DevSecOps ()

{ Integrating && Maturing a Security Culture }

Philip Kulp, D.Sc.

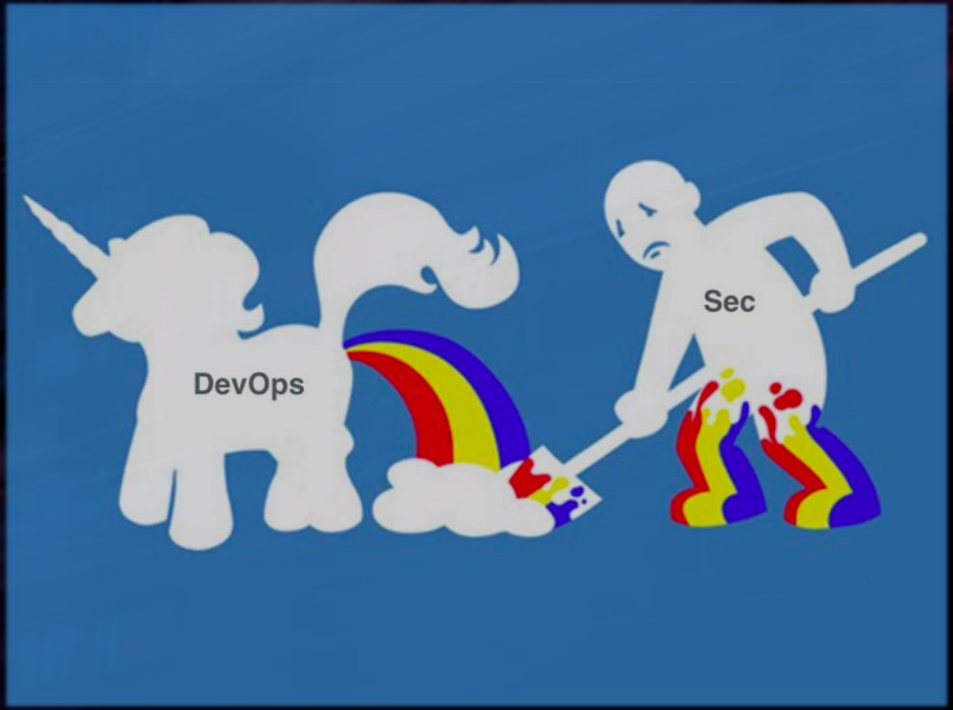
[Agenda]

- Important concepts for DevSecOps
- NIST SSDF
- DevSecOps pipeline
  - Planning and Awareness
  - Development
  - Delivery/Testing
  - Deployment
  - Continuous Monitoring

Photo by Glenn Carstens-Peters on Unsplash

[Culture Change]

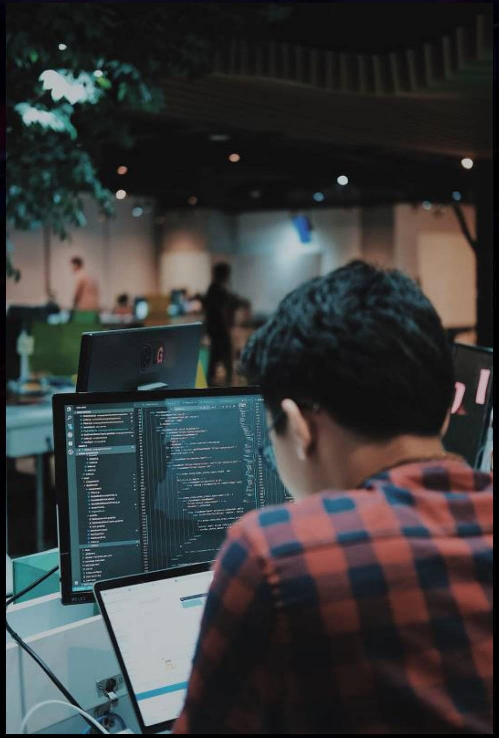
- Established DevOps
  - Security needs to integrate
  - Requires executive buy-in
- Developer culture and demands
  - Tight coding schedule
  - Limited testing schedule
- Testing not designed for security
  - Designed to pass, not fail
- Operations culture and demands
  - Keep everything running



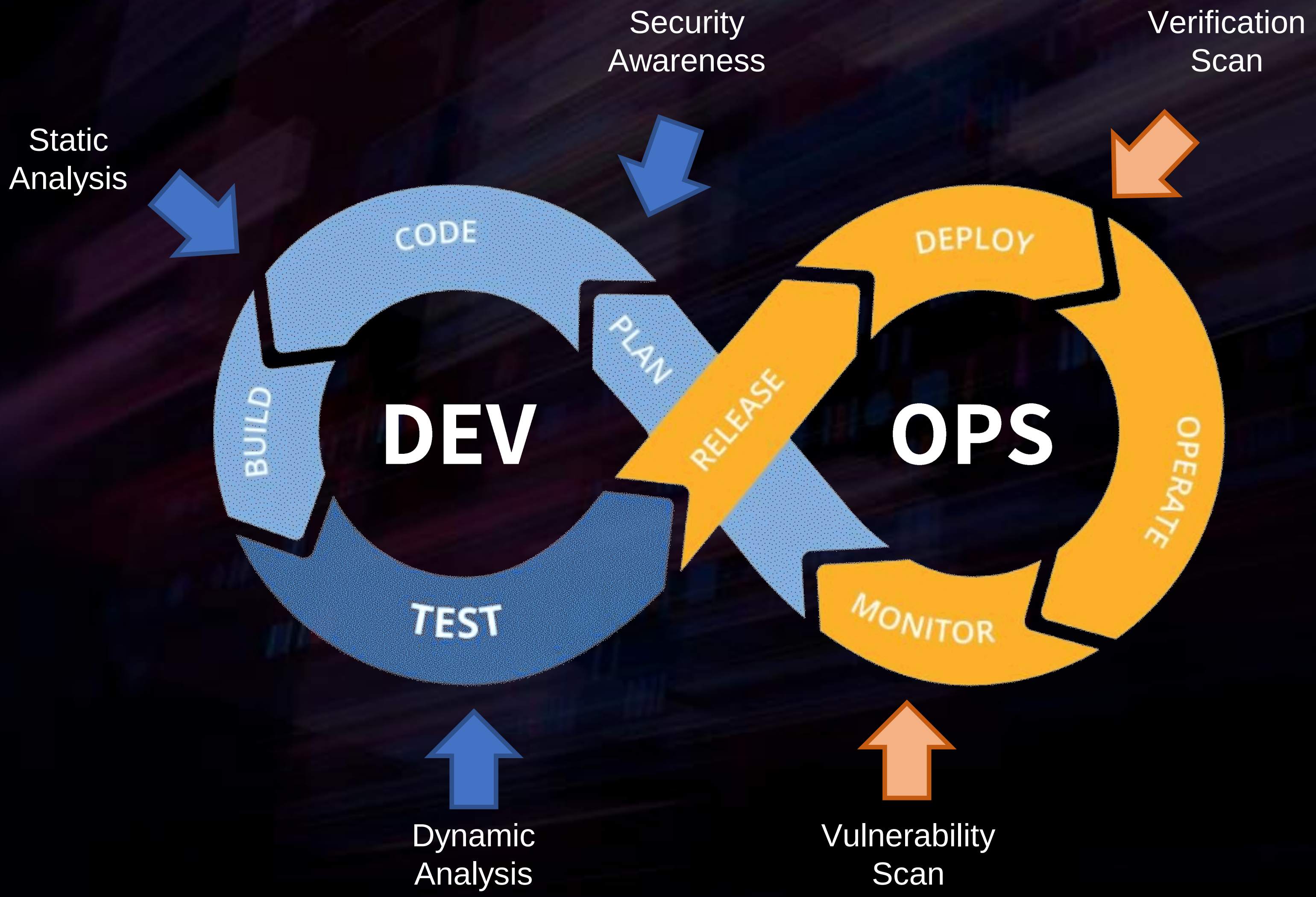
Pete Cheslok: https://twitter.com/petecheslock/status/595617204273618944

[Security Challenges]

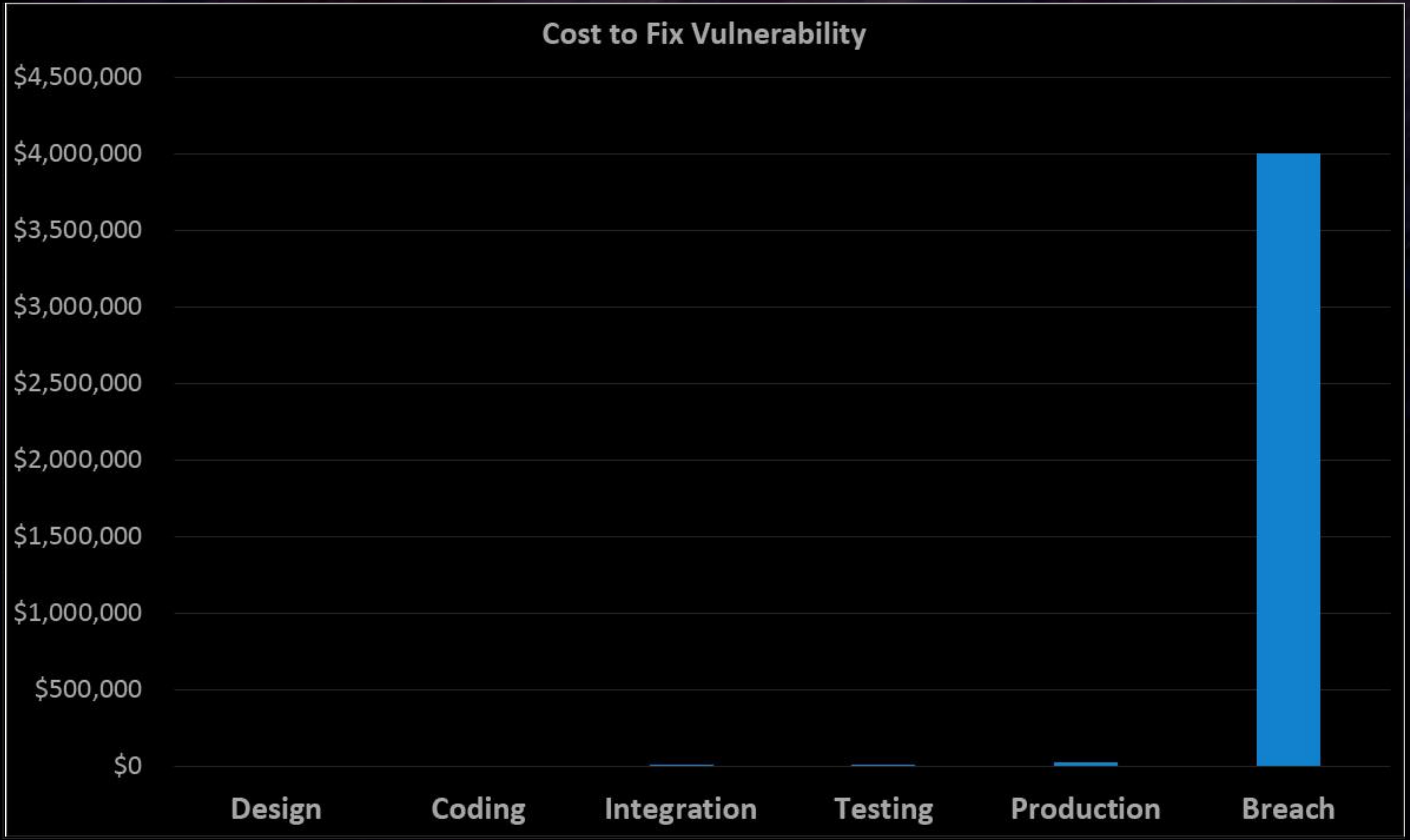
- Continuous Integration (CI) / Continuous Delivery (CD)
- Agile development
- Automation
  - Cannot slow down the existing process
  - Security tool integration with existing pipeline
- Auditors coding knowledge
  - Java, .Net, Python, PHP, Ruby
  - JSP, JavaScript, HTML5, CSS, and more



[DevOps + Sec]


Security Awareness
Verification Scan
Static Analysis
CODE
PLAN
BUILD
DEV
TEST
DEPLOY
RELEASE
OPS
OPERATE
MONITOR
Dynamic Analysis
Vulnerability Scan

[What is the Cost?]



IBM Security 2019 Cost of Data Breach Report

[Static Analysis]

- Whitebox testing
- Performed on uncompiled code
  - Also non-compiled code such as JavaScript
- More vulnerabilities found this way
- Labor intensive
  - Automation helps
  - Discovery of complex vulnerabilities difficult

[Dynamic Analysis]

- Blackbox testing
- Evaluation of running code
    - Code execution in the environment
- Static analysis looks for known bad patterns
- Dynamic analysis looks for known results
- Find in whitebox, confirm with blackbox

[Question for you]

- Are both SAST and DAST needed?
  - SAST may not see 3rd party libraries
  - DAST checks app server, configs, services, VM, container

Photo by Artem Maltsev on Unsplash

[3rd Party Libraries]

- Software Composition Analysis (SCA)
- May be added during the build
- May be part of the application server
- Vulnerabilities appear when chained
- Different version used in test and production
- Remote loaded JavaScript

Photo by Arisa Chattasa on Unsplash

[NIST SSDF]

- Secure Software Development
  - Prepare the Organization (PO)
  - Protect Software (PS)
  - Produce Well-Secured Software (PW)
  - Respond to Vulnerability Reports (RV)
- Define security requirements for software development
  - NIST SSDF PO.1 ✏
  - Policies, coding standards, periodic reviews
- Implement a supporting toolchain (PO.3) ✏

NIST Secure Software Development Framework (SSDF)

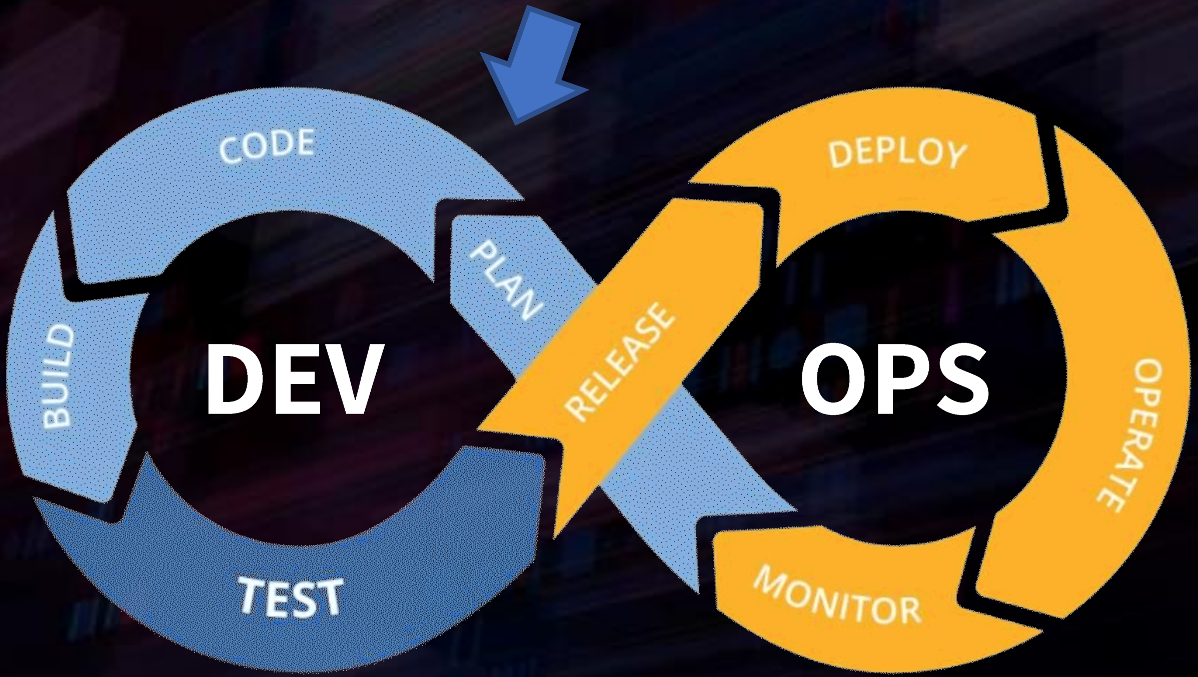
[NIST SSDF] *(cont’d)*

- Security needs a plan
  - Roles & responsibilities (PO.2) ✏
  - Explain the problem
  - Benefit
  - Costs and savings
  - Communication between teams
- Protect code from tampering (PS.1) ✏
  - Source repository
  - Version control
  - Signing, hashes

Photo by TK Hammonds on Unsplash

[Security Before Development]

Security Awareness



[Security Impact]

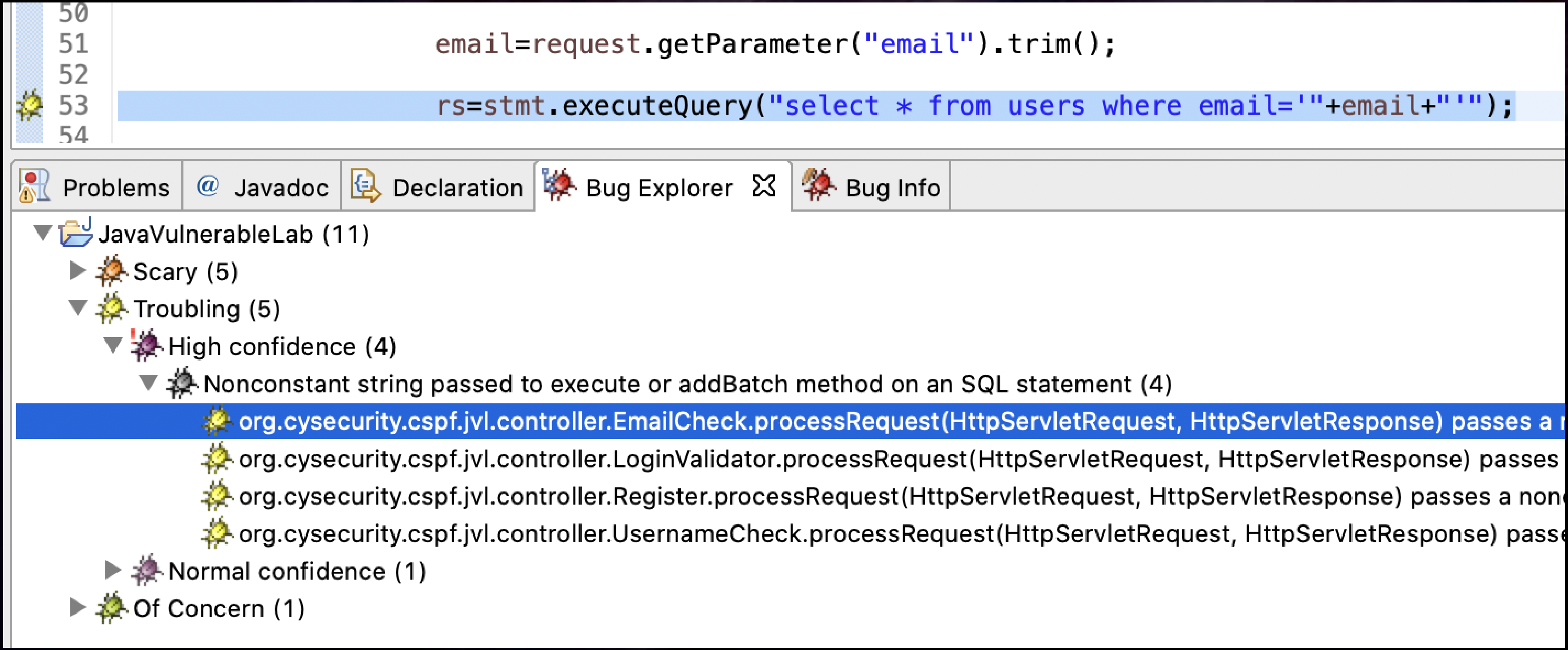
- Key Performance Indicators (PO.4) ✏
- Measure to
  - Quantify risk
  - Track integration and impacts
- DevOps has their own
  - Need distinct security metrics
  - Also need to measure security impact
- Compare delta
  - Before security feature added
  - After introduction to pipeline

Photo by Moritz Mentges on Unsplash

[Before Coding Begins]

- IDE plugins
  - Spotbugs (Java)
  - Puma Scan (.Net)
  - SonarLint
- Secure coding standards
  - Carnegie Mellon (CERT): Software Engineering Institute
  - OWASP Secure Coding Practices
- Code review procedures (PW.7) ✏
  - Review own code
  - Peer review & tools to facilitate collaboration
  - Document lessons learned

[IDE: SpotBugs]



```
                    email=request.getParameter("email").trim();

                    rs=stmt.executeQuery("select * from users where email='"+email+"'");

```

- JavaVulnerableLab (11)
  - Scary (5)
  - Troubling (5)
    - High confidence (4)
      - Nonconstant string passed to execute or addBatch method on an SQL statement (4)
        - org.cysecurity.cspf.jvl.controller.EmailCheck.processRequest(HttpServletRequest, HttpServletResponse) passes a r
        - org.cysecurity.cspf.jvl.controller.LoginValidator.processRequest(HttpServletRequest, HttpServletResponse) passes
        - org.cysecurity.cspf.jvl.controller.Register.processRequest(HttpServletRequest, HttpServletResponse) passes a none
        - org.cysecurity.cspf.jvl.controller.UsernameCheck.processRequest(HttpServletRequest, HttpServletResponse) passe
    - Normal confidence (1)
  - Of Concern (1)

[Training & Certification]

- Secure coding exercises
  - RangeForce
  - Codebashing
- Certifications
  - SANS: Secure Coding, Secure DevOps, App Security
  - CMU CERT: Secure Coding, Engineering Software Assurance
  - ISC^2: Software Development (CSSLP)

[Contract Enforcement]

- Limited or unclear
  - "No bugs"
  - Secure code
- Need to specify requirement to fix
  - Specific timeline for mitigation
- Define use of 3rd party libraries
  - Specific requirement to maintain patching
- Compliance to specific coding standard
- Require secure coding certifications

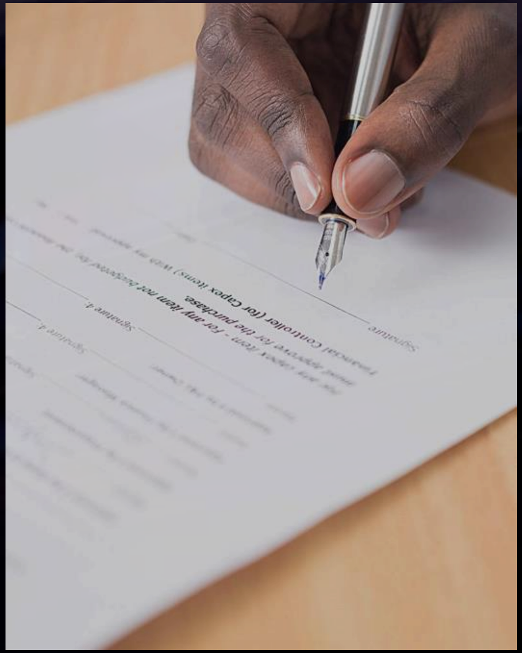


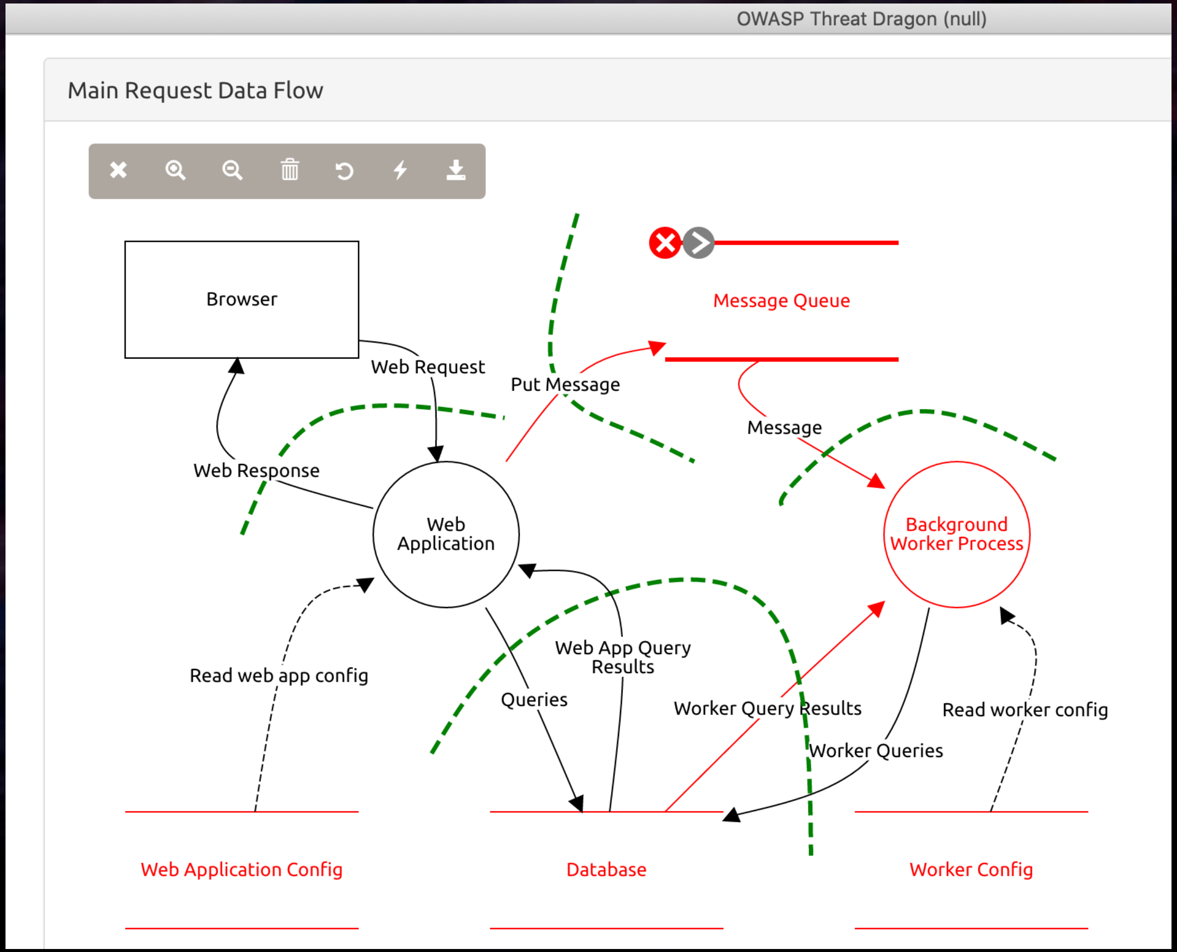
Photo by Cytonn Photography on Unsplash

[Cyber Staff]

- Understand culture change
  - Part of the process
  - Need to be agile
- Coding knowledge
  - Tools, IDEs
  - Terminology: commit, merge, build
- Ops knowledge
  - Deployment
  - Microservices / containers
  - Cloud architecture

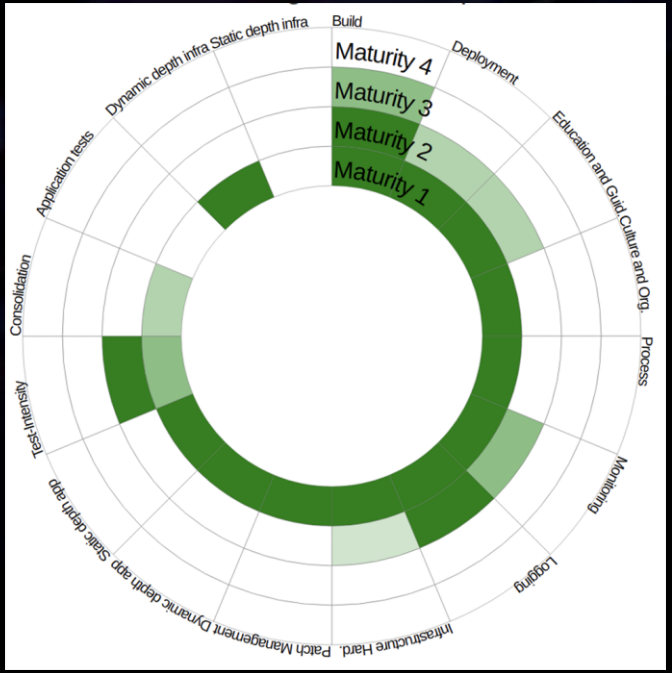
[Threat Model]

- Develop threat catalog (PW.1) ✎
  - OWASP Threat Dragon
- Associate threats to risks (PW.2) ✎
- Identify unique threats



[OWASP DevSecOps Maturity Model]

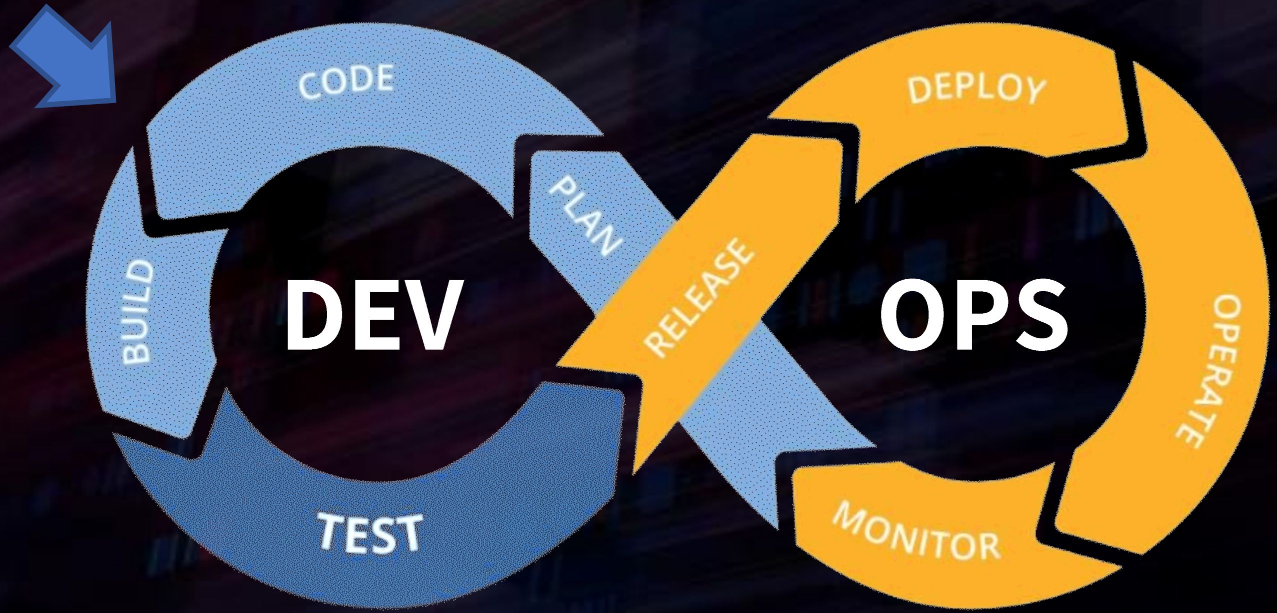
- 16 Dimensions
  - Build, deployment, process, monitoring, infrastructure hardening, and more
- Level 1
  - Basic understanding of security practices
- Level 2
  - Understanding of security practices
- Level 3
  - High understanding of security practices
- Level 4
  - Advanced understanding of security practices at scale



https://owasp.org/www-project-devsecops-maturity-model/

[Development Security]

Static App Security Test (SAST)
Software Composition Analysis



[Common Weakness Enumeration]

- CWE Top 25 Most Dangerous Software Errors
  - Common Vulnerabilities and Exposures (CVE) data
  - CWE mappings from NIST National Vulnerability Database (NVD)

Rank	ID	Name	Score
1	CWE-119	Improper Restriction of Operations within the Bound of a Memory Buffer	75.56
2	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	45.69
3	CWE-20	Improper Input Validation	43.61
4	CW-200	Information Exposure	32.12

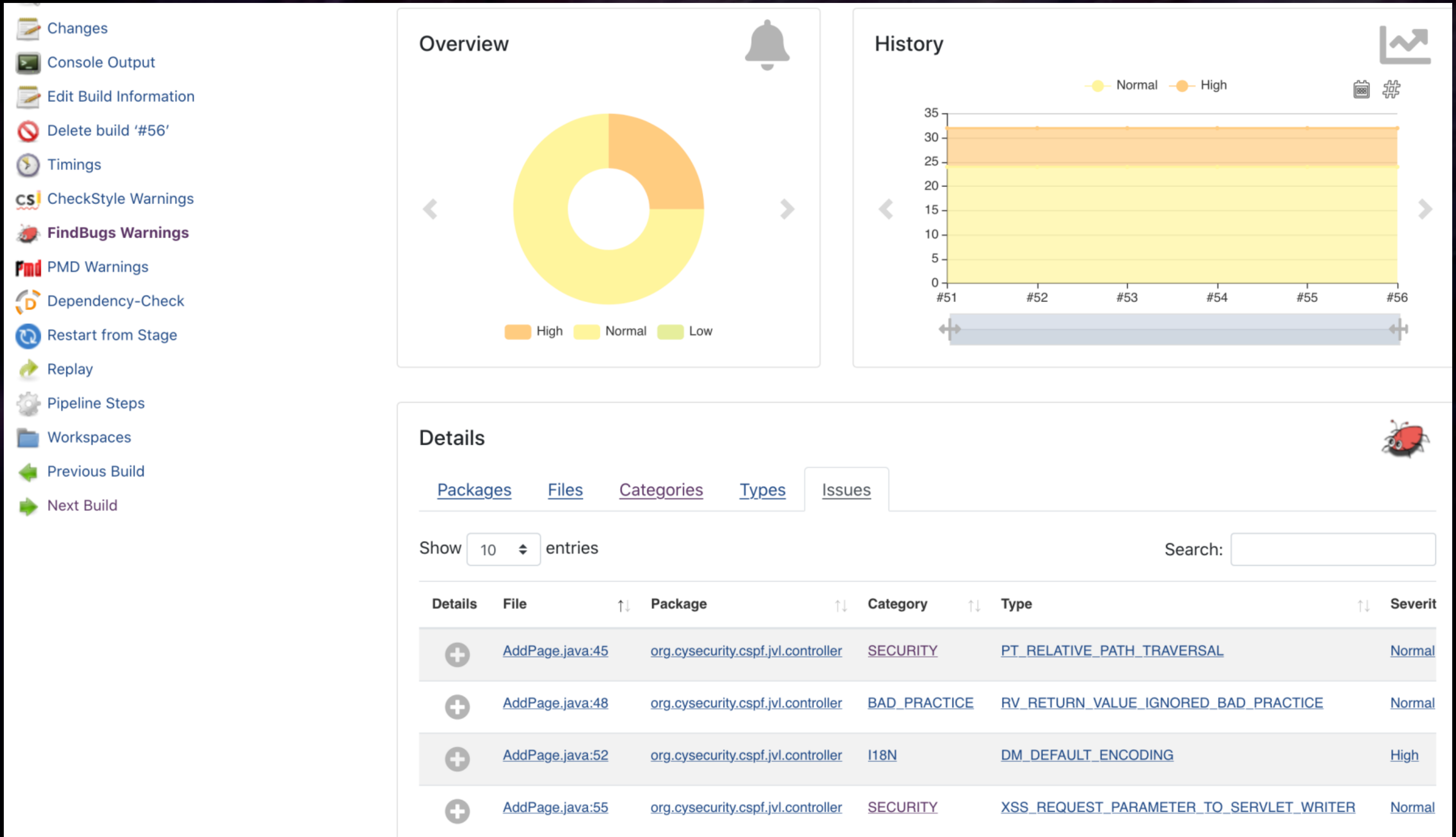
https://cwe.mitre.org/top25/archive/2019/2019_cwe_top25.html

[Static Analysis Tools] (PW.5)

- Java
  - PMD
  - SpotBugs
- Puma Scan (.Net)
- OWASP Dependency Check
- Commercial
  - Veracode, Micro Focus, Checkmarx

https://pumascan.com
https://spotbugs.github.io

[Jenkins: SAST]



Details	File	Package	Category	Type	Severit
⊕	AddPage.java:45	org.cysecurity.cspf.jvl.controller	SECURITY	PT_RELATIVE_PATH_TRAVERSAL	Normal
⊕	AddPage.java:48	org.cysecurity.cspf.jvl.controller	BAD_PRACTICE	RV_RETURN_VALUE_IGNORED_BAD_PRACTICE	Normal
⊕	AddPage.java:52	org.cysecurity.cspf.jvl.controller	I18N	DM_DEFAULT_ENCODING	High
⊕	AddPage.java:55	org.cysecurity.cspf.jvl.controller	SECURITY	XSS_REQUEST_PARAMETER_TO_SERVLET_WRITER	Normal

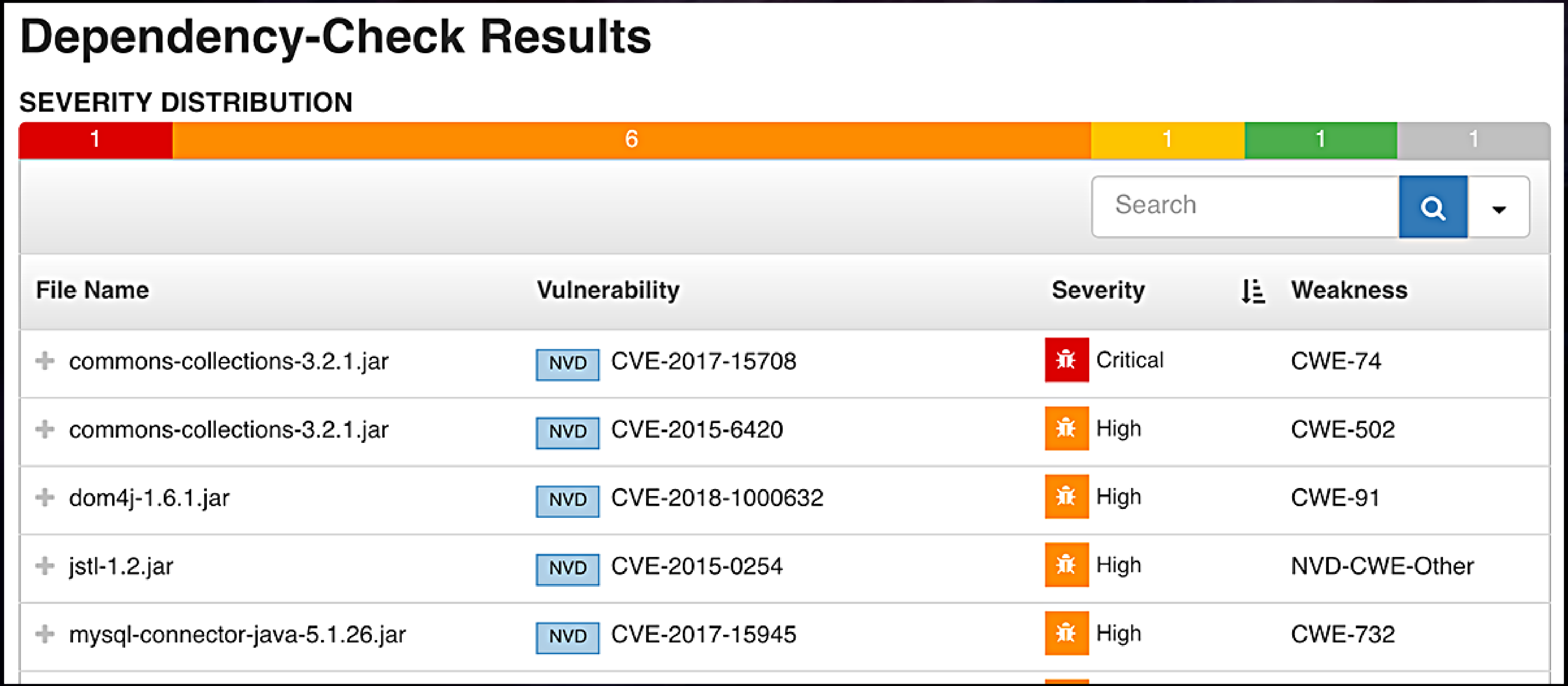
[Supply Chain Security]

- Software Composition Analysis (SCA)
- 3rd Party Libraries
  - Java Maven *(pom.xml)*
  - .Net NuGet *(.nuspec)*
  - Python PIP *(requirements.xml)*
  - Node NPM *(package.json)* *[1, 2]*
- Git/other libraries
- Docker Hub public repository
- Create internal trusted repository (PW.4) ✏

[1] https://www.zdnet.com/article/hacking-20-high-profile-dev-accounts-could-compromise-half-of-the-npm-ecosystem/

[2] https://qz.com/646467/how-one-programmer-broke-the-internet-by-deleting-a-tiny-piece-of-code/

[Jenkins: SCA]



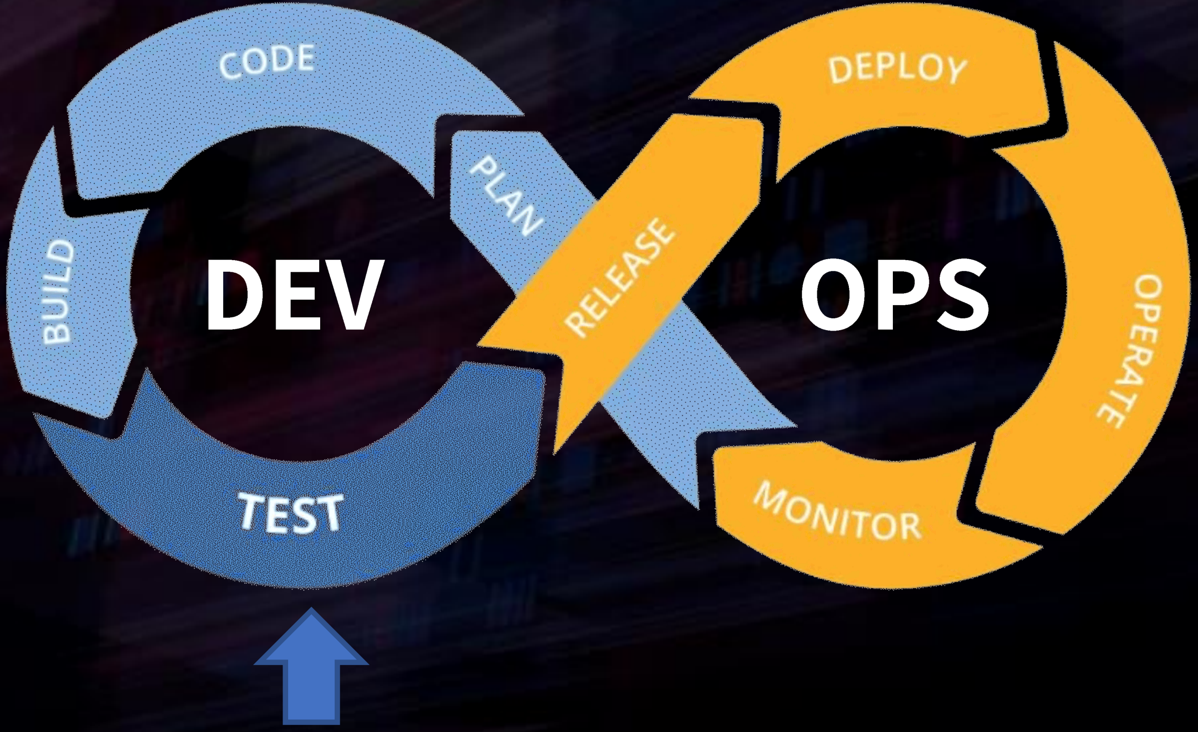
Dependency-Check Results

SEVERITY DISTRIBUTION

1 6 1 1 1

File Name	Vulnerability	Severity	Weakness
+ commons-collections-3.2.1.jar	NVD CVE-2017-15708	Critical	CWE-74
+ commons-collections-3.2.1.jar	NVD CVE-2015-6420	High	CWE-502
+ dom4j-1.6.1.jar	NVD CVE-2018-1000632	High	CWE-91
+ jstl-1.2.jar	NVD CVE-2015-0254	High	NVD-CWE-Other
+ mysql-connector-java-5.1.26.jar	NVD CVE-2017-15945	High	CWE-732

[Delivery Security]



Dynamic App Security Test (DAST)
Compliance testing

[Scanning Tools]

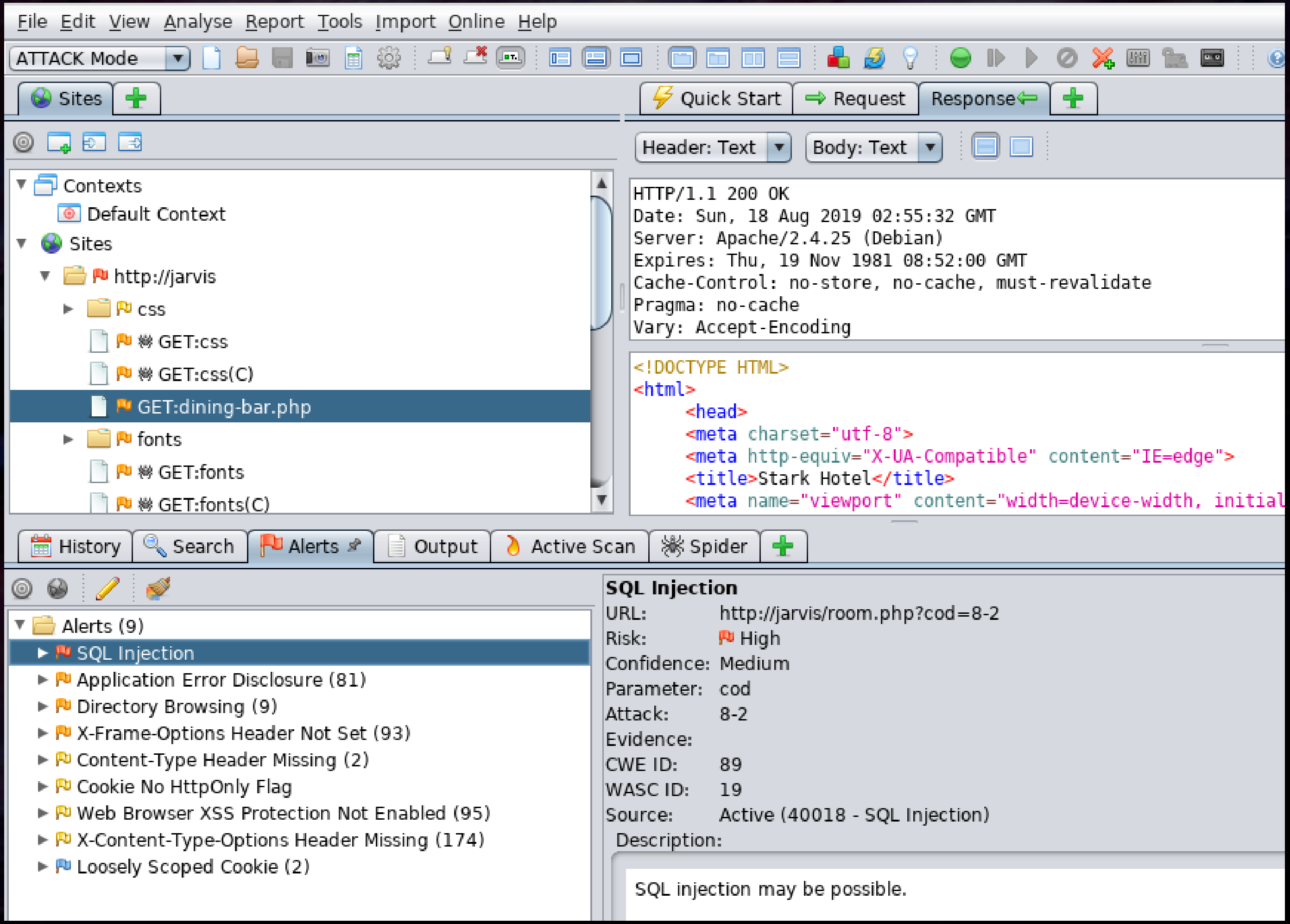
- Test executable code for vulnerabilities (PW.8) ✏
- Vulnerability scanning
  - Nexpose (opensource w/limitation)
  - OpenVAS
- WebApp scanning
  - Arachni
  - Nikto (CIRT)
  - ZAP (OWASP)
  - Xenotix XSS (OWASP)
- Nmap scripts
  - SSL/SSH cipher enumeration
  - Wordpress (themes, password brute, plugins)

https://owasp.org/www-project-zap/
https://nmap.org

[OWASP ZED Attack Proxy]

- STACK
  - Code
  - Web server
  - App server
  - HTTP methods
  - CMS

- Active scan
- Crawl
- Request editor
- Spider



```
HTTP/1.1 200 OK
Date: Sun, 18 Aug 2019 02:55:32 GMT
Server: Apache/2.4.25 (Debian)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
```

```
<!DOCTYPE HTML>
<html>
    <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Stark Hotel</title>
    <meta name="viewport" content="width=device-width, initial
```

```
Alerts (9)
  SQL Injection
  Application Error Disclosure (81)
  Directory Browsing (9)
  X-Frame-Options Header Not Set (93)
  Content-Type Header Missing (2)
  Cookie No HttpOnly Flag
  Web Browser XSS Protection Not Enabled (95)
  X-Content-Type-Options Header Missing (174)
  Loosely Scoped Cookie (2)
```

SQL Injection

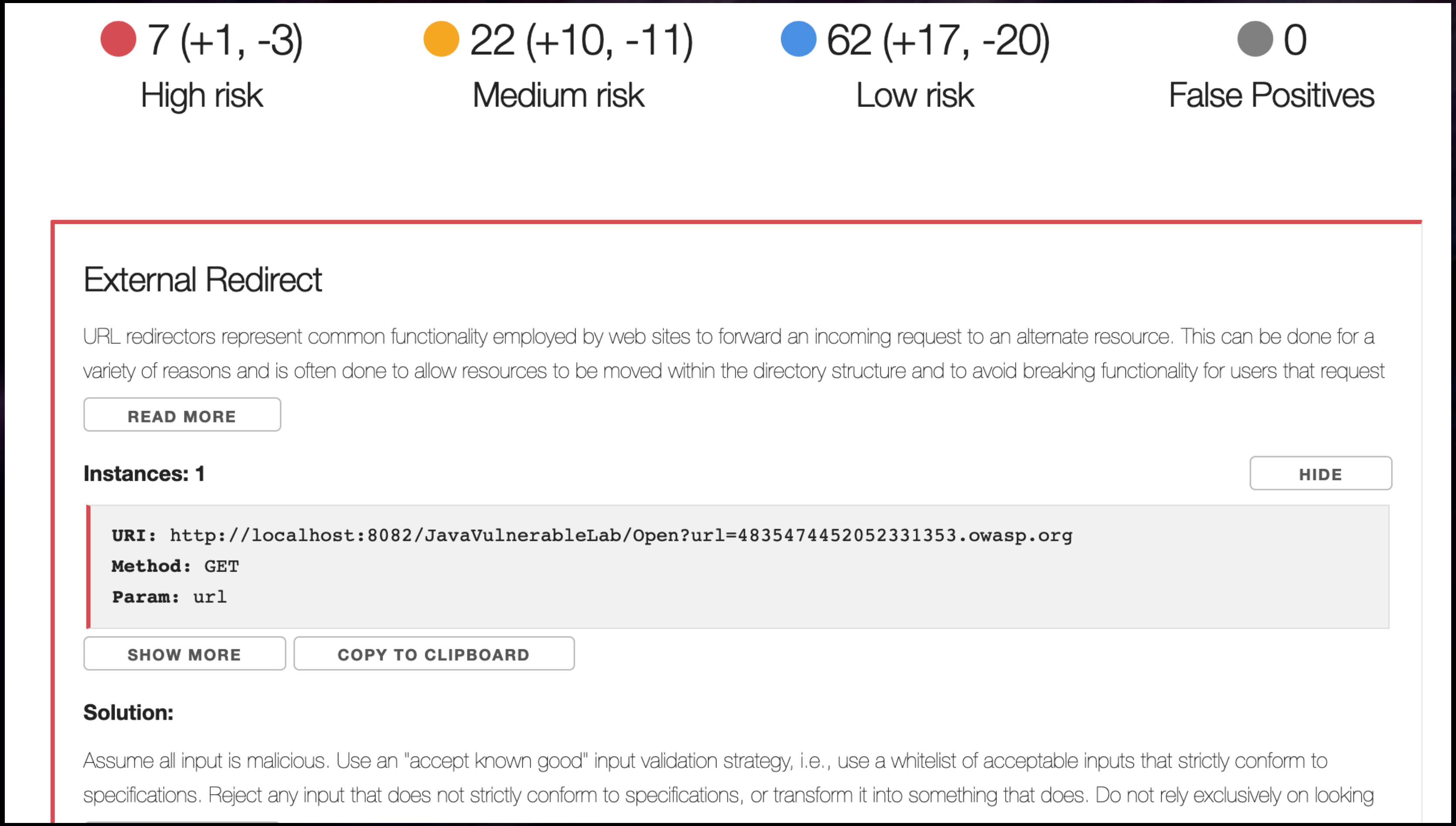
URL:	http://jarvis/room.php?cod=8-2
Risk:	High
Confidence:	Medium
Parameter:	cod
Attack:	8-2
Evidence:	
CWE ID:	89
WASC ID:	19
Source:	Active (40018 - SQL Injection)

Description:

SQL injection may be possible.

https://owasp.org/www-project-zap/

[Jenkins: DAST]



● 7 (+1, -3)	● 22 (+10, -11)	● 62 (+17, -20)	● 0
High risk	Medium risk	Low risk	False Positives

External Redirect

URL redirectors represent common functionality employed by web sites to forward an incoming request to an alternate resource. This can be done for a variety of reasons and is often done to allow resources to be moved within the directory structure and to avoid breaking functionality for users that request

Instances: 1

```
URI: http://localhost:8082/JavaVulnerableLab/Open?url=4835474452052331353.owasp.org
Method: GET
Param: url
```

Solution:

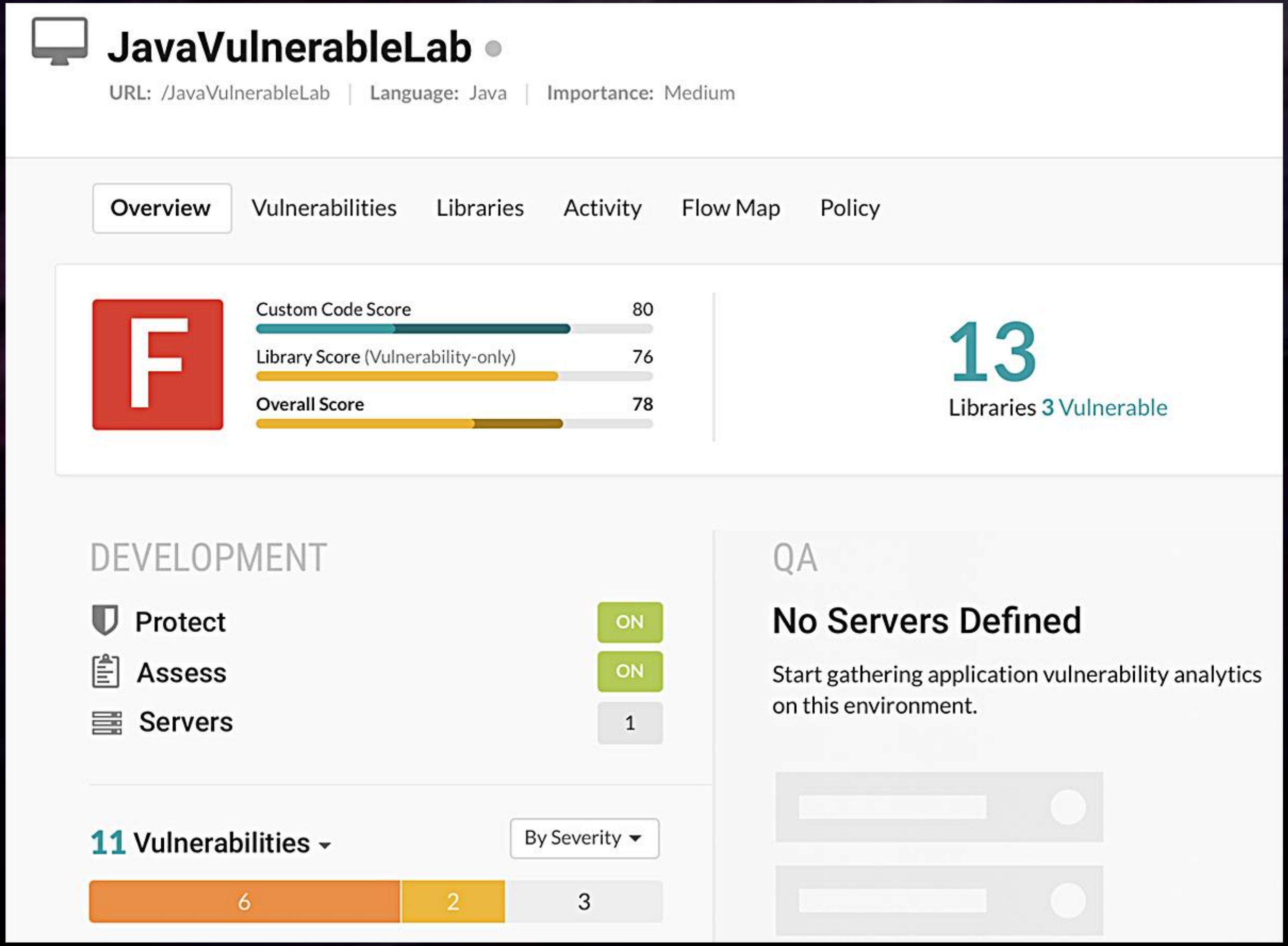
Assume all input is malicious. Use an "accept known good" input validation strategy, i.e., use a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. Do not rely exclusively on looking

[Interactive Application Security Testing]

- Identifies vulnerabilities in application
  - Not a scanner
- Uses instrumentation to gather data
  - Events sent to analysis engine
  - Rules engine identifies deviations
- Uses multiple data points
  - Request data
  - Tokens
  - State changes

https://dzone.com/refcardz/introduction-to-iast

[IAST: Contrast Security]



JavaVulnerableLab ●

URL: /JavaVulnerableLab | Language: Java | Importance: Medium

Custom Code Score 80

Library Score (Vulnerability-only) 76

Overall Score 78

13 Libraries 3 Vulnerable

DEVELOPMENT

Protect ON

Assess ON

Servers 1

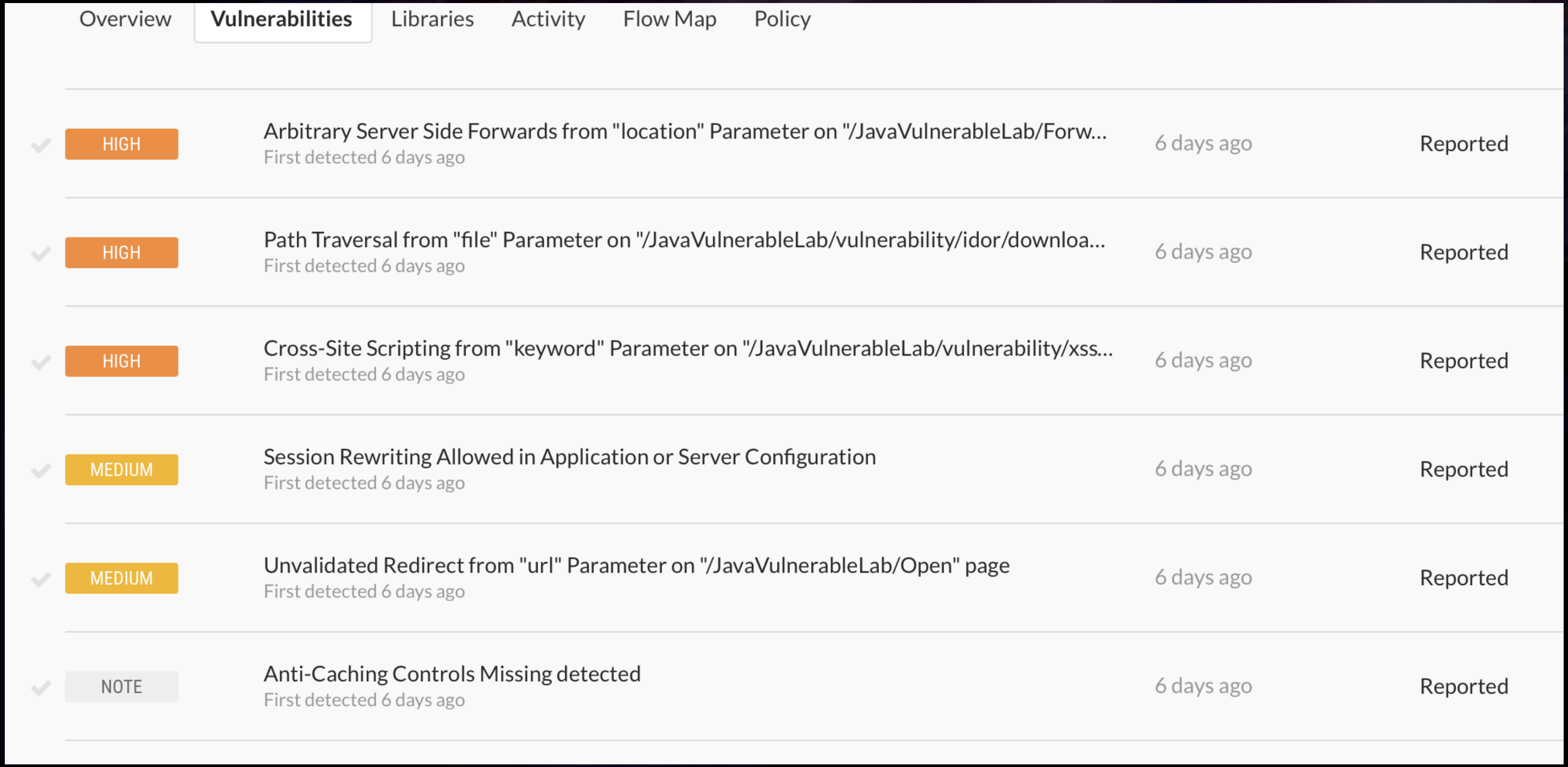
11 Vulnerabilities

QA

No Servers Defined

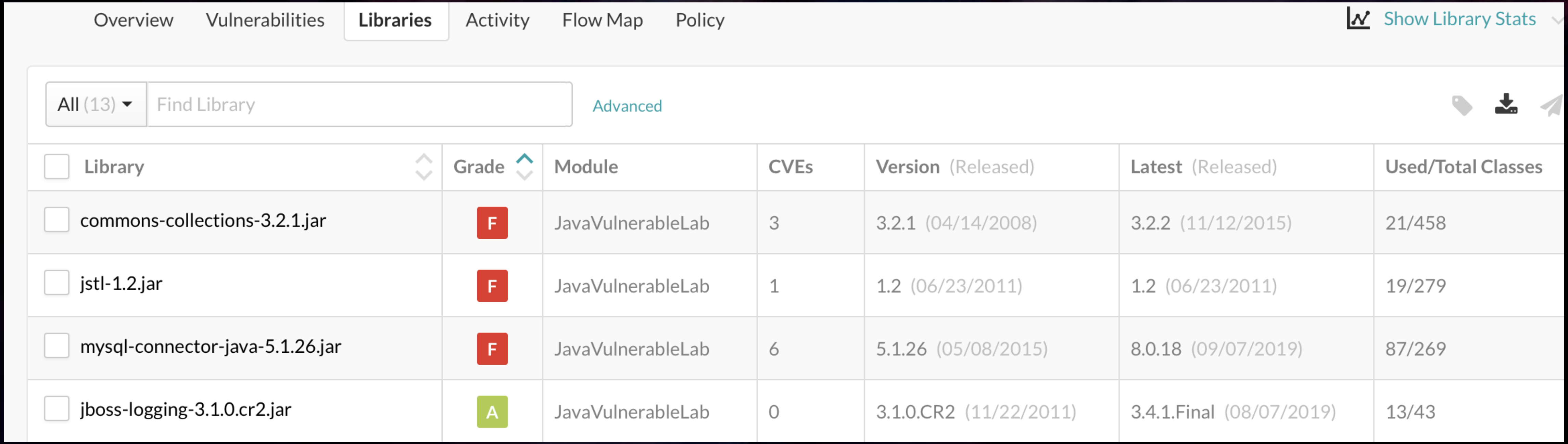
Start gathering application vulnerability analytics on this environment.

[IAST: Contrast Security]



✓	HIGH	Arbitrary Server Side Forwards from "location" Parameter on "/JavaVulnerableLab/Forw… First detected 6 days ago	6 days ago	Reported
✓	HIGH	Path Traversal from "file" Parameter on "/JavaVulnerableLab/vulnerability/idor/downloa… First detected 6 days ago	6 days ago	Reported
✓	HIGH	Cross-Site Scripting from "keyword" Parameter on "/JavaVulnerableLab/vulnerability/xss… First detected 6 days ago	6 days ago	Reported
✓	MEDIUM	Session Rewriting Allowed in Application or Server Configuration First detected 6 days ago	6 days ago	Reported
✓	MEDIUM	Unvalidated Redirect from "url" Parameter on "/JavaVulnerableLab/Open" page First detected 6 days ago	6 days ago	Reported
✓	NOTE	Anti-Caching Controls Missing detected First detected 6 days ago	6 days ago	Reported

[IAST: Contrast Security]



□ Library	Grade	Module	CVEs	Version (Released)	Latest (Released)	Used/Total Classes
□ commons-collections-3.2.1.jar	F	JavaVulnerableLab	3	3.2.1 (04/14/2008)	3.2.2 (11/12/2015)	21/458
□ jstl-1.2.jar	F	JavaVulnerableLab	1	1.2 (06/23/2011)	1.2 (06/23/2011)	19/279
□ mysql-connector-java-5.1.26.jar	F	JavaVulnerableLab	6	5.1.26 (05/08/2015)	8.0.18 (09/07/2019)	87/269
□ jboss-logging-3.1.0.cr2.jar	A	JavaVulnerableLab	0	3.1.0.CR2 (11/22/2011)	3.4.1.Final (08/07/2019)	13/43

[Question for You]

- Do you understand why app with IAST still needs to be scanned?
  - IAST tool only provides notifications
  - It does not generate scans

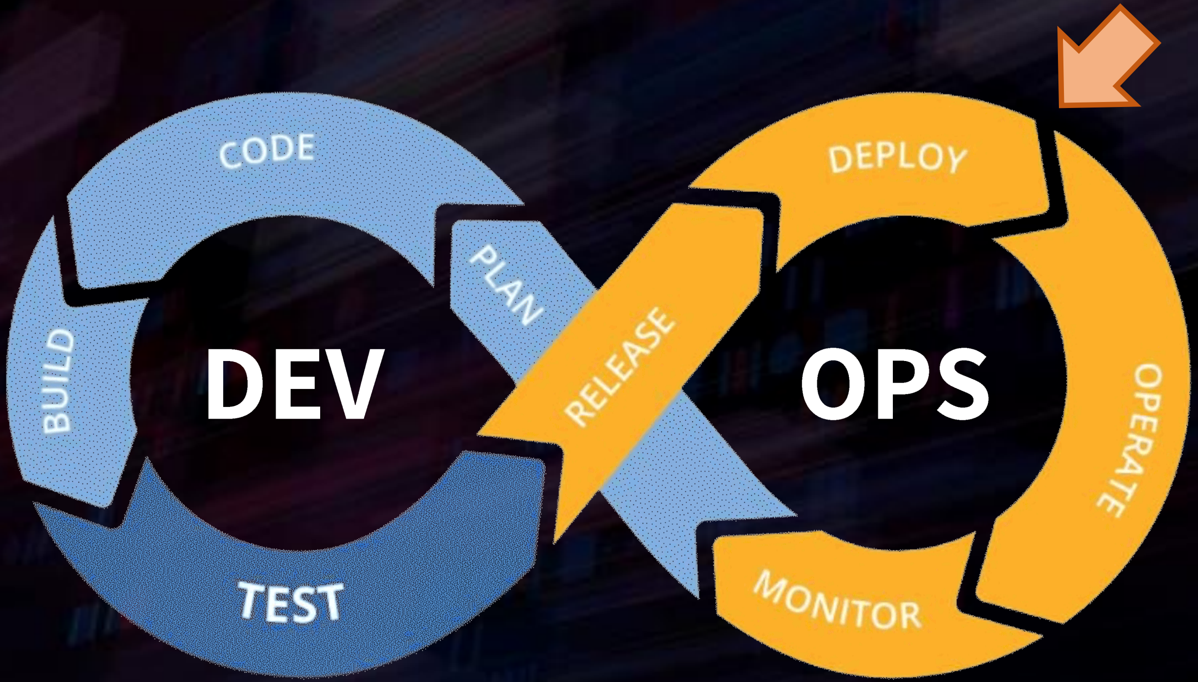
Photo by Artem Maltsev on Unsplash

[Business Logic Testing]

- DAST tools test patterns
    - SQL injection
    - XSS output
- Not good at testing authorization
    - Especially for all application roles
- Automate test cases for role privileges
    - Data
    - Actions
    - API endpoints

[Deploy Security]

Verification Scan
Infrastructure as Code (IaC)



Release Security Requirements:

Verify software release integrity (PS.2)
Protect each software release (PS.3)

[Infrastructure as Code]

- No manual changes in production
- Deployment rebuilds everything
  - Network
  - Instances
  - Containers
- Why do this?
  - Easier deployments
  - Improved resilience to disaster
  - Reduced system administration
  - Infrastructure versioning with code

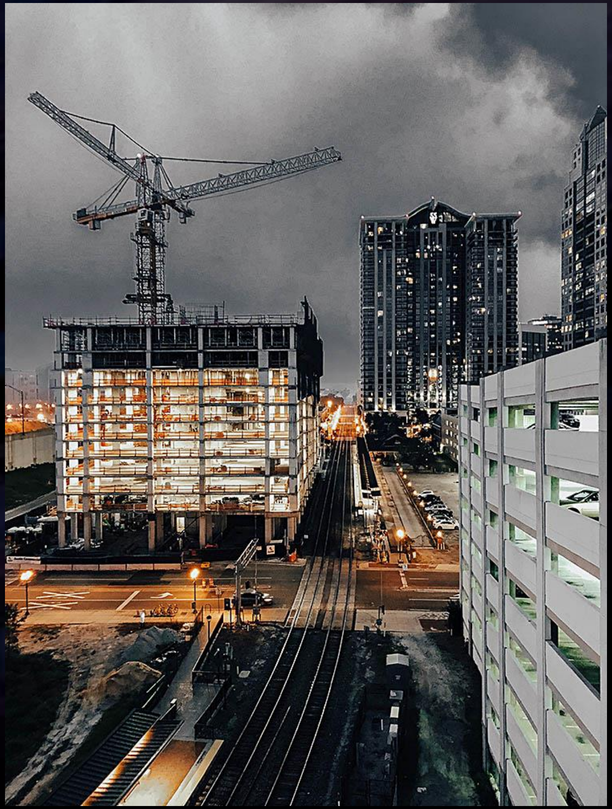
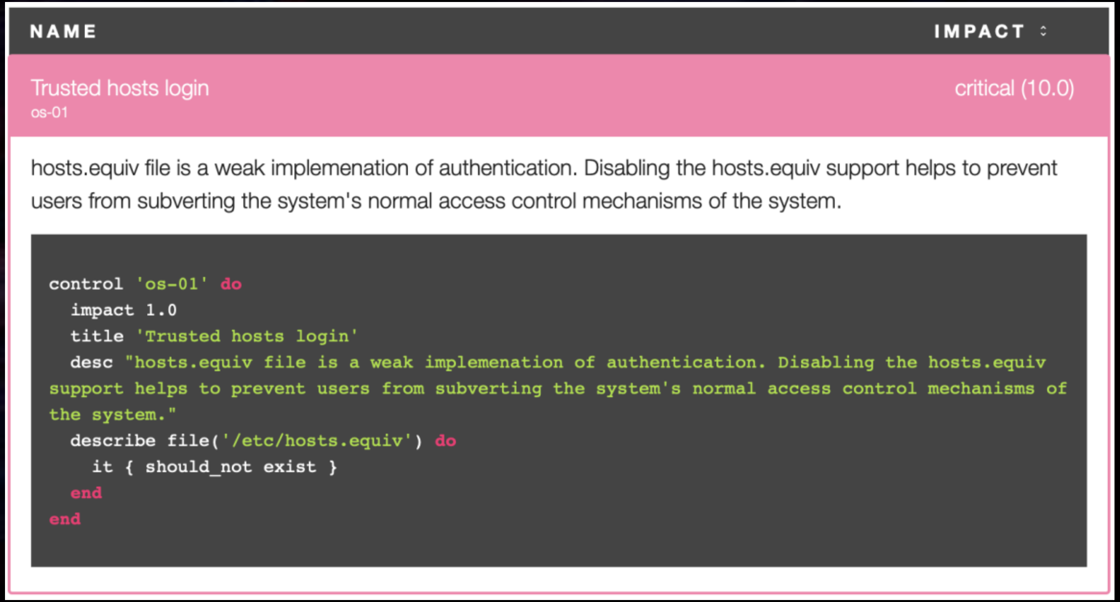


Photo by Nathan Waters on Unsplash

[DevSec Hardening Framework]

- Compliance as Code
  - PCI DSS, HIPPA, SOX, etc.
  - Secure setting by default (PW.9) ✎
- Configurations
  - Databases
  - Web Server
  - SSH, SSL, Docker, K8S
  - Linux, Windows
- Recipes
  - InSpec, Ansible, Chef, Puppet



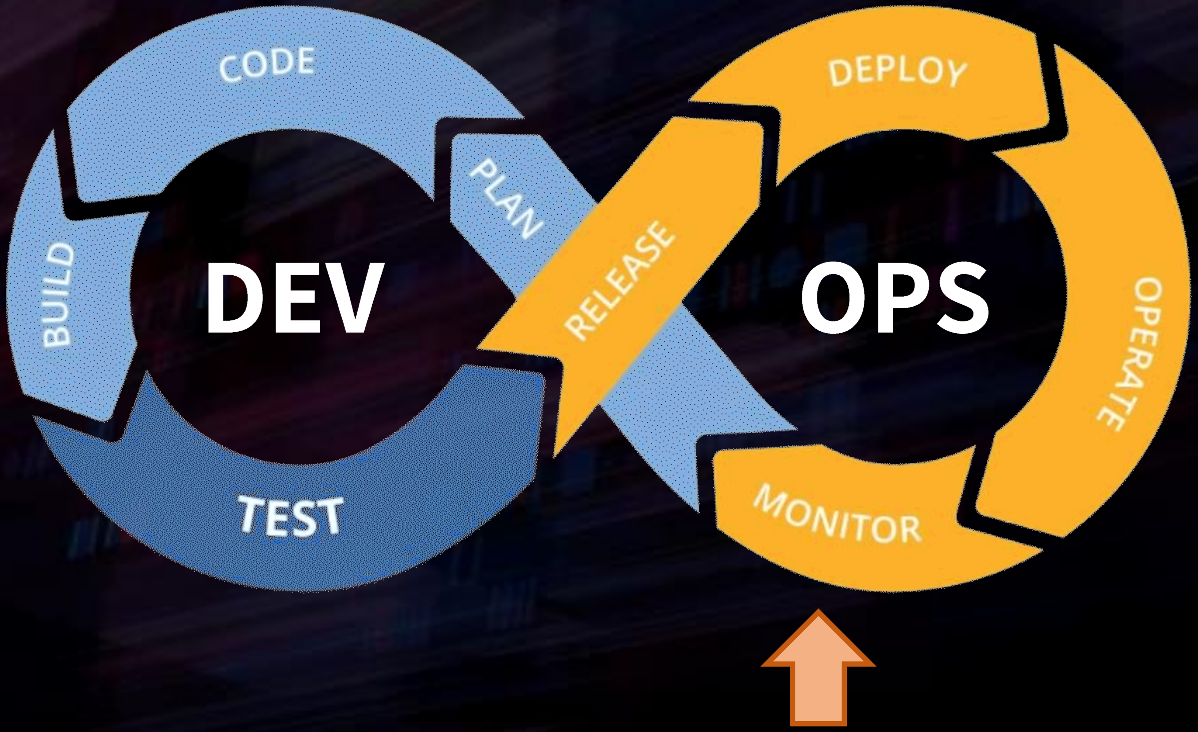
NAME	IMPACT ⇕
Trusted hosts login os-01	critical (10.0)

hosts.equiv file is a weak implemenation of authentication. Disabling the hosts.equiv support helps to prevent users from subverting the system's normal access control mechanisms of the system.

```
control 'os-01' do
  impact 1.0
  title 'Trusted hosts login'
  desc "hosts.equiv file is a weak implemenation of authentication. Disabling the hosts.equiv
support helps to prevent users from subverting the system's normal access control mechanisms of
the system."
  describe file('/etc/hosts.equiv') do
    it { should_not exist }
  end
end
```

https://www.inspec.io

[Security for O&M]



Continuous Monitoring

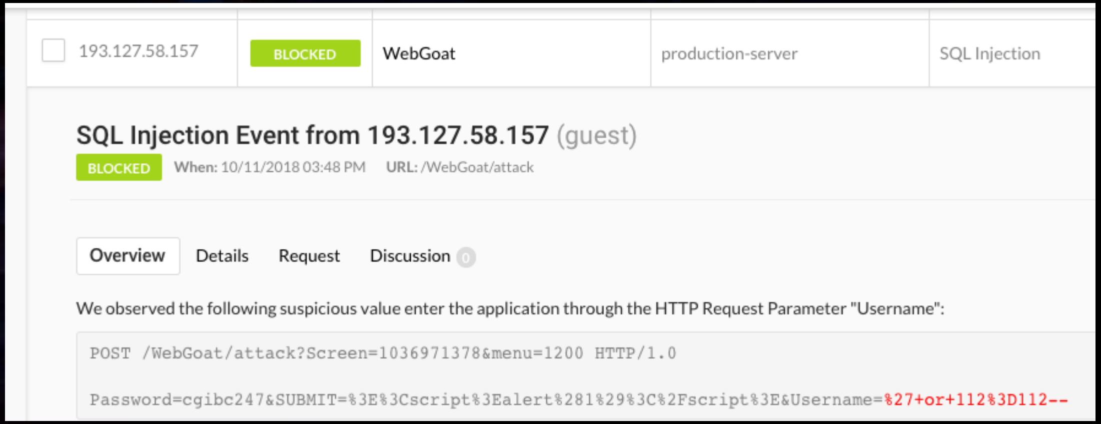
[SSDF: Respond to Vulnerability Report]

- Identify and confirm vulnerabilities (RV.1)
  - Establish a vulnerability response program
  - Monitor vulnerability databases
  - Confirm security toolchain
- Assess and Prioritize Remediation (RV.2)
  - Issue or bug tracking
- Identify root cause of vulnerabilities (RV.3)
  - Document root cause
  - Lessons learned
  - Implement changes to SSDF practices

NIST Secure Software Development Framework (SSDF)

[Runtime App Self-Protection]

- Prevent exploitation within app server
- Work in tandem with WAF
  - Doesn’t resolve app vulnerabilities
  - Block attack before reaching app
- Performance impact
- Differentiate attack success
- Contrast Protect
  - Community Edition (CE)



□ 193.127.58.157	BLOCKED	WebGoat	production-server	SQL Injection

SQL Injection Event from 193.127.58.157 (guest)

BLOCKED When: 10/11/2018 03:48 PM URL: /WebGoat/attack

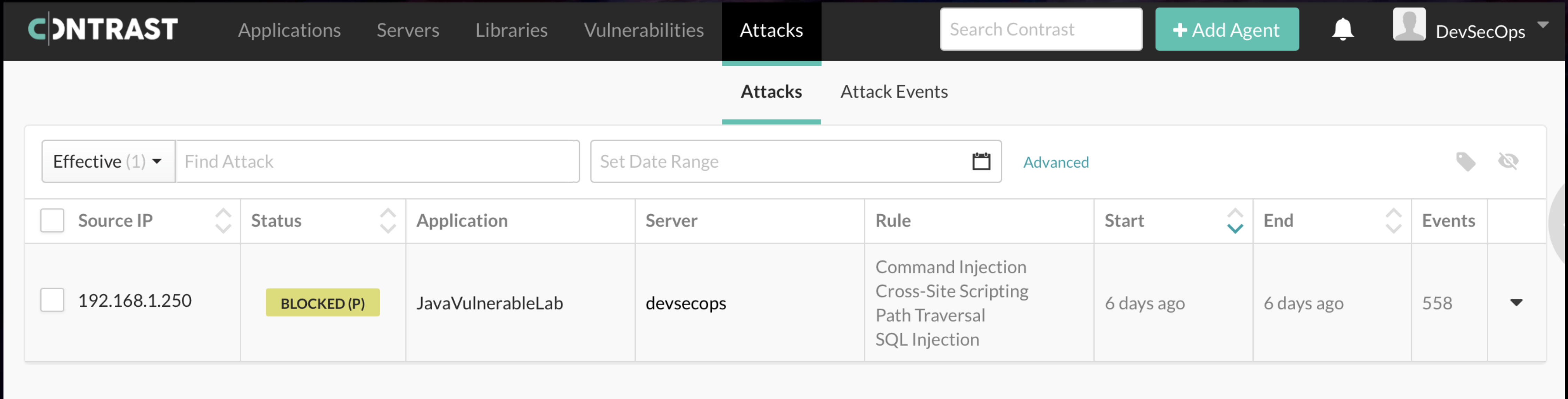
We observed the following suspicious value enter the application through the HTTP Request Parameter "Username":

```
POST /WebGoat/attack?Screen=1036971378&menu=1200 HTTP/1.0

Password=cgibc247&SUBMIT=%3E%3Cscript%3Ealert%281%29%3C%2Fscript%3E&Username=%27+or+112%3D112--
```

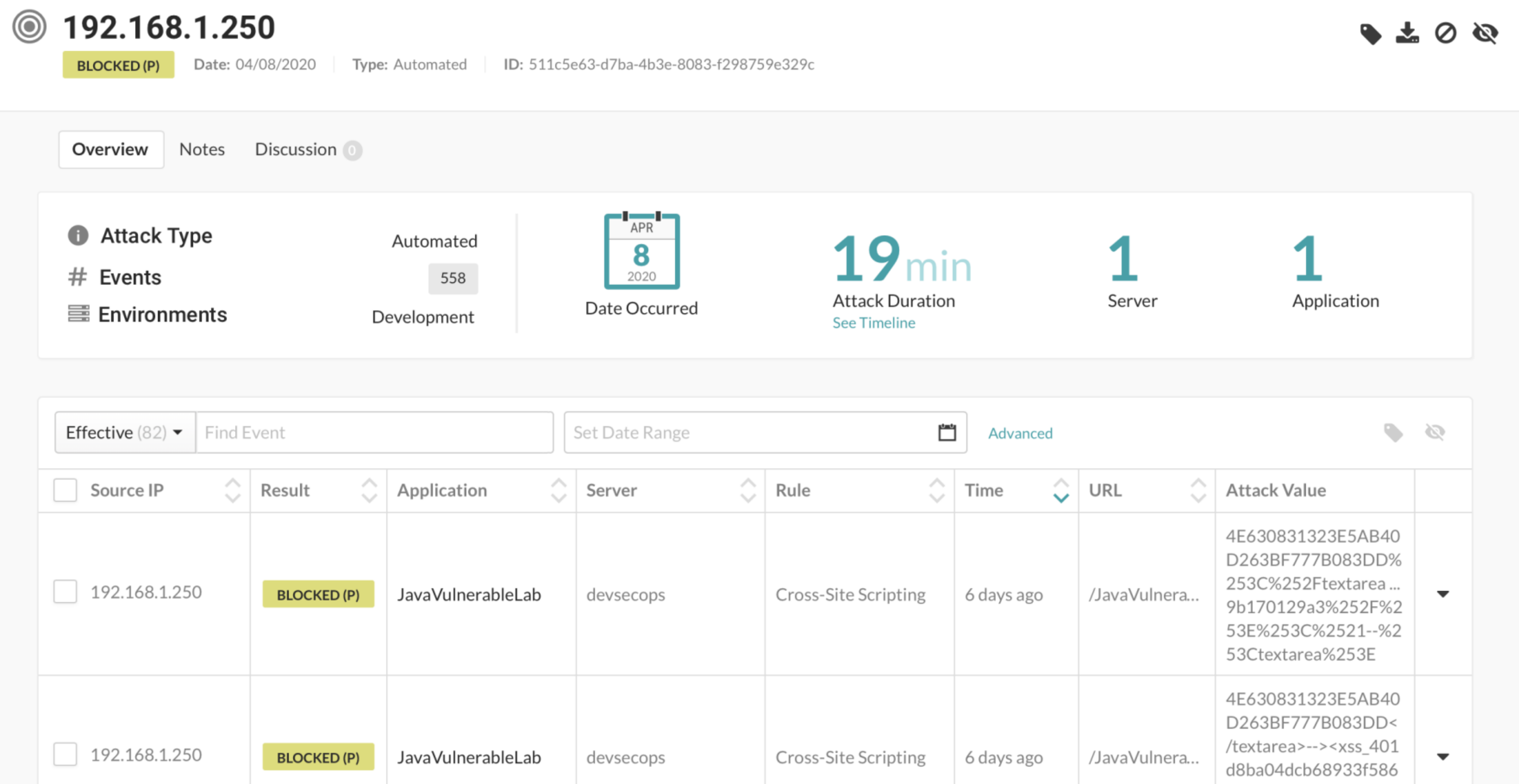
https://www.contrastsecurity.com/contrast-community-edition

[RASP: Contrast Security]



☐ Source IP	Status	Application	Server	Rule	Start	End	Events
☐ 192.168.1.250	BLOCKED (P)	JavaVulnerableLab	devsecops	Command Injection Cross-Site Scripting Path Traversal SQL Injection	6 days ago	6 days ago	558

[RASP: Contrast Security]



192.168.1.250

BLOCKED (P) | **Date:** 04/08/2020 | **Type:** Automated | **ID:** 511c5e63-d7ba-4b3e-8083-f298759e329c

Attack Type	Automated
# Events	558
Environments	Development

APR 8 2020 Date Occurred

19 min Attack Duration
See Timeline

1 Server

1 Application

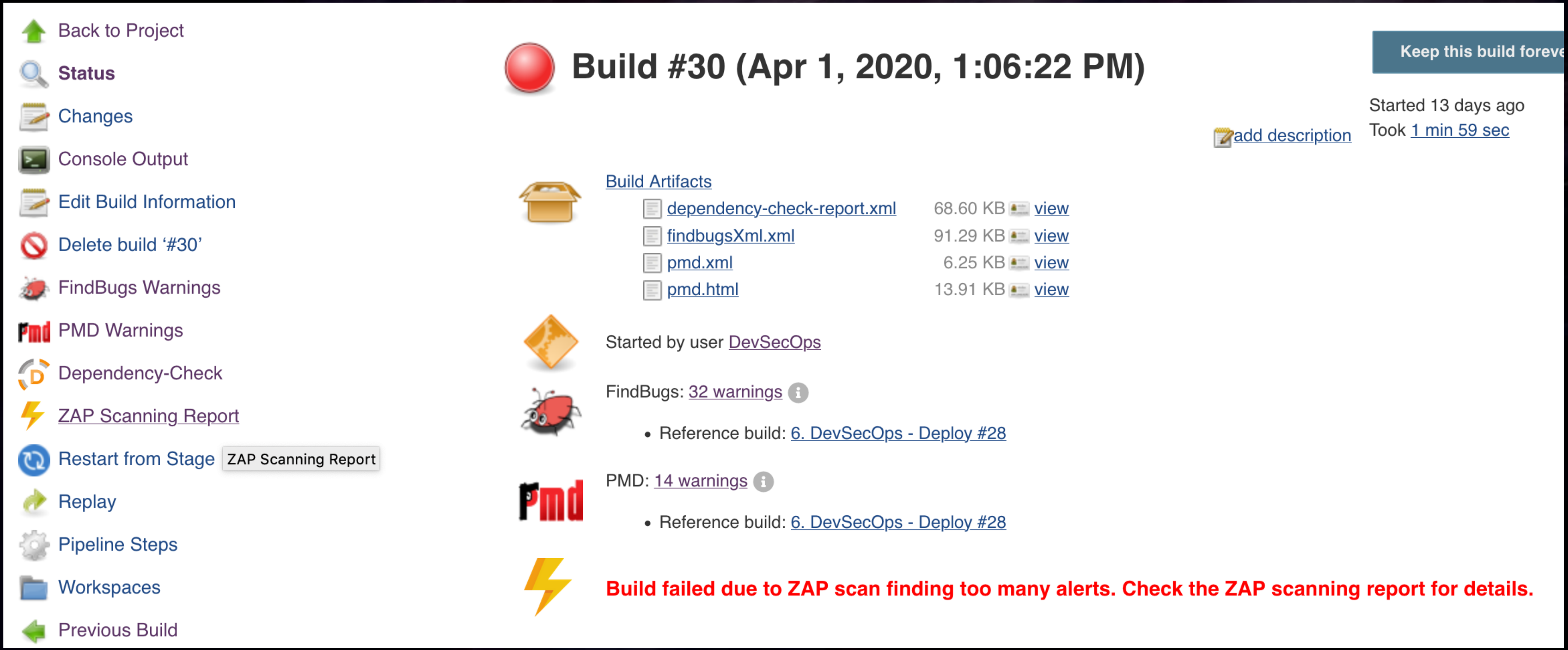
□ Source IP	Result	Application	Server	Rule	Time	URL	Attack Value	
□ 192.168.1.250	BLOCKED (P)	JavaVulnerableLab	devsecops	Cross-Site Scripting	6 days ago	/JavaVulnera...	4E630831323E5AB40D263BF777B083DD%253C%252Ftextarea ... 9b170129a3%252F%253E%253C%2521--%253Ctextarea%253E	▾
□ 192.168.1.250	BLOCKED (P)	JavaVulnerableLab	devsecops	Cross-Site Scripting	6 days ago	/JavaVulnera...	4E630831323E5AB40D263BF777B083DD</textarea>--><xss_401d8ba04dcb68933f586	▾

[Question for You]

- Should you rely on a RASP as a primary defense?
    - No, part of a suite of solutions
    - WAF
    - Vulnerability scanning
    - Robust DevSecOps

Photo by Artem Maltsev on Unsplash

[Jenkins: All Stages]



Build #30 (Apr 1, 2020, 1:06:22 PM)

Started 13 days ago
Took 1 min 59 sec

Build Artifacts

dependency-check-report.xml	68.60 KB	view
findbugsXml.xml	91.29 KB	view
pmd.xml	6.25 KB	view
pmd.html	13.91 KB	view

Started by user DevSecOps

FindBugs: 32 warnings

- Reference build: 6. DevSecOps - Deploy #28

PMD: 14 warnings

- Reference build: 6. DevSecOps - Deploy #28

Build failed due to ZAP scan finding too many alerts. Check the ZAP scanning report for details.

[Container Security Monitoring]

- Falco
  - CNCF Incubating Project
- Kubernetes threat detection engine
- Rules to detect malicious or unexpected activity
  - Exploit unpatched/new vulnerabilities
  - Insecure configurations
  - Leaded or weak credentials
  - Insider threats
- Linux system call monitoring
- Pairs with Kubernetes application context and API

https://falco.org

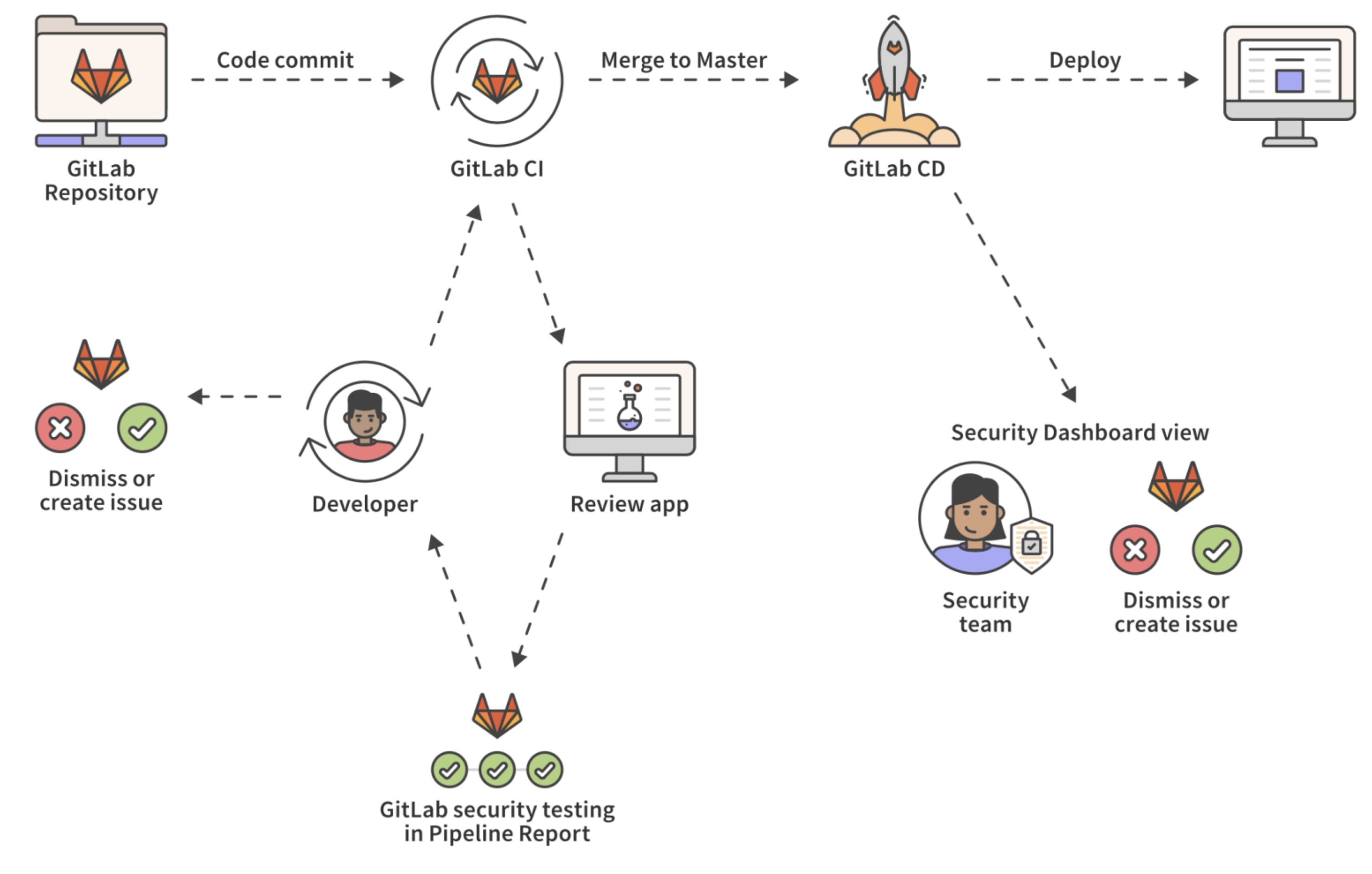
Thank you!

QUESTIONS ?

https://www.linkedin.com/in/philipkulp

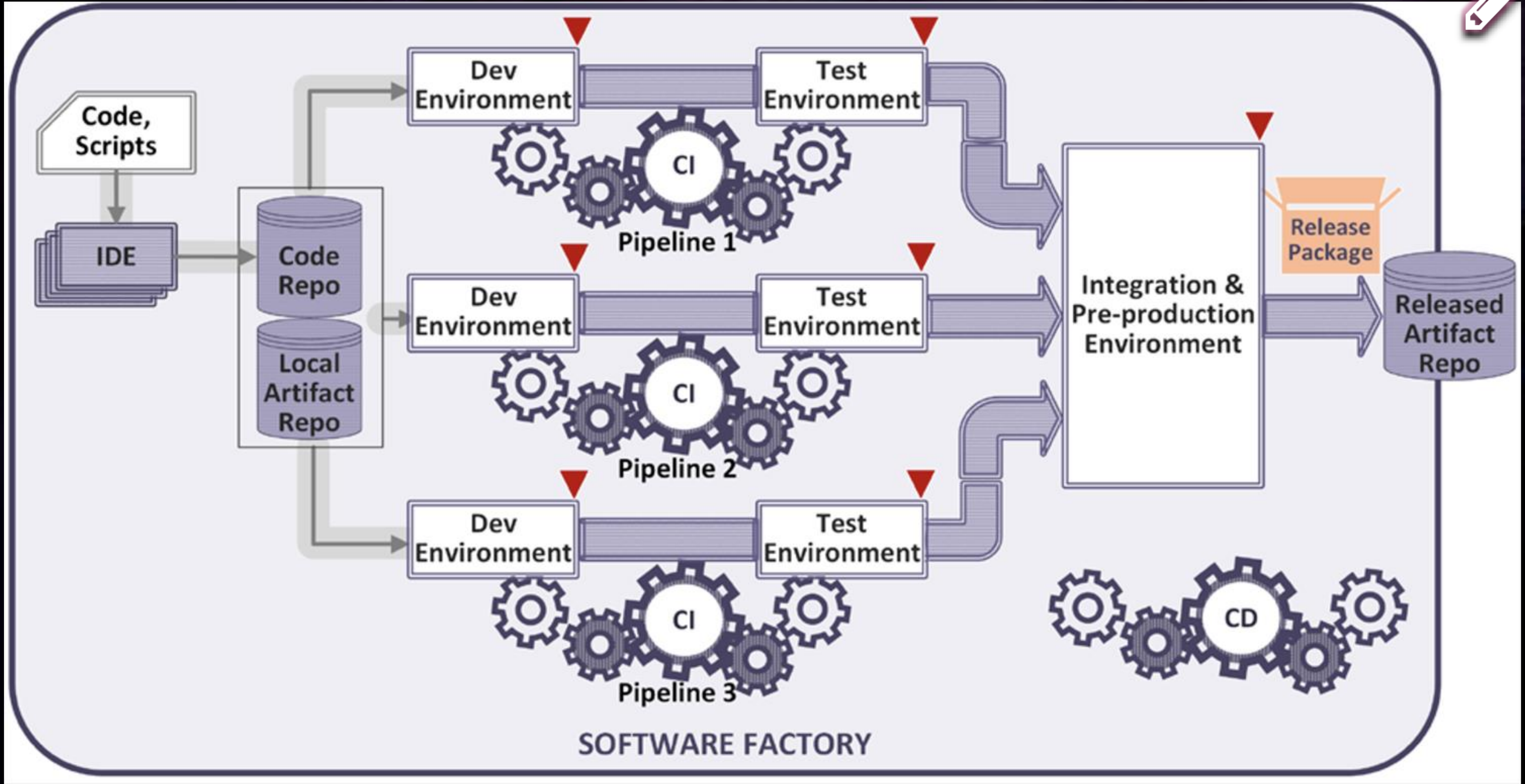
[BACKUP SLIDES]

Gitlab Secure CI/CD



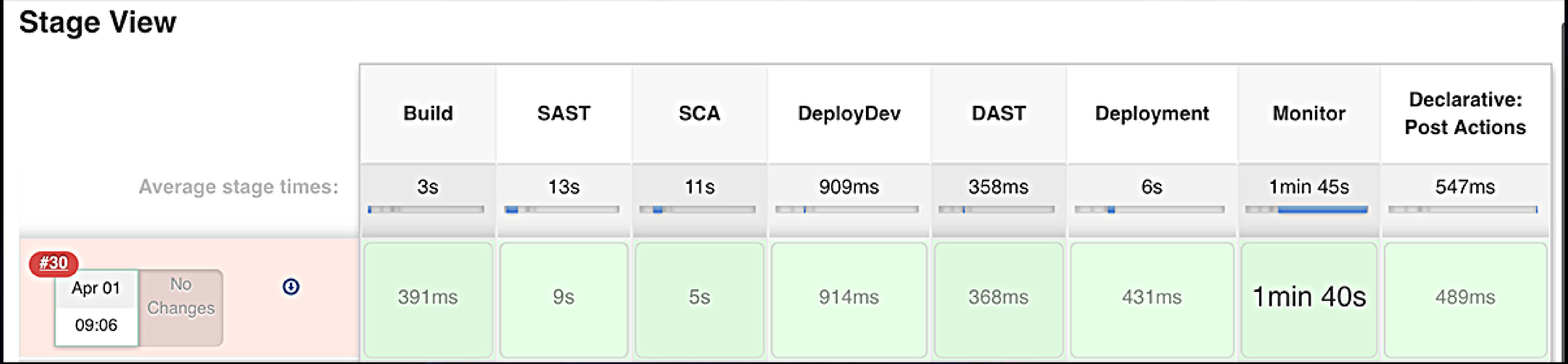
https://about.gitlab.com/solutions/dev-sec-ops/

Continuous Integration



DoD Enterprise DevSecOps Reference Design

[Jenkins Pipeline]



Stage View

	Build	SAST	SCA	DeployDev	DAST	Deployment	Monitor	Declarative: Post Actions
Average stage times:	3s	13s	11s	909ms	358ms	6s	1min 45s	547ms
#30 Apr 01 09:06 No Changes	391ms	9s	5s	914ms	368ms	431ms	1min 40s	489ms

[Coding Languages]

- So many languages
  - Java, .Net, Python, PHP, Ruby
  - JSP, JavaScript, HTML5, CSS, and more
- Developer’s security knowledge?
- Vulnerabilities depend on implementation
  - Infrastructure
  - Security depends on database

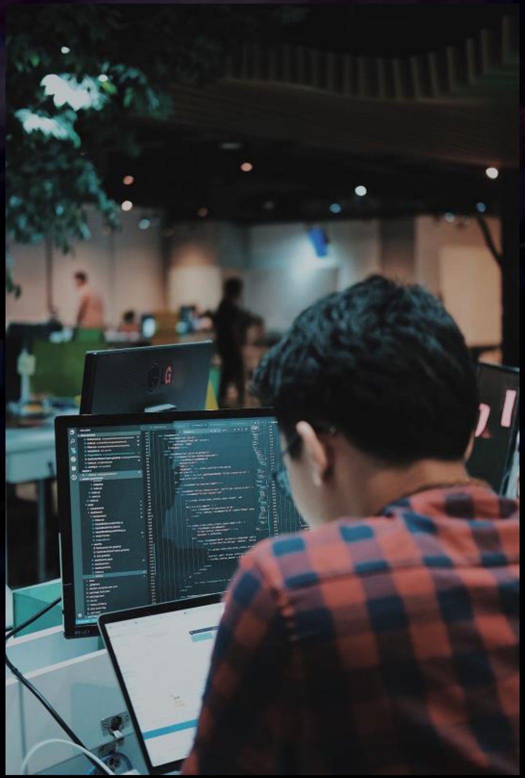


Photo by AArif Riyanto on Unsplash

[3rd Party / Supply Chain] (PW.3) ✎

- **Exploit Vector**
  - Python, Node, PHP, Java, JavaScript, Go
  - Libs and deps used without review [1]
  - 20 libs in NPM nexus to half ecosystem [2]
  - 11 lines of code broke the Internet [3]

- **Objective**
  - Equifax – Apache Struts
  - ASUS breach

Mitigation

- Review 3rd party libs regularly
- Don't *"pip install -r requirements.txt"*
- Open source project reviews

[1] https://medium.com/@bertusk/cryptocurrency-clipboard-hijacker-discovered-in-pypi-repository-b66b8a534a8
[2] https://www.zdnet.com/article/hacking-20-high-profile-dev-accounts-could-compromise-half-of-the-npm-ecosystem/
[3] https://qz.com/646467/how-one-programmer-broke-the-internet-by-deleting-a-tiny-piece-of-code/

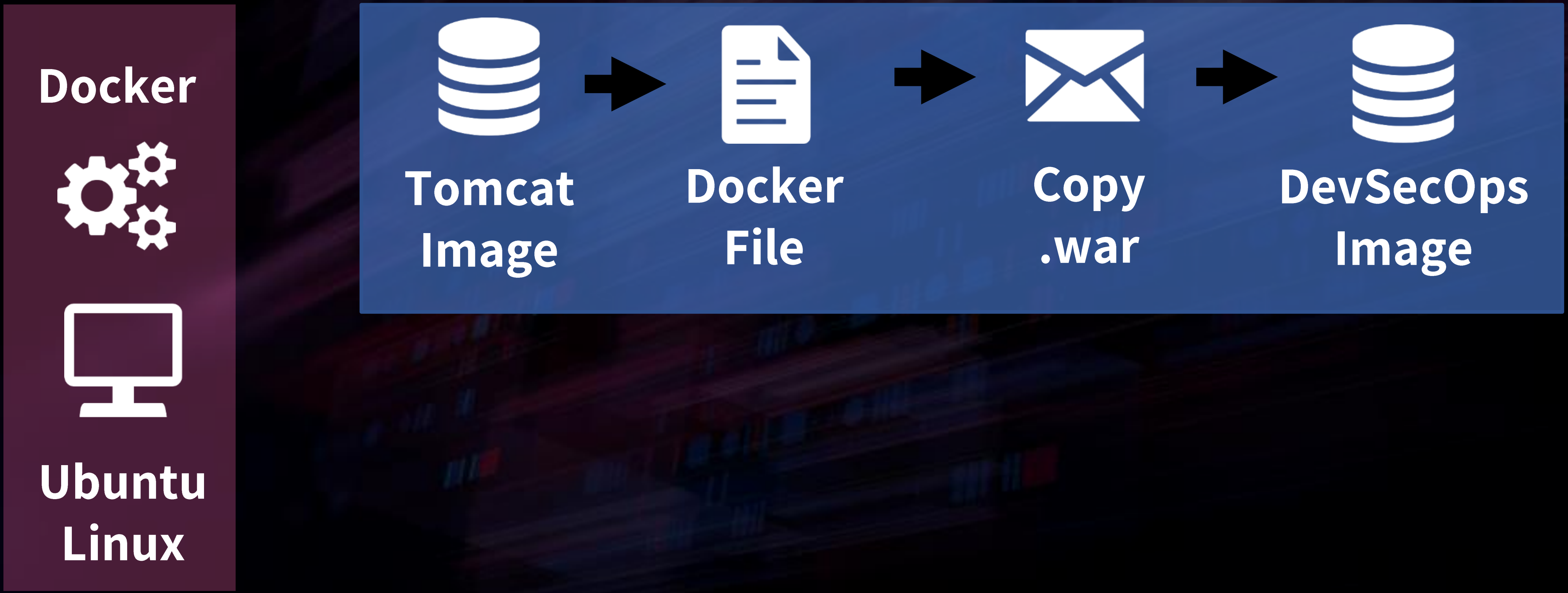
[Delivery]

- Transition from Build to Deployment
  - Tests readiness to release
- Package app for deployment
  - Provide a mechanism for verifying software release integrity (PS.2) ✎
  - Archive and protect each software release (PS.3) ✎
- Security testing performed
- Other testing
- Go / no-go decision

[IaC] *(cont'd)*

- Idempotence
  - Action always produces the same result
- Immutable
  - Architecture cannot be changed in production
- Prod, dev, and test the same
  - Avoid deployment failures from environment drift
- No changes, re-deploy new version
- Deliver
  - Rapidly
  - Reliably
  - At scale

[Docker Custom Image]

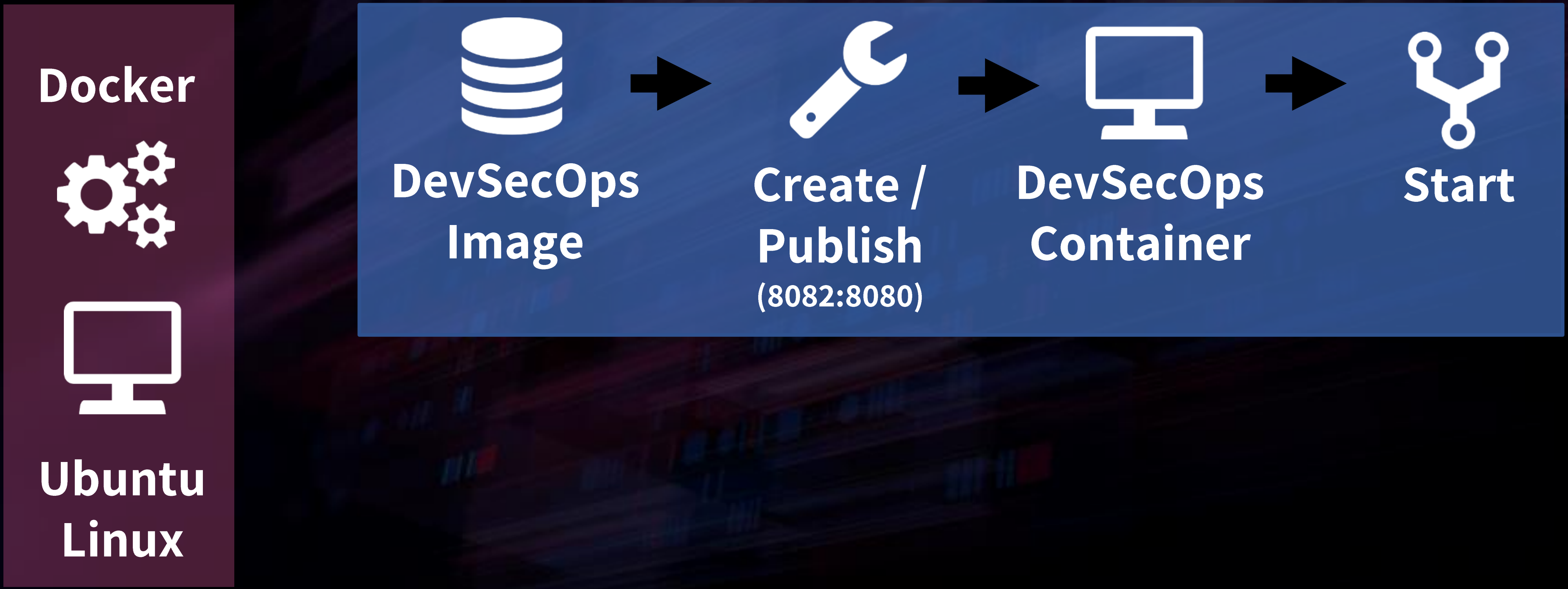


Docker

Ubuntu Linux

Tomcat Image → Docker File → Copy .war → DevSecOps Image

[Docker Custom Container]



Docker

Ubuntu Linux

DevSecOps Image → Create / Publish (8082:8080) → DevSecOps Container → Start

[Monitoring Tools]

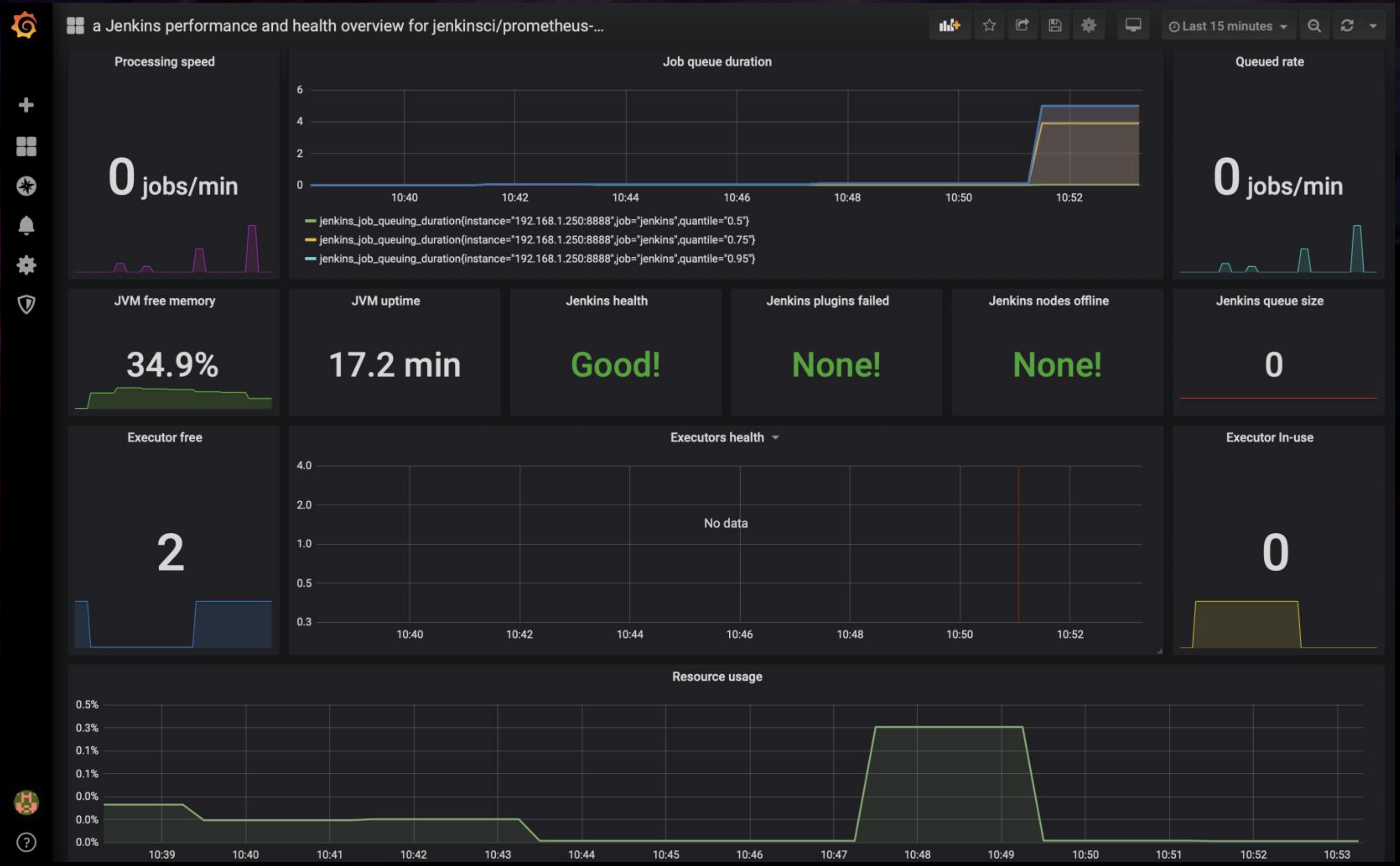
- WAF
- CloudTrail/CloudWatch
- Centralized logging
- SIEM
- Incident tracking
- DLP
- Non-security
  - Nagios
  - CloudWatch alerts

[Monitoring: Metrics & Alerting]

- Prometheus
  - CNCF Graduated Project
- Dimensional data model stored as time series
- PromQL query language
- Data visualization with Grana
- Alerting rules
- Integration with 3rd party exporters
  - Jenkins, Jira
  - MySQL, CouchDB, MongoDB
  - Apache, Nginx, Squid
  - CloudWatch, Nagios, Azure Monitor

https://prometheus.io

[Prometheus: Monitor Jenkins]



a Jenkins performance and health overview for jenkinsci/prometheus-...

Last 15 minutes

Processing speed: 0 jobs/min

Job queue duration

- jenkins_job_queuing_duration{instance="192.168.1.250:8888",job="jenkins",quantile="0.5"}
- jenkins_job_queuing_duration{instance="192.168.1.250:8888",job="jenkins",quantile="0.75"}
- jenkins_job_queuing_duration{instance="192.168.1.250:8888",job="jenkins",quantile="0.95"}

Queued rate: 0 jobs/min

JVM free memory: 34.9%

JVM uptime: 17.2 min

Jenkins health: Good!

Jenkins plugins failed: None!

Jenkins nodes offline: None!

Jenkins queue size: 0

Executor free: 2

Executors health: No data

Executor In-use: 0

Resource usage