Scan Quality
- Vulnerabilities

GRAMMATECH

DevSecOps for Binaries

- Open Source and 3rd Party binaries can contain N-Day vulnerabilities
- Binary SCA can detect vulnerable components throughput the development cycle
- Early detection of vulnerabilities:
  - Gives developers extra time to update components
  - Reduces the risk of releasing software with known exploits

GRAMMATECH

Conclusions

Conclusions

- Security must be built in to the development cycle, not just left to QA before release
- Early detection of vulnerabilities reduces:
  - the cost of remediation
  - the risk of missed release dates
- Static Code Analysis and Binary SCA can help you achieve this goal

Questions?

Walter Capitani
Director, Product Management
wcapitani@grammatech.com