How to leverage static code analysis in your CICD pipelines for continuous code quality

Dana Epp
Microsoft Regional Director
https://danaepp.com
When the code is incorrect, you can't really talk about security. When the code is faulty, it cannot be safe.

- Gene ‘Spaf’ Spafford
Quality is not an act, it is a habit.

- Aristotle
If you can’t champion code quality with your team, how can you ever champion secure code?

- Dana Epp
Passing static code analysis doesn’t prove your code is safe... but failing it pretty much signals it isn’t.

- Dana Epp
Resolving bugs early and often reduces associated costs.
WHY IS THAT?
Most studies show that inspection is cheaper than testing. [We] found that code reading detected 80% more faults per hour than testing.

- Basili and Selby 1987
## Comparing defect detection approaches

<table>
<thead>
<tr>
<th>Removal Step</th>
<th>Lowest Rate</th>
<th>Modal Rate</th>
<th>Highest Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Informal design reviews</td>
<td>25%</td>
<td>35%</td>
<td>40%</td>
</tr>
<tr>
<td><strong>Formal design inspections</strong></td>
<td>45%</td>
<td>55%</td>
<td>65%</td>
</tr>
<tr>
<td>Informal code reviews</td>
<td>20%</td>
<td>25%</td>
<td>35%</td>
</tr>
<tr>
<td><strong>Formal code inspections</strong></td>
<td>45%</td>
<td>60%</td>
<td>70%</td>
</tr>
<tr>
<td>Modeling or prototyping</td>
<td>35%</td>
<td>65%</td>
<td>80%</td>
</tr>
<tr>
<td>Personal desk-checking of code</td>
<td>20%</td>
<td>40%</td>
<td>60%</td>
</tr>
<tr>
<td>Unit test</td>
<td>15%</td>
<td>30%</td>
<td>50%</td>
</tr>
<tr>
<td>New function (component) test</td>
<td>20%</td>
<td>30%</td>
<td>35%</td>
</tr>
<tr>
<td>Integration test</td>
<td>25%</td>
<td>35%</td>
<td>40%</td>
</tr>
<tr>
<td>Regression test</td>
<td>15%</td>
<td>25%</td>
<td>30%</td>
</tr>
<tr>
<td>System test</td>
<td>25%</td>
<td>40%</td>
<td>55%</td>
</tr>
<tr>
<td>Low-volume beta test (&lt;10 sites)</td>
<td>25%</td>
<td>35%</td>
<td>40%</td>
</tr>
<tr>
<td>High-volume beta test (&gt;1,000 sites)</td>
<td>60%</td>
<td>75%</td>
<td>85%</td>
</tr>
</tbody>
</table>

Source: Adapted from *Programming Productivity* (Jones 1986a), “Software Defect-Removal Efficiency” (Jones 1996), and “What We Have Learned About Fighting Defects” (Shull et al. 2002).
What can static code analysis do for me??
Know the quality of your code at all times

```java
if (Provider.class == roleTypeClass) {
    Type providedType = ReflectionUtils.getTypeClass(providedType);
    Class providedClass = ReflectionUtils.getTypeClass(providedType);
}
```

A "NullPointerException" could be thrown; "providedClass" is nullable here.

<table>
<thead>
<tr>
<th>Reliability</th>
<th>Bugs</th>
<th>Security</th>
<th>Maintainability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>Security Vulnerabilities</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Security Hotspots</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Technical Debt</td>
<td>6 days</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Code Smells</td>
<td>319</td>
</tr>
</tbody>
</table>

New code

- Reliability: 1 (B)
- Security: 0 (A)
- Maintainability:
  - Technical Debt: 0 (A)
  - Code Smells: 0 (A)
Detect bugs

A "NullPointerException" could be thrown; "getFilter()" can return null.

1. Implies 'filter' can be null.
2. 'Exception' is caught.
3. 'getFilter()' can return null.
4. Result of 'getFilter()' is dereferenced.

Insert a <!DOCTYPE> declaration to before this <html> tag.

Add a 'favicon' declaration in this 'header' tag.
Detect ‘code smells’

```csharp
namespace AdWorks.MVC.Controllers
{
    public class HomeController : Controller
    {
        public IActionResult Index()
        {
            dynamic obj = "hello";

            // Remove this useless assignment to local variable 'obj'.
            obj = new { name = "fred" };
            obj = 10;
        }
    }
}
```
Explore more execution paths
Discover cognitive complexity issues

```csharp
int offset = 0;
while (true)
{
    int count = offset;
    byte b;
    while (count < MaxCharBytesSize && (b = _reader.ReadByte()) > 0)
    {
        _byteBuffer[count++] = b;
    }
    int byteCount = count - offset;
    totalBytesRead += byteCount;

    if (count < MaxCharBytesSize && builder == null)
    {
        // pref optimization to avoid reading into a string builder
    }
}

return new string(_charBuffer, 0, length);

else
{
    // calculate the index of the end of the last full character in the buffer
    int lastFullCharStop = GetLastFullCharStop(count - 1);

    int charCount = Encoding.UTF8.GetChars(_byteBuffer, 0, lastFullCharStop + 1, _charBuffer, 0);

    if (builder == null)
    {
        builder = new StringBuilder(MaxCharBytesSize * 2);
    }
    builder.Append(_charBuffer, 0, charCount);

    if (lastFullCharStop + byteCount - 1).
```
Find security vulnerabilities

// dumpObj:
dumpObj: function( spec ) {
    var val = "<undefined>";
    try {
        val = eval( "this."+spec ).toString();
    }
    catch( exception ) {
    }
    this.dump( spec + '=' + val + '\n' );
},

Review the arguments of this "eval" call to make sure they are validated.

Vulnerability Critical Open Not assigned 30min effort Comment
cwe, owasp-a3
Review security ‘hotspots’

```java
/**
 * Sets the maximum age of the cookie in seconds.
 */

public CookieBuilder setExpiry(int expiry) {
    this.expiry = expiry;
    return this;
}

public Cookie build() {
    Cookie cookie = new Cookie(requireNonNull(name), value);
    cookie.setPath(getContextPath(request));
    cookie.setSecure(isHttps(request));
    cookie.setHttpOnly(httpOnly);
    cookie.setMaxAge(expiry);
    return cookie;
}
```
Enforce security best practices

```java
public static KeyPair generateRsaOrDsa(boolean rsa) throws Exception {
    if (rsa) {
        KeyPairGenerator keyPairGen =
            KeyPairGenerator.getInstance("RSA");
        keyPairGen.initialize(1024);
        RSAKeyGenParameterSpec keySpec = new RSAKeyGenParameterSpec(1024,
            RSAKeyGenParameterSpec.F0);
        keyPairGen.initialize(keySpec);
        KeyPair rsaKeyPair = keyPairGen.generateKeyPair();
        return rsaKeyPair;
    } else {
```
Untrusted input analysis (taint analysis)

```csharp
public string GetThing(string id)
{
    var connection = new SqlConnection();
    try
    {
        connection.ConnectionString = "db info";
        connection.Open();

        var selectSql = string.Format("select from MyStuff where id={0}" , id);
        var selectCommand = new SqlCommand(selectSql, connection);

        var dataReader = selectCommand.ExecuteReader();
        return dataReader.GetString(0);
    }
    catch (Exception ex)
    {
    }
    finally
    {
    }
```
**OWASP / SANS security reports**

- **Requires SonarQube Enterprise**

<table>
<thead>
<tr>
<th>Categories</th>
<th>Security Vulnerabilities</th>
<th>Security Hotspots</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 - Injection</td>
<td>2</td>
<td>302</td>
</tr>
<tr>
<td>A2 - Broken Authentication</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A3 - Sensitive Data Exposure</td>
<td>1</td>
<td>133</td>
</tr>
<tr>
<td>A4 - XML External Entities (XXE)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A5 - Broken Access Control</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A6 - Security Misconfiguration</td>
<td>3</td>
<td>31</td>
</tr>
<tr>
<td>A7 - Cross-Site Scripting (XSS)</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>A8 - Insecure Deserialization</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>A9 - Using Components with Known Vulnerabilities</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A10 - Insufficient Logging &amp; Monitoring</td>
<td>0</td>
<td>17</td>
</tr>
</tbody>
</table>
OWASP / SANS security reports

- **Security Category**
  - **SonarSource**
    - Others: 2.4k
    - Insecure Configuration: 20
    - Weak Cryptography: 1
  - **OWASP Top 10**
    - A3 - Sensitive Data Exposure: 1
    - A6 - Security Misconfiguration: 1
  - **SANS Top 25**
    - Porous Defenses: 1

- **CWE**
  - No CWE associated: 2.2k
  - CWE-546 - Suspicious Comment: 81
  - CWE-563 - Assignment to Variable: 60
  - CWE-397 - Declaration of Throws: 26
  - CWE-489 - Leftover Debug Code: 20
  - CWE-570 - Expression is Always False: 20
  - CWE-571 - Expression is Always True: 20
  - CWE-476 - NULL Pointer Dereference: 19
  - CWE-493 - Critical Public Variable: 19
  - CWE-327 - Use of a Broken or Risky Algorithm: 1
  - CWE-328 - Reversible One-Way Hash: 1
  - CWE-759 - Use of a One-Way Hash: 1
  - CWE-760 - Use of a One-Way Hash: 1
  - CWE-916 - Use of Password Hash: 1

- Available in SonarCloud
Our DevOps toolchain

- Azure Boards
- Azure Repos
- Azure Pipelines
- Azure Artifacts

- sonarlint
- sonarcloud
Our Stack

• Typescript code targeting NodeJS deployed to Web App for Containers
• C++ code targeting Linux shell deployed to Azure Container Instances
• C# code targeting .NET Core 3.1 deployed to Azure Container Instances
• C# code targeting .NET Core 2.1 deployed to Azure Functions
• Typescript code targeting Angular 8 deployed to Azure CDN / Frontdoor
Languages SonarQube supports
We start with SonarLint – Democratize quality
We enforce peer code review before merge

- Require a minimum number of reviewers
  - Minimum number of reviewers: 2
  - Requestors can approve their own changes
  - Allow completion even if some reviewers vote to wait or reject
  - Reset code reviewer votes when there are new changes

- Check for linked work items
  - Policy requirement
    - Required: Block pull requests from being completed unless they have at least one linked work item.
    - Optional: Warn if there are no linked work items, but allow pull requests to be completed.

- Check for comment resolution
  - Check to see that all comments have been resolved on pull requests.
Merge triggers build pipeline

- Inject static code analysis agent into build environment, configured to your project in SonarCloud
- Execute static code analysis
- Report results to SonarCloud
Build success triggers release pipeline

Enable Deployment Gates

Quality Gate enforcement

Pre-deployment conditions
GraphCollector to QA

Triggers
Define the trigger that will start deployment to this stage

Pre-deployment approvals
Select the users who can approve or reject deployments to this stage

Gates
Define gates to evaluate before the deployment. Learn more

The delay before evaluation

Deployment gates

Check SonarCloud Quality Gate status
SonarCloud Quality Gate status check (Preview)

Task version 1.* (preview)

Display name
Check SonarCloud Quality Gate status

Output Variables
More information / links

Tools

- Azure DevOps: https://dev.azure.com
- SonarLint: https://www.sonarlint.org/
- SonarQube: https://www.sonarqube.org/
- SonarCloud: https://www.sonarcloud.io

Follow

- Dana Epp: https://danaepp.com
- AuditWolf: https://www.auditwolf.com
Questions??

Dana Epp
Microsoft Regional Director
https://danaepp.com