CONTINUOUS SECURITY: INTEGRATE SECURITY INTO YOUR DEVOPS PIPELINES

BILL DINGER
Computer Guy| @adazlian
wdinger@gmail.com

SECURITY LAPSES RUIN CUSTOMERS TRUST IN YOU

I am rugged and, more importantly, my code is rugged.
I recognize that software has become a foundation of our modern world.
I recognize the awesome responsibility that comes with this foundational role.
I recognize that my code will be used in ways I cannot anticipate, in ways it was not designed, and for longer than it was ever intended.
I recognize that my code will be attacked by talented and persistent adversaries who threaten our physical, economic, and national security.
I recognize these things - and I choose to be rugged.
I am rugged because I refuse to be a source of vulnerability or weakness.
I am rugged because I assure my code will support its mission.
I am rugged because my code can face these challenges and persist in spite of them.
I am rugged, not because it is easy, but because it is necessary and I am up for the challenge.

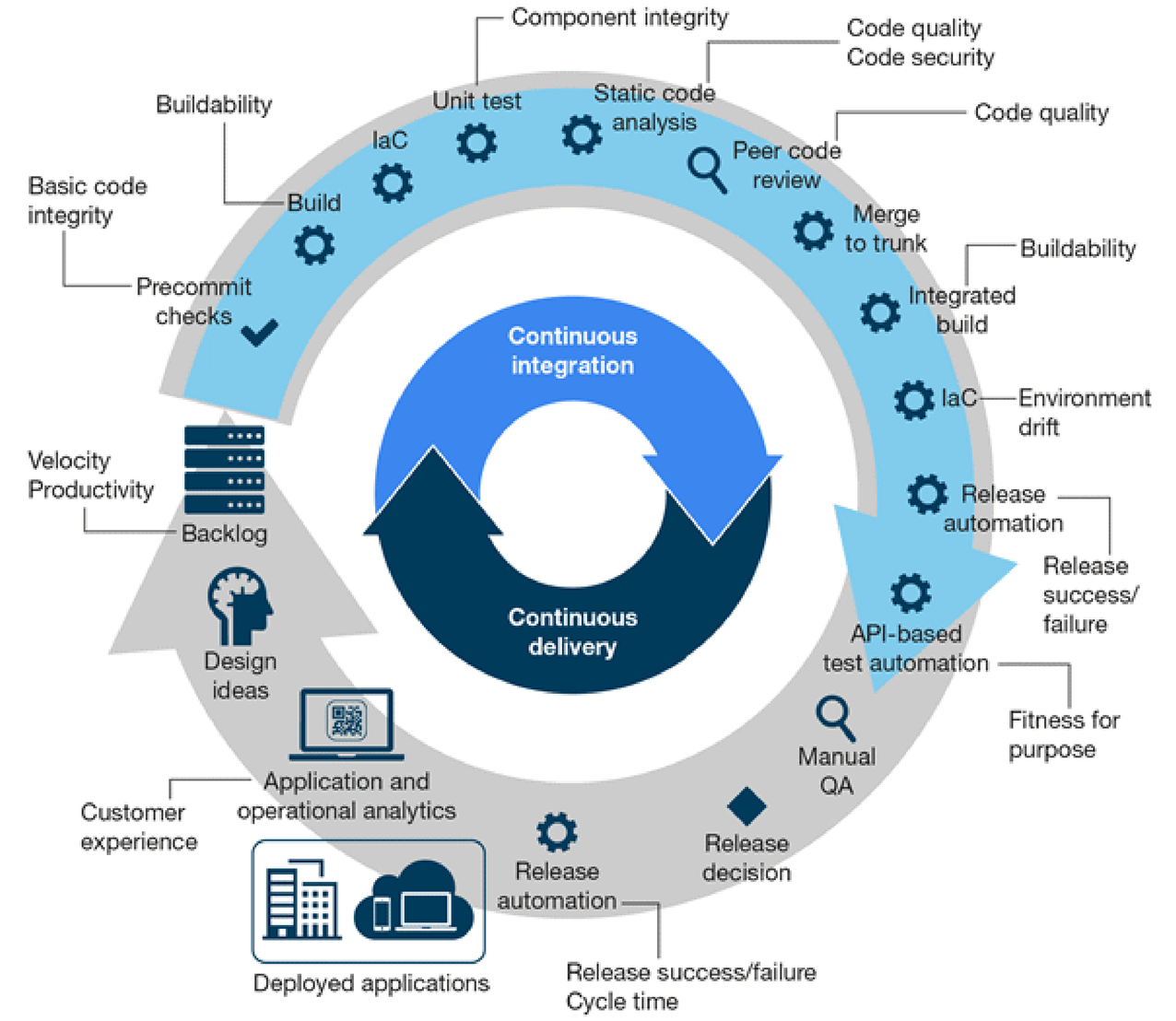
SHIFT LEFT: TEST EARLY & TEST OFTEN

“Another finding in our research is that teams that build security into their work also do better at continuous delivery […] When the tools provided actually make life easier for the engineers who use them, they will adopt them of their own free will.”

“We found that high performers were spending 50% less time remediating security issues than low performers. In other words, by building security into their daily work, as opposed to retrofitting security concerns at the end, they spent significantly less time addressing security issues.”

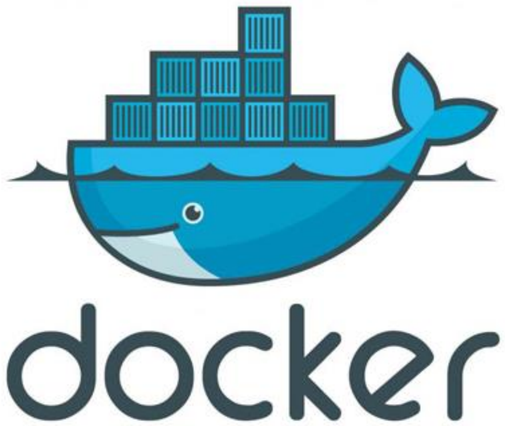
Accelerate – Nicole Forsgren, Jez Humble & Gene Kim

BRIDGING THE GAP



https://www.forrester.com/report/Evolve+Or+Retire+Administrators+Are+Now+Developers

COUPLE ASSUMPTIONS…



WHAT IS OWASP?

- **Open Web Application Security Project** – a 501(c)(3) focused on developing and improving the security of software
- Formed in 2001, its core purpose is to “Be the thriving global community that drives visibility and evolution in the safety and security of the world’s software"
- Provides numerous resources besides the OWASP Top 10, including personal favorites such as the Cheat Sheet Series, SAMM Project and Zap Attack Proxy Project

OUR TOOLS

Code:	
	Dependency Check
	Zed Attack proxy (ZAP)
	SSL Advanced Forensic Tool (O-SAFT)
	Sonarqube Rules
Documentation:	Application Security Verification Standard
	Cheat Sheet Series

OWASP DEPENDENCY CHECK

OWASP DEPENDENCY CHECK

Benefits

Protects against the OWASP #9 – using components with known security vulnerabilities automatically as part of your build.

How it works

Scans dependencies and compares them against the NIST National Vulnerability Database (NVD) for any posted CVEs. Supports suppression lists, report generation, failure depending on severity, etc.

Implementations

- Java, JavaScript, & .NET are officially supported.
- Python, Ruby, PHP, Go, Swift, C/C++ experimental support.
- Command Line/Jenkins/Gradle/maven/Ant plugins and tooling all exist in various forms of maturity.

DEPENDENCY CHECK IN ACTION

Java (Maven)

```
mvn dependency-check:check
```

```xml
<plugin>
  <groupId>org.owasp</groupId>
  <artifactId>dependency-check-maven</artifactId>
  <version>6.3.1</version>
  <configuration>
    <cveValidForHours>6</cveValidForHours>
    <failBuildOnCVSS>4</failBuildOnCVSS>
    <format>XML</format>
  </configuration>
  <executions>
    <execution>
      <goals>
        <goal>check</goal>
      </goals>
      <phase>validate</phase>
    </execution>
  </executions>
</plugin>
```

```
[INFO] Central analyzer disabled
[INFO] Checking for updates
[INFO] Skipping NVD check since last check was within 24 hours.
[INFO] Skipping RetireJS update since last update was within 24 hours.
[INFO] Check for updates complete (24 ms)
[INFO] Analysis Started
[INFO] Finished Archive Analyzer (1 seconds)
[INFO] Finished File Name Analyzer (0 seconds)
[INFO] Finished Jar Analyzer (0 seconds)
[INFO] Finished Dependency Merging Analyzer (0 seconds)
[INFO] Finished Version Filter Analyzer (0 seconds)
[INFO] Finished Hint Analyzer (0 seconds)
[INFO] Created CPE Index (1 seconds)
[INFO] Skipping CPE Analysis for npm
[INFO] Finished CPE Analyzer (1 seconds)
[INFO] Finished False Positive Analyzer (0 seconds)
[INFO] Finished NVD CVE Analyzer (0 seconds)
[INFO] Finished Vulnerability Suppression Analyzer (0 seconds)
[INFO] Finished Dependency Bundling Analyzer (0 seconds)
[INFO] Analysis Complete (3 seconds)
[INFO] ------------------------------------------------------------------------
```

DEPENDENCY CHECK IN ACTION

Command Line

```
D:\_Code\BikeShop\dependency-check\bin>dependency-check -s ..\**\*.dll --project BikeShop
```

MSBuild Task

```
<Target Name="OWASPDepdendencyCheck" AfterTargets="Build">

<!-- Executes the OWASP Command Line Checker
    For additional information and command line options see
    https://owasp.org/www-project-dependency-check/ and for CLI
    options see https://jeremylong.github.io/DependencyCheck/dependency-check-cli/
    -->
    <!-- This is the *nix version of the command line checker -->
    <Exec Condition="'$(OS)' == 'Unix'" Command="cd ../../../ %3B cli/bin/depender
    <!-- This is the windows version of the command line checker -->
    <Exec Condition="'$(OS)' == 'Windows_NT'" Command="cd ..\..\..\ %3B cli/bin/de

</Target>
```

DEPENDENCY CHECK IN ACTION

PHP (experimental – Composer Only)

```
aawjd@neliml49581 php % ../cli/bin/dependency-check.sh  --project owaspdemo  --o
ut . --scan . --enableExperimental --data owaspdata --cveValidForHours 24
```

DEPENDENCY CHECK IN ACTION

Jenkins Integration

0 vulnerabilities.

- No warnings since build 619.
- New zero warnings highscore: no warnings for 4 days!

DependencyCheck Result

Warnings Trend

All Warnings	New Warnings	Fixed Warnings
153	138	0

Summary

Total	High Priority	Normal Priority	Low Priority
153	24	111	18

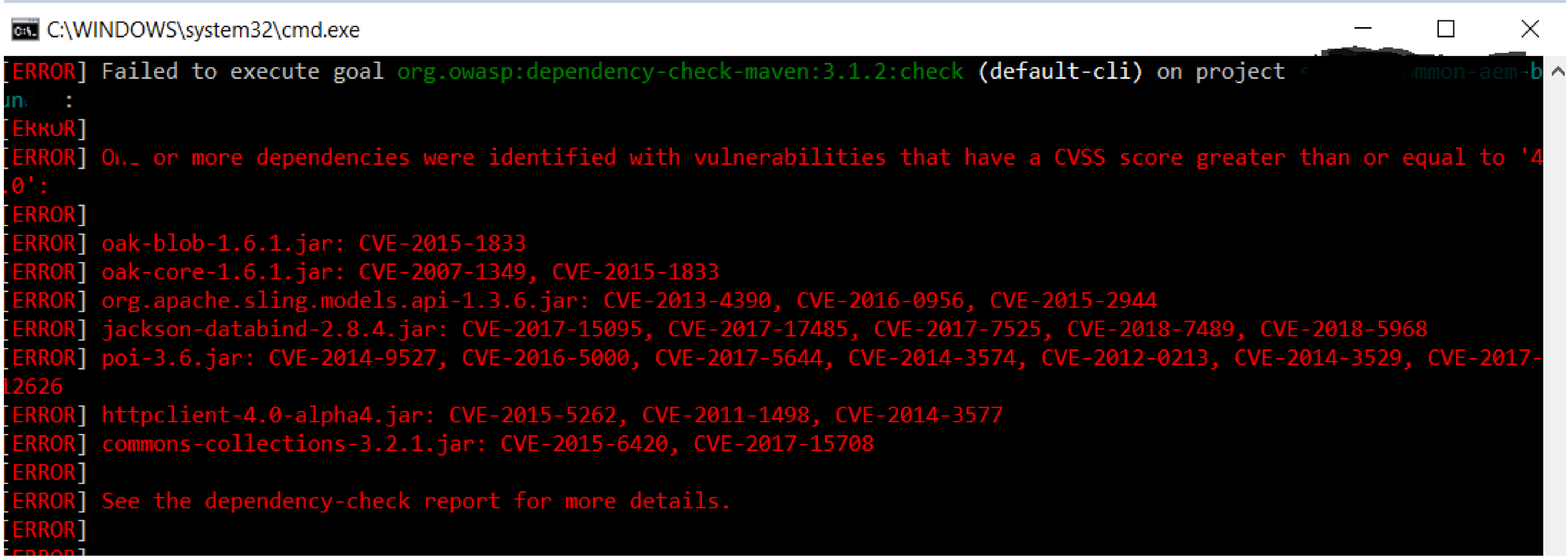
Details

Files | **Categories** | Types | Warnings | Details | New | High | Normal | Low

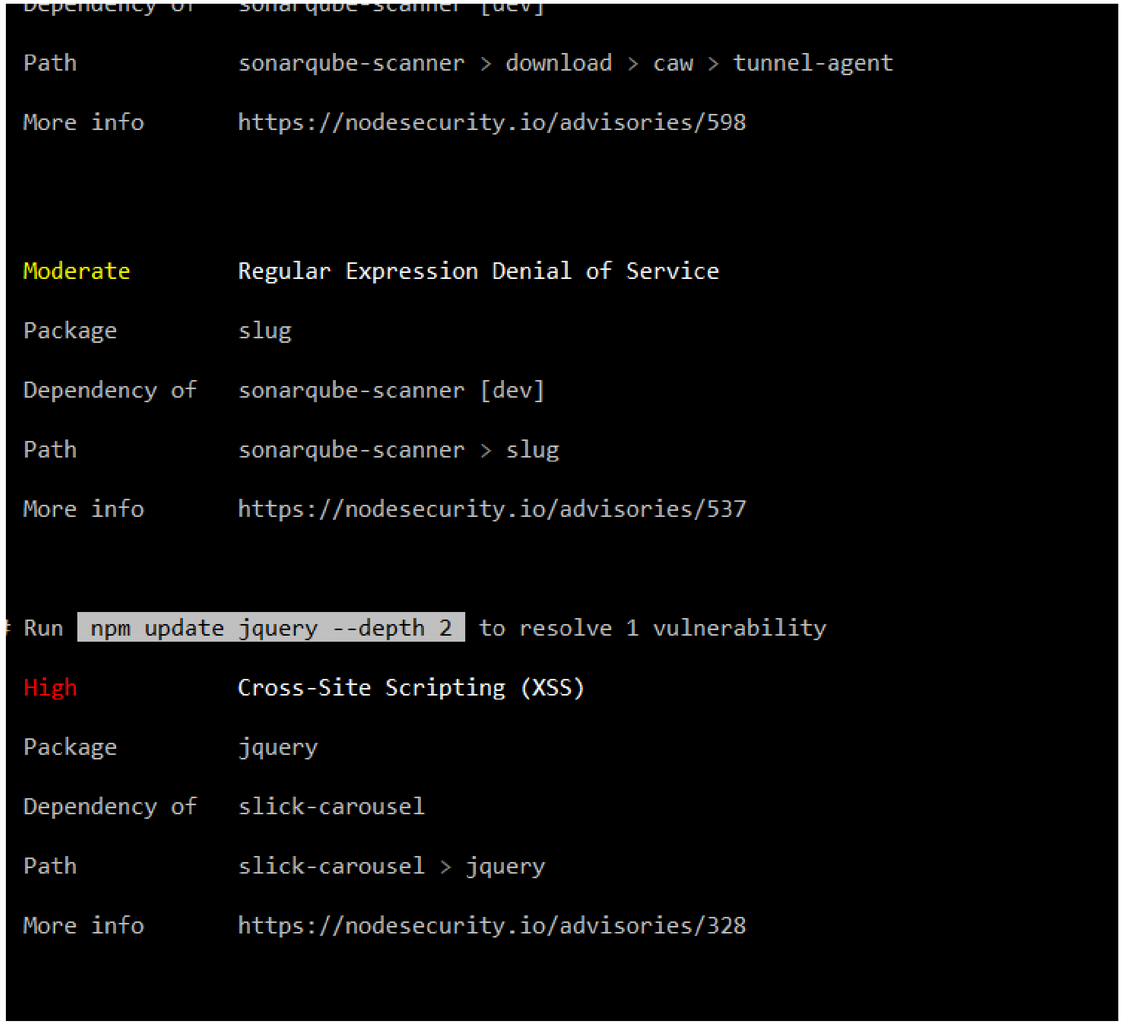
Category	Total	Distribution
CWE-119 Improper Restriction of Operations within the Bounds of a Memory Buffer	5	
CWE-134 Uncontrolled Format String	1	
CWE-189 Numeric Errors	2	
CWE-20 Improper Input Validation	7	
CWE-200 Information Exposure	5	
CWE-22 Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	4	
CWE-264 Permissions, Privileges, and Access Controls	4	
CWE-287 Improper Authentication	2	
CWE-310 Cryptographic Issues	2	
CWE-399 Resource Management Errors	7	
CWE-59 Improper Link Resolution Before File Access ('Link Following')	4	
CWE-79 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	14	
CWE-89 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	2	
CWE-94 Improper Control of Generation of Code ('Code Injection')	10	
Total	**153**	

```
configure { project ->
  project / 'builders' / 'org.jenkinsci.plugins.DependencyCheck.DependencyCheckBuilder'(plugin: 'dependency-check-jenkins-plugin@3.1.2.1') {
    skipOnScmChange false
    skipOnUpstreamChange false
    scanpath ''
    outdir ''
    datadir ''
    suppressionFile ''
    hintsFile ''
    zipExtensions ''
    isAutoupdateDisabled false
    includeHtmlReports false
    includeVulnReports false
    includeJsonReports false
    includeCsvReports false
  }
}
publishers {
  dependencyCheck('**/dependency-check-report.xml') {
    healthLimits(3, 20)
    thresholdLimit('high')
    defaultEncoding('UTF-8')
    canRunOnFailed(true)
    useStableBuildAsReference(true)
    useDeltaValues(true)
    computeNew(true)
    shouldDetectModules(true)
    thresholds(
      unstableTotal: [all: 1, high: 2, normal: 3, low: 4],
      failedTotal: [all: 5, high: 6, normal: 7, low: 8],
      unstableNew: [all: 9, high: 10, normal: 11, low: 12],
      failedNew: [all: 13, high: 14, normal: 15, low: 16]
      )
  }
```

HOW BAD CAN IT BE?



SPECIAL BONUS TOOL: NPM AUDIT!



OWASP SONARQUBE RULES

OWASP SONARQUBE RULES

Benefits

Several “suggested” rulesets to add to your static code analysis to catch common security issues before they can ever be merged in.

How it works

Integrates into Sonarqube natively as a ruleset. Performs static code analysis to see if the code violates any of the rules and flags as necessary. If running SonarQube as part of your build pipeline (and if you aren’t…) it’ll raise violations as critical allowing you to fail builds or fail merge checks.

Implementations

- Java/Groovy/C#/ Javascript/Typescript/Ruby/Python/PHP/Swift/C#/

SONARQUBE RULES IN ACTION

Quality Gates, in line reporting, corrective action

Conditions

Only project measures are checked against thresholds. Sub-projects, directories and files are ignored. More

Metric	Over Leak Period	Operator	Warning	Error
Bugs	No	is greater than	0	5
Security Rating	Never	is worse than	A	B
Vulnerabilities	No	is greater than	0	0

Projects

Every project not specifically associated to a quality gate will be associated to this one by default.

```
public Document callAPI(String url) throws IOException {
        Document doc = null;
        HttpClient httpClient = new DefaultHttpClient();
```

DefaultHttpClient with default constructor is not compatible with TLS 1.2 — 3 months ago — L202
Vulnerability — Major — Open — Not assigned — cryptography, owasp-a6

```
        HttpGet get = new HttpGet(url);
```

Concatenating user-controlled input into a URL — 3 months ago — L203
Vulnerability — Major — Open — Not assigned — No tags

SonarQube analysis reported 15 issues

- 1 critical
- 14 minor

Watch the comments in this conversation to review them.

ZED ATTACK PROXY

ZED ATTACK PROXY

Benefits

Automatically hits your site/server probing for vulnerabilities dynamically.
A free DAST (Dynamic Application Security Tool).

How it works

Scans your machine with a configurable set of rules looking for common vulnerabilities and outputs a report. Looks for all the common OWASP top 10 violations.

Implementations

Has a large, customizable rule set as well as integration points with Jenkins and prebuilt docker images as well as an official Jenkins plugin.

ZAP OPTIONS

- GUI – Java based, uses SWING
- CLI – Command line interface
- REST - API
- Jenkins Plugin

ZED ATTACK PROXY

Benefits

Automatically hits your site/server probing for vulnerabilities dynamically.

How it works

Scans your machine with a configurable set of rules looking for common vulnerabilities and outputs a report. Looks for all the common OWASP top 10 violations.

Implementations

Has a large, customizable rule set as well as integration points with Jenkins and prebuilt docker images as well as an official Jenkins plugin.

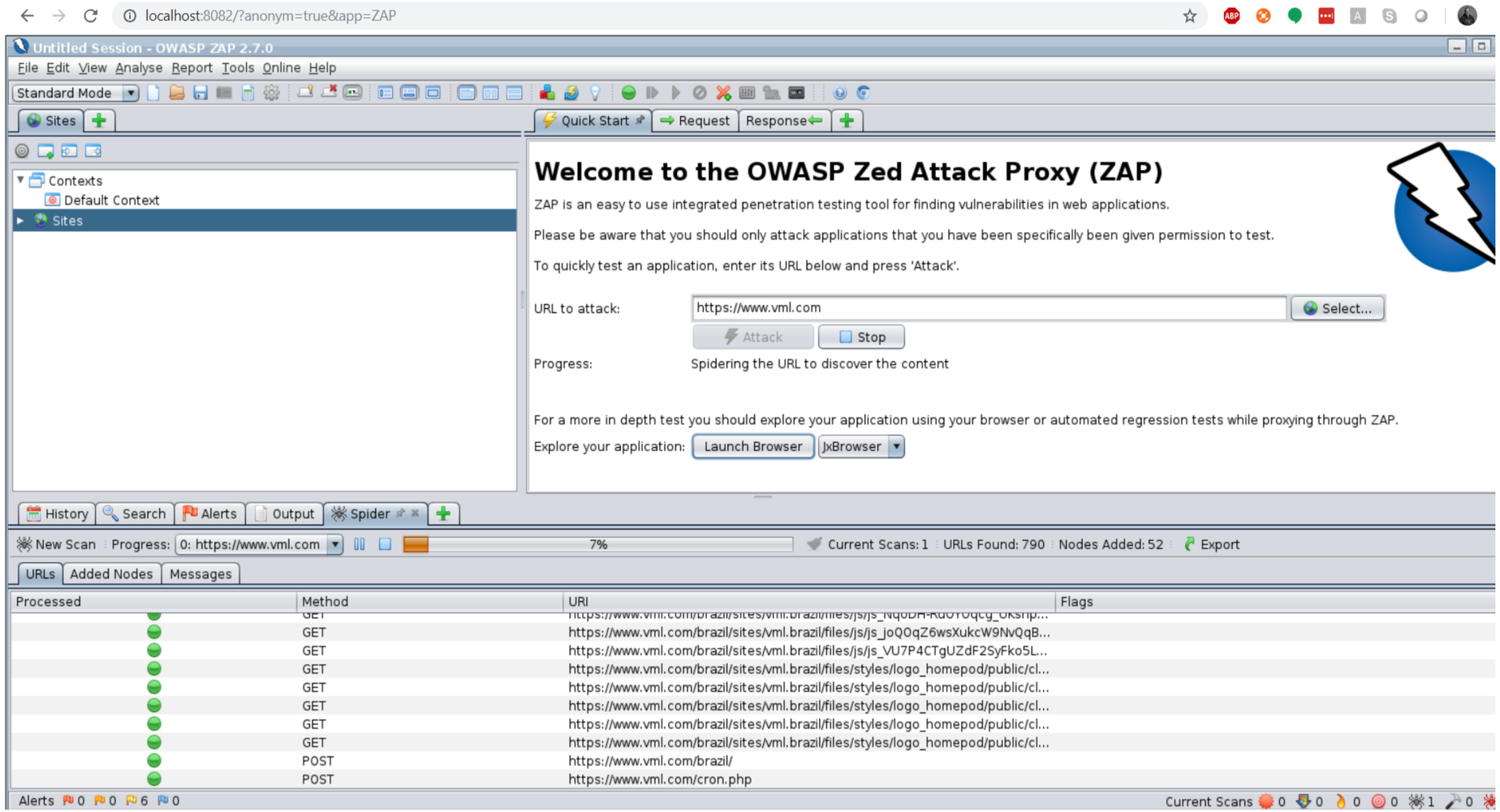
https://www.zaproxy.org/

ZAP IN ACTION

GUI

There is a GUI available to run a “traditional” scan but this
Isn’t very scriptable.

Great for exploratory testing and figuring out your options.

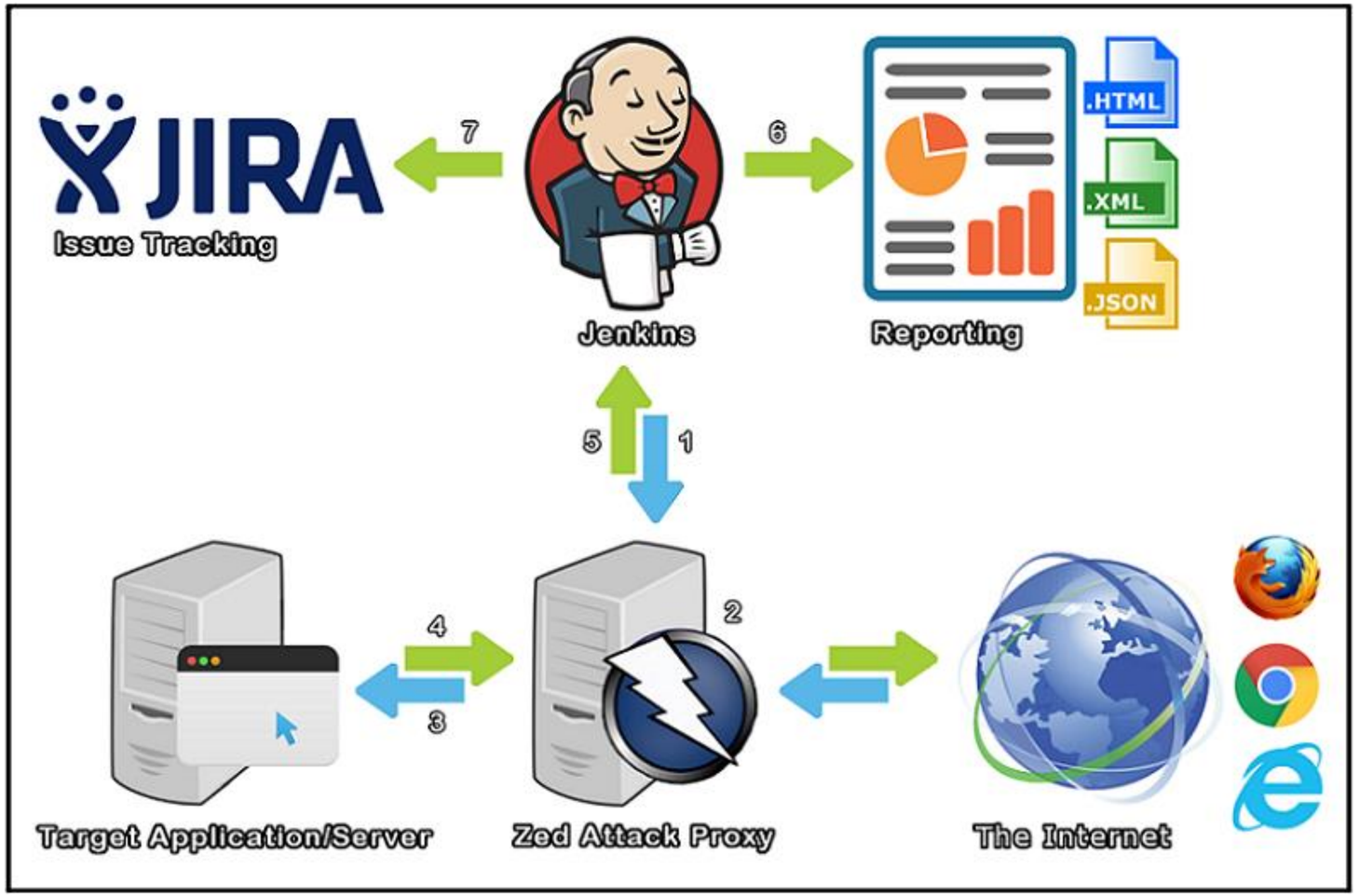


ZAP IN ACTION: JENKINS PLUGIN

Also an official Jenkins plugin.

Has Integration with JIRA.

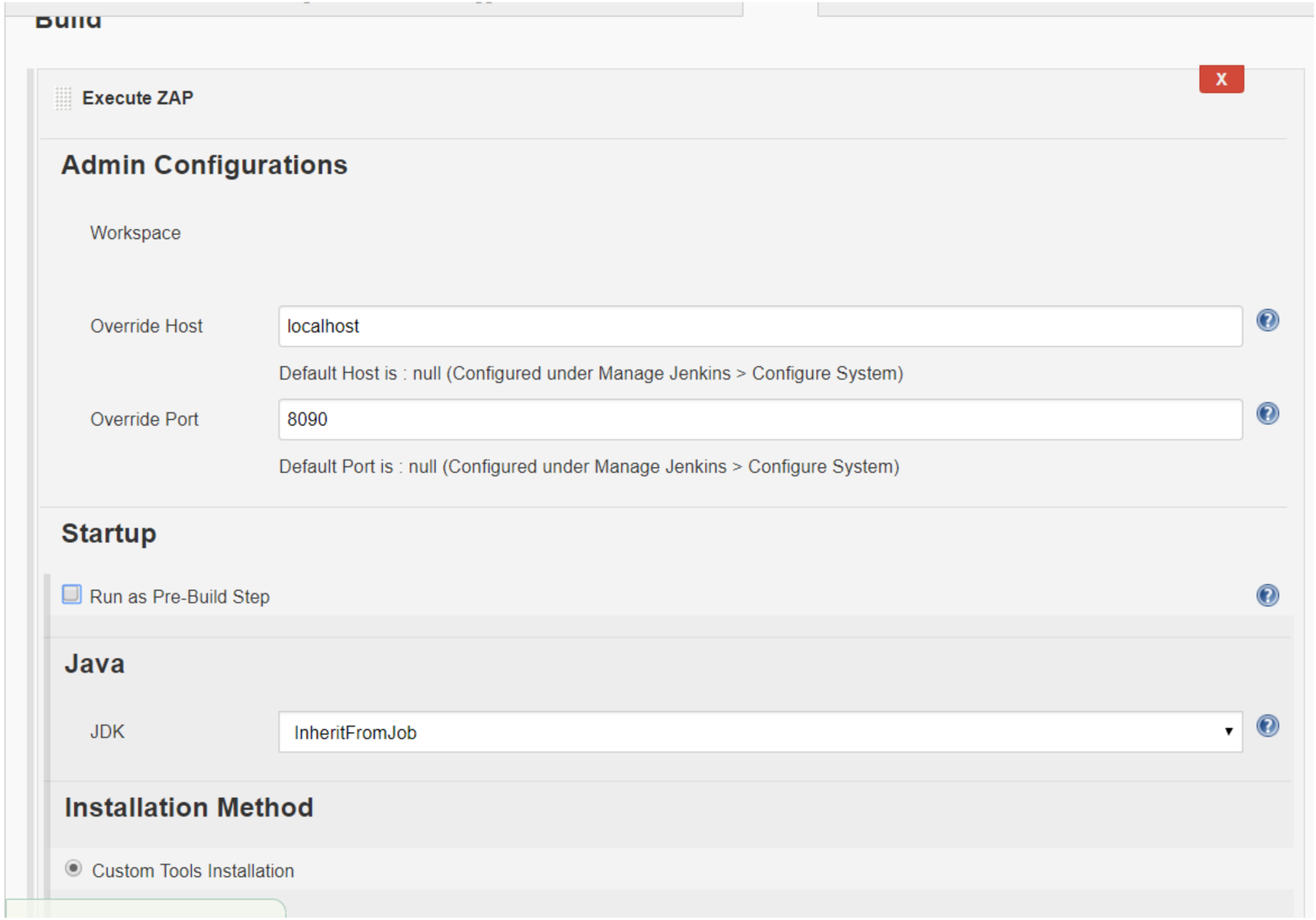
Requires full install of OWASP ZAP.



ZAP IN ACTION: JENKINS PLUGIN

Drawbacks

Great integration, but requires owasp zap server to be running on same container as Jenkins OR have Jenkins slave installed on your owasp container.



ZAP IN ACTION: ZAP CLI

Quick Scan

Runs a very basic scan, limited to one minute

```
C:\Users\bdinger>docker run -i owasp/zap2docker-stable zap-cli quick-scan --self-contained --start-options "-config api.
disablekey=true" http://owasp.localdev.com
[INFO]         Starting ZAP daemon
[INFO]         Running a quick scan for http://owasp.localdev.com
[INFO]         Issues found: 0
[INFO]         Shutting down ZAP daemon
```

Baseline

Completely safe, usually very quick.

Active Scan

Not safe, potentially legal / production impacting.

ZAP IN ACTION: ZAP-API

Invoke VIA API

ZAP api provides a rich feature set.

Best use case is in docker container setups.

Suggestion is to chain to your Jenkins instance and invoke when deployments are run.

ZAP IN ACTION: OPENAPI

ZAP Swagger Integration

Ability to scan using OpenAPI(Swagger) definitions to verify contracts are enforced

Excellent way to verify your API is properly secured.

https://www.zaproxy.org/docs/desktop/addons/openapi-support/

ZED ATTACK PROXY: TIPS

Run as the last stage of your pipeline

ZAP requires that the code is deployed & running before it can be tested like automated QA tests.

Or as a schedule task

In case the complexity of a build being deployed, then being failed back, is too risky for your environment you can instead run it on a set schedule and use the results of that to inform deployment decisions.

Lots of Resources

Zap consumes major resources running and will easily bring a machine to its knees. Provision it well and (suggested) to use a docker container and scale/destroy as needed. Don’t run it on same box as your Jenkins machine!

SSL ADVANCED FORENSIC TOOL (O-SAFT)

SSL ADVANCED FORENSIC TOOL

Benefits

SSL checking tool for the lazy. Checks for common SSL/TLS issues and cipher attacks.

How it works

A series of Perl scripts (or a docker image!) that executes prebuilt attacks and checks against your sites to ensure they are correctly configured.

Implementations

Docker Image / Perl Scripts / Tar Package -> https://github.com/OWASP/O-Saft

OSAFT IN ACTION: TLS SCAN

Docker Image

A docker image is provided all setup and ready to be executed for you.

Example commands:

docker run owasp/o-saft +check www.kcdc.info

docker run owasp/o-saft +hastls11 www.kcdc.info

docker run owasp/o-saft --help=commands

APPLICATION SECURITY VERIFICATION STANDARDS

ASVS

Benefits

A guidebook to securing your application stack.

How it works

A series of checklists and procedures covering the whole gamut of security to ensure your critical infrastructure (like your DevOps pipelines) is fully secured.

Implementations

Documentation available on OWASPs site -> https://owasp.org/www-project-application-security-verification-standard/

ASVS EXAMPLE AUDIT

#	Description	1	2	3	Since
2.1	Verify all pages and resources by default require authentication except those specifically intended to be public (Principle of complete mediation).	✓	✓	✓	1.0
2.2	Verify that all password fields do not echo the user's password when it is entered.	✓	✓	✓	1.0
2.4	Verify all authentication controls are enforced on the server side.	✓	✓	✓	1.0
2.6	Verify all authentication controls fail securely to ensure attackers cannot log in.	✓	✓	✓	1.0
2.7	Verify password entry fields allow, or encourage, the use of passphrases, and do not prevent long passphrases/highly complex passwords being entered.	✓	✓	✓	3.0
2.8	Verify all account identity authentication functions (such as update profile, forgot password, disabled / lost token, help desk or IVR) that might regain access to the account are at least as resistant to attack as the primary authentication mechanism.	✓	✓	✓	2.0
2.9	Verify that the changing password functionality includes the old password, the new password, and a password confirmation.	✓	✓	✓	1.0
2.12	Verify that all suspicious authentication decisions are logged. This should include requests with relevant metadata needed for security investigations.		✓	✓	2.0
2.13	Verify that account passwords make use of a sufficient strength encryption routine and that it withstands brute force attack against the encryption routine.		✓	✓	3.0
2.16	Verify that credentials are transported using a suitable encrypted link and that all pages/functions that require a user to enter credentials are done so using an encrypted link.	✓	✓	✓	3.0

OWASP CHEAT SHEET SERIES

CHEAT SHEET SERIES

Benefits

Helps prevent against misconfiguration by providing vetted documentation

How it works

A series of cheat sheets to handle common configuration tasks – like how to store passwords, handle session, access control, logging, key management, secure programming in various languages, etc

Implementations

Documentation available on OWASPs site ->

https://cheatsheetseries.owasp.org/

REMEMBER: IF YOUR CI/CD STACK IS COMPROMISED YOU'VE LOST EFFECTIVE CONTROL OF ALL APPS

GO FORTH AND BUILD RUGGED SOFTWARE

RESOURCES

OWASP Top 10 Project:
https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

OWASP ASVS Page
https://owasp.org/www-project-application-security-verification-standard/

OWASP Docker Hub:
https://hub.docker.com/u/owasp/

OWASP Flagship Projects:
https://owasp.org/projects/

OWASP Dependency Check:
https://jeremylong.github.io/DependencyCheck/index.html

Docker & OWASP Setup:
https://blog.mozilla.org/fxtesteng/2016/05/11/docker-owasp-zap-part-one/

OWASP Zap automated scan list based on Swagger definitions
https://www.nearform.com/blog/zed-attack-proxy-in-a-ci-pipeline/ &
https://www.zaproxy.org/docs/desktop/addons/openapi-support/

THANK YOU.

https://github.com/BillDinger/ContinuousSecurity

@adazlian | wdinger@gmail.com