# OWASP Dependency Check

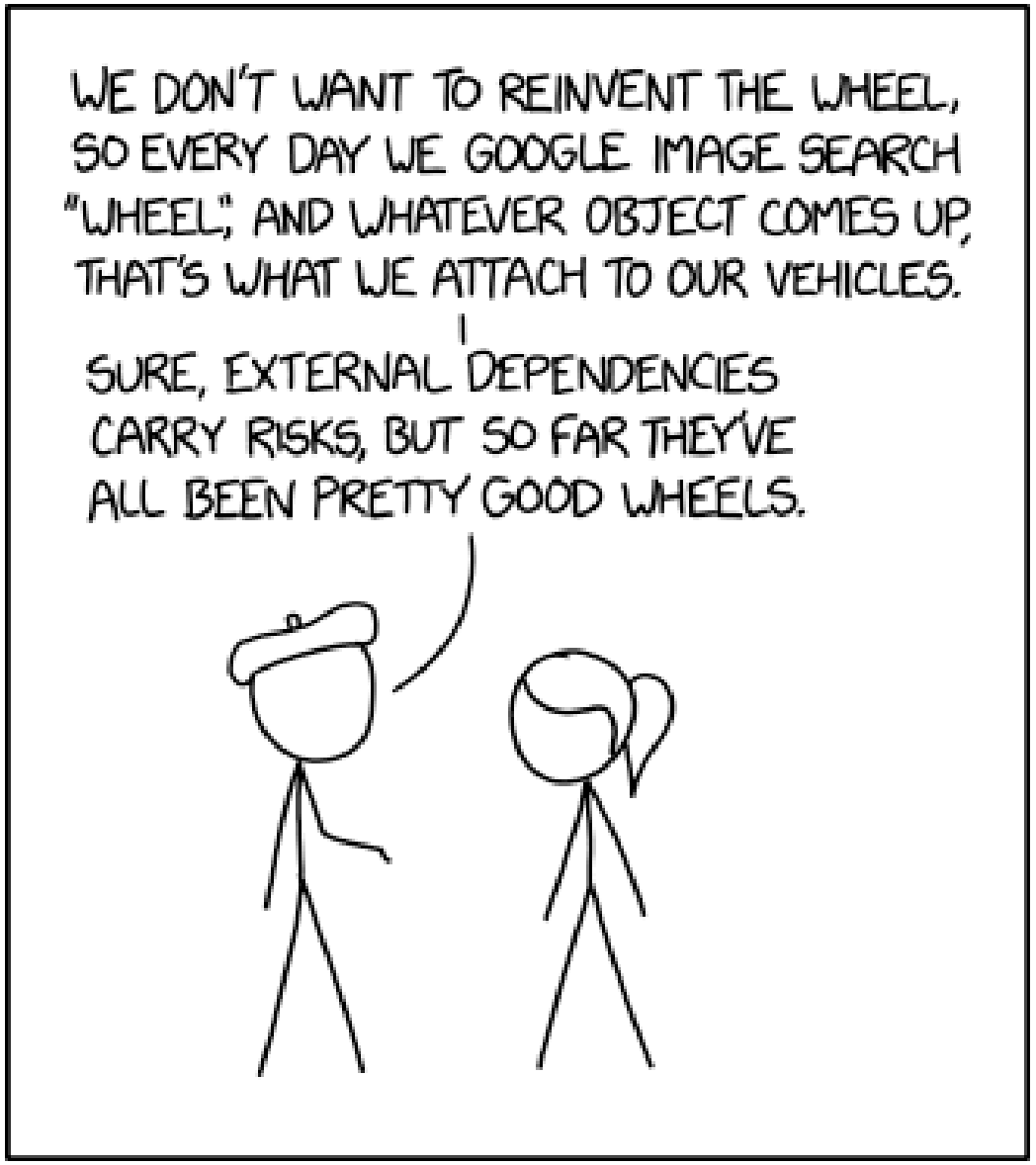
## A Primer

# The Problem

## In A Nutshell

1. Modern applications depend on open source,
2. they contain many 3rd party components **and their vulnerabilities**.

*"Reinvent The Wheel"* by xkcd: https://xkcd.com/2140/



# The Problem

The problem has been recognized by the OWASP Top 10 Web Application Security Risks.

## OWASP Top 10

> *"**#9 Using Components with Known Vulnerabilities.** Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts."* [owasp.org]

For more Information see: Using Components with Known Vulnerabilities

# OWASP Dependency Check

Tool for checking project dependencies for known vulnerabilities.

- Developed by OWASP / Jeremy Long
  - Project site on owasp.org
  - Online documentation on github.io

# Azure Pipeline Integration

## Hosted Agents

```yaml
- task: dependency-check-build-task@5
  displayName: 'Dependency Check: Run'
  inputs:
    projectName: MyProject        # name of the project
    scanPath: path/to/scanPath    # path of artifacts to scan
    failOnCVSS: 0                 # threshold when to fail build
    format: 'HTML'                # output format
    enableExperimental: false     # use experimental analyzers
    enableRetired: false          # use retired analyzers
    enableVerbose: false          # run in verbose mode
```

# Azure Pipeline Integration

## On-Premise Agents

Dependency-Check needs JRE/JDK to run

```yaml
- task: JavaToolInstaller@0
  displayName: 'Dependency Check: Install OpenJDK'
  inputs:
    versionSpec: "13"
    jdkArchitectureOption: x64
    jdkSourceOption: LocalDirectory
    jdkFile: "path/to/openjdk-13.0.2_windows-x64_bin.zip"
    jdkDestinationDirectory: "DependencyCheck/Binaries/Externals"
    cleanDestinationDirectory: true

- task: dependency-check-build-task@5
  ...
```

# Azure Pipeline Integration

## On-Premise Agents

.NET Analyzers require .NET Core

```yaml
- task: UseDotNet@2
  displayName: 'Dependency Check: Install .NET Core sdk'
  inputs:
    packageType: sdk
    version: 2.x
    installationPath: $(Agent.ToolsDirectory)/dotnet

- task: JavaToolInstaller@0
  ...

- task: dependency-check-build-task@5
  ...
```

# Common Vulnerabilities and Exporsers (CVE)

# Common Weakness Enumeration (CWE)

# Common Vulnerability Scoring System (CVSS)

# Common Platform Enumeration (CPE)

# How It Works

# Reading Reports

# False Positives

```yaml
- task: dependency-check-build-task@5
  displayName: 'Dependency Check: Run'
  inputs:
    ...
    suppressionPath: 'path/to/DependencyCheck/Supressions.xml' # add a supression file
    ...
```

# False Negatives

```yaml
- task: dependency-check-build-task@5
  displayName: 'Dependency Check: Run'
  inputs:
    ...
    additionalArguments: '--hints "$(Build.SourcesDirectory)/path/to/DependencyCheck/Hints.xml"' # add a hints file
    ...
```

# Thanks!