



Vulnerability Scanning for App Dependencies

Key Takeaways

All rights reserved to nnSoftware GmbH

No part of this publication may be reproduced, copied, transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of nnSoftware GmbH

About TechWorld with Nana

TechWorld with Nana is an established name in the DevOps and Cloud industry, and it stands for the quality trainings helping 1,000s of engineers acquire the most in-demand skills in this field.



Our mission is enable individual engineers as well as companies to take advantage of the recent developments in Cloud and DevOps fields, to use technologies and concepts in order to create efficient, automated, streamlined DevSecOps processes in organisations.



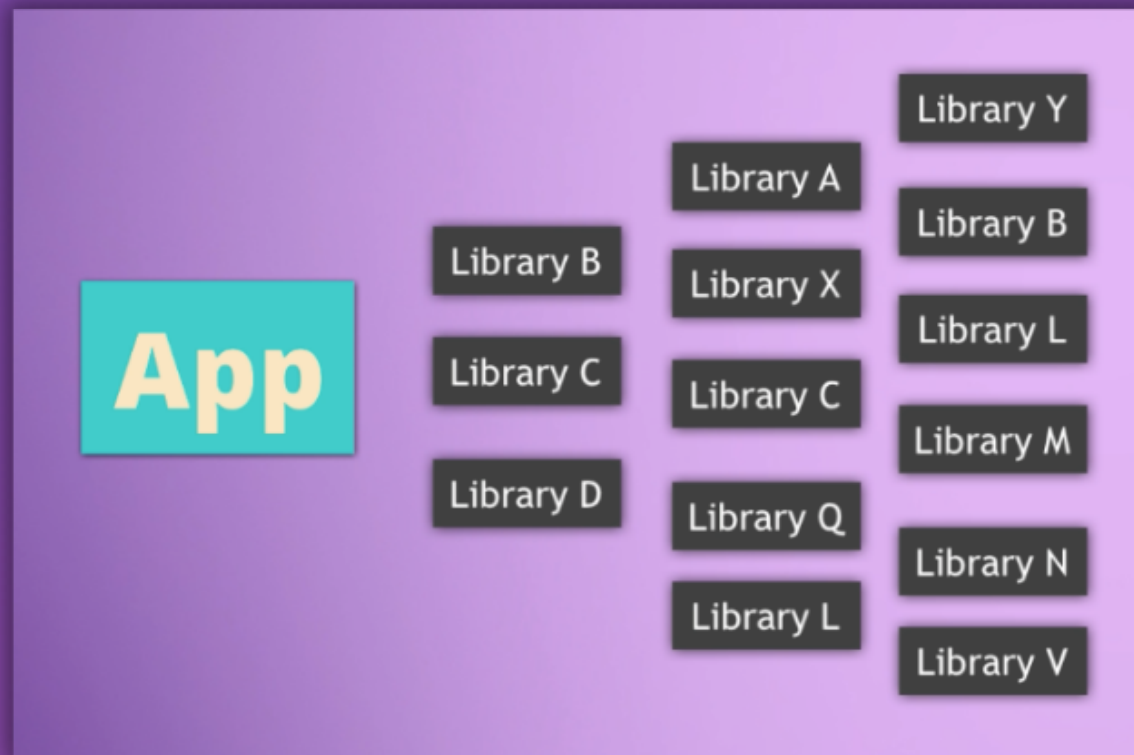
Why Software Composition Analysis (SCA)?

Introduction

Application code is made up of 3rd party libraries



- Developers use many 3rd-party frameworks and libraries to write the code
- So main parts of the final application code are actually external 3rd party open source components



These are called “dependencies” of an application

- Dependencies are external libraries, frameworks and modules that the application relies on to function properly
- Each library used, can depend on other libraries

Introduction

- No matter the programming language, the application always has dependencies
- Dependencies of an application are defined in a dependencies file

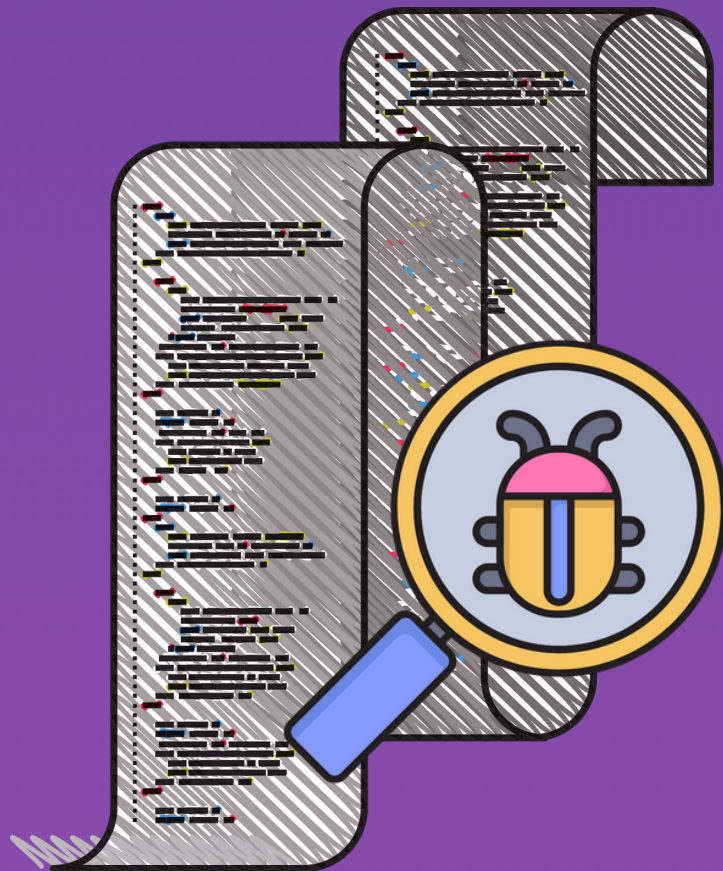
```
    "dependencies": {  
      "body-parser": "^1.19.0",  
      "check-dependencies": "^1.1.0",  
      "clarinet": "^0.12.4",  
      "colors": "1.4.0",  
      "compression": "^1.7.4",  
      "concurrently": "^5.3.0",  
      "config": "^3.3.7",  
      "cookie-parser": "^1.4.5",  
      "cors": "^2.8.5",  
      "dottie": "^2.0.2",  
      "download": "^8.0.0",  
      "errorhandler": "^1.5.1",  
      "exif": "^0.6.0",  
      "express": "^4.17.1",  
      "express-ipfilter": "^1.2.0",  
      "express-jwt": "0.1.3",  
      "express-rate-limit": "^5.3.0",  
      "express-robots-txt": "^0.4.0",  
      "express-security.txt": "^2.0.0",  
      "feature-policy": "^0.5.0",  
      "file-stream-rotator": "^0.5.7",  
      "file-type": "^16.1.0",  
      "filesniffer": "^1.0.3",  
      "finale-rest": "^1.1.1",  
      "fs-extra": "^9.0.1",  
      "fuzzball": "^1.3.0",  
      "glob": "^7.1.6",  
      "graceful-fs": "^4.2.6",  
      "grunt": "^1.2.1",  
      "grunt-contrib-concat": "^1.1.0"
```

In JavaScript apps, dependencies are defined in "package.json" file

Why SCA?



For Hackers it **doesn't matter where the code that ends up in your final application coes from**



To ensure we use secure third party code, we do vulnerability scans for those libraries as well

What is SCA?

Software Composition Analysis (SCA)

- Scans for **vulnerabilities in your dependencies**
- It does also static code analysis



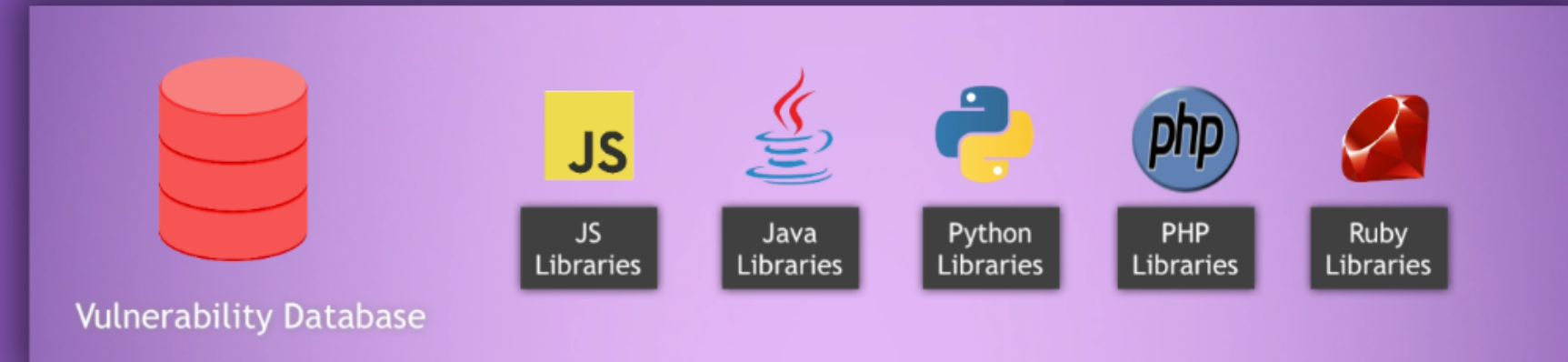
SCA Tools

- Various tools available **per language** (npm audit, retire.js for JavaScript)
- Tools available that **support multiple programming languages** (dependency-check by OWASP)
- Always use popular, verified tools and combination of multiple complimentary tools

Vulnerability Databases

Public vulnerability databases

- Often libraries are used by thousands of other companies/projects
- So generally libraries are scanned continuously for vulnerabilities
- This data is saved in a public vulnerability database



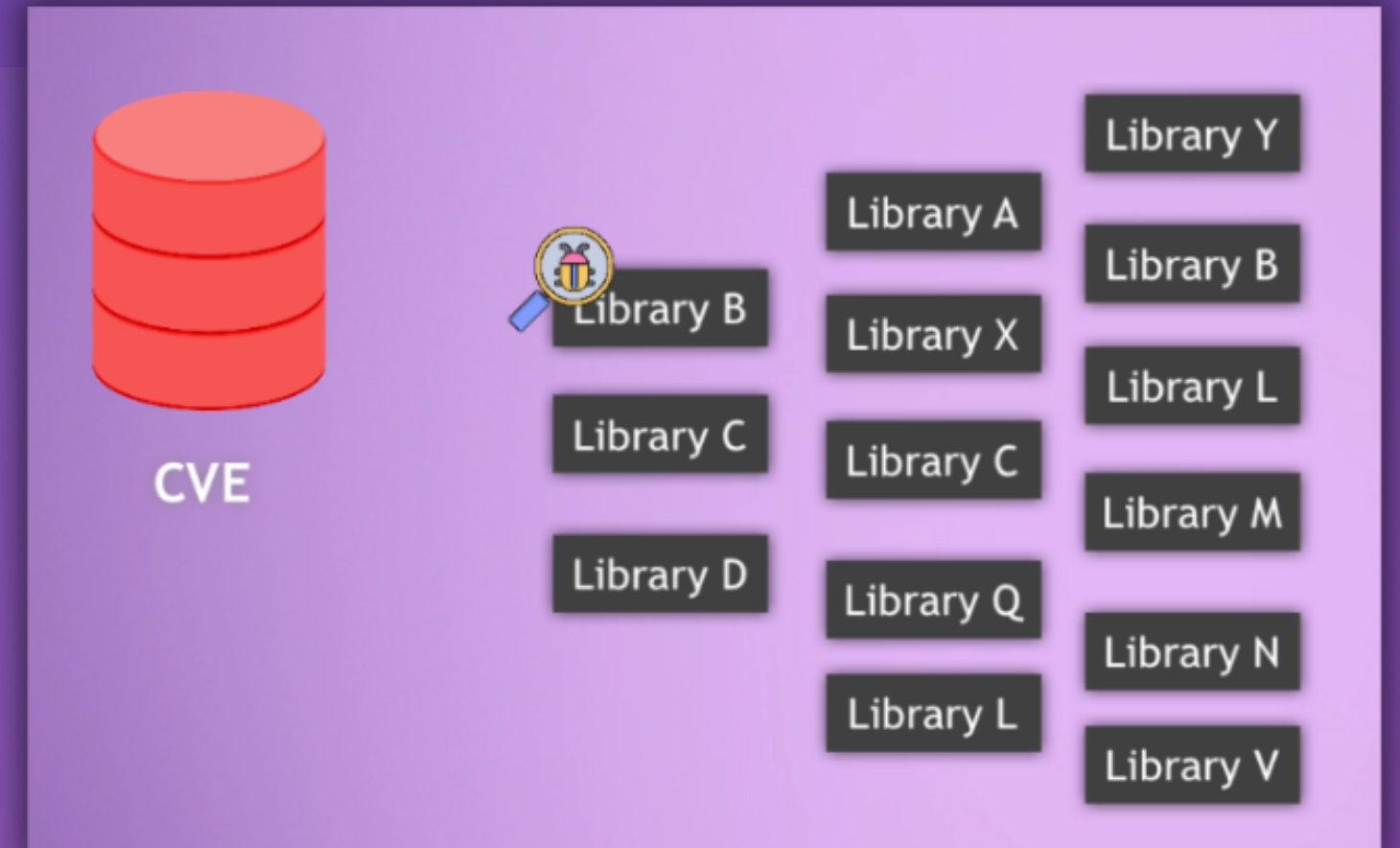
Each library contains information, such as:

- What issues are in which version of the library?
 - In which version is the issue fixed?
- (This data gets updated regularly)

Vulnerability Databases

Common Vulnerabilities and Exposures

- Free service that identifies and catalogs known software vulnerabilities
- CVE is not, in itself, an actionable vulnerability database
- List is maintained by a large community of trusted entities and individuals
- It's used by many security tools



- SCA tool validates whether we use a library with a vulnerable version



Use SCA Tool



SCA Tool

- For any tool you use, you can proceed like this:



1 - Use official Docker Image of Tool

2 - Execute command

3 - Check documentation, what parameters and behavior you need

- DevOps and tools are dynamic and evolving



Retire.js

What it is and how it works

- Popular open source scanner for JavaScript libraries
- Maintains database of known vulnerabilities
- Scans code of libraries in node_modules folder

Add to CI Pipeline

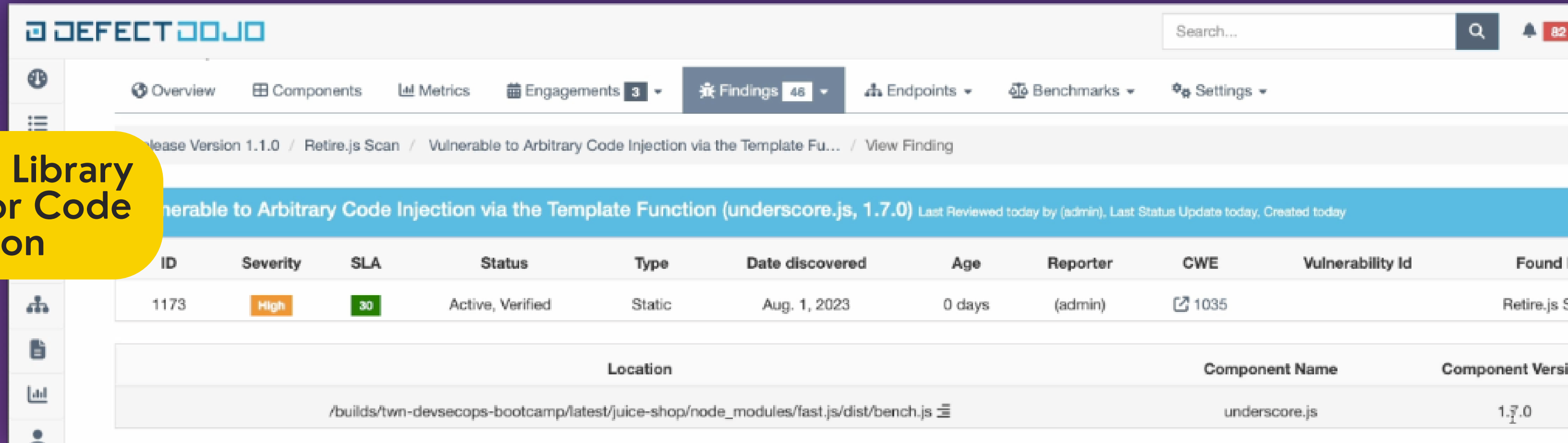
```
80  retire:
81    stage: test
82    image: node:18-bullseye
83    cache:
84      key:
85        files:
86          - yarn.lock
87      paths:
88        - node_modules/
89        - yarn.lock
90        - .yarn
91    policy: pull
92    before_script:
93      - npm install -g retire
94    script:
95      - retire --path . --outputformat json --outputpath retire.json
96    artifacts:
97      when: always
98      paths:
99        - retire.json
```



Remediation Example of Vulnerable Dependencies

Example detected vulnerability

Vulnerable Library
allowing for Code
Injection



The screenshot shows the DefectDojo interface. At the top, there's a search bar and navigation tabs: Overview, Components, Metrics, Engagements (3), Findings (48), Endpoints, Benchmarks, and Settings. Below the tabs, a breadcrumb trail reads: Release Version 1.1.0 / Retire.js Scan / Vulnerable to Arbitrary Code Injection via the Template Fu... / View Finding. The main heading is 'Vulnerable to Arbitrary Code Injection via the Template Function (underscore.js, 1.7.0)'. Below this is a table with columns: ID, Severity, SLA, Status, Type, Date discovered, Age, Reporter, CWE, Vulnerability Id, and Found. The first row shows ID 1173, Severity High, SLA 30, Status Active, Verified, Type Static, Date discovered Aug. 1, 2023, Age 0 days, Reporter (admin), CWE 1035, and Found Retire.js S. Below the table is a section with columns: Location, Component Name, and Component Version. The first row shows Location /builds/twn-devsecops-bootcamp/latest/juice-shop/node_modules/fast.js/dist/bench.js, Component Name underscore.js, and Component Version 1.7.0.

ID	Severity	SLA	Status	Type	Date discovered	Age	Reporter	CWE	Vulnerability Id	Found
1173	High	30	Active, Verified	Static	Aug. 1, 2023	0 days	(admin)	1035		Retire.js S

Location	Component Name	Component Version
/builds/twn-devsecops-bootcamp/latest/juice-shop/node_modules/fast.js/dist/bench.js	underscore.js	1.7.0

🔒 CVE-2021-23358 Detail

Description


The package underscore from 1.13.0-0 and before 1.13.0-2, from 1.3.2 and before 1.12.1 are vulnerable to Arbitrary Code Injection via the template function, particularly when a variable property is passed as an argument as it is not sanitized.

Which is fixed above in a specific version

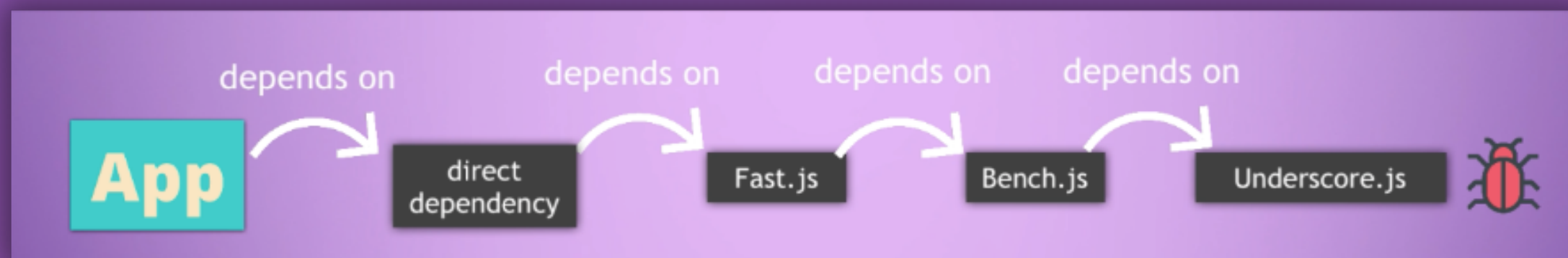
Direct vs Transitive Dependencies



Looks like an easy fix of just updating library version, but it's not...

Location	Component Name
/builds/twn-devsecops-bootcamp/latest/juice-shop/node_modules/fast.js/dist/bench.js 	underscore.js

- **Direct dependency** is package you include in your own project
- **Transitive (indirect) dependency** is a package used by one of your direct dependencies
- It's like a nested tree



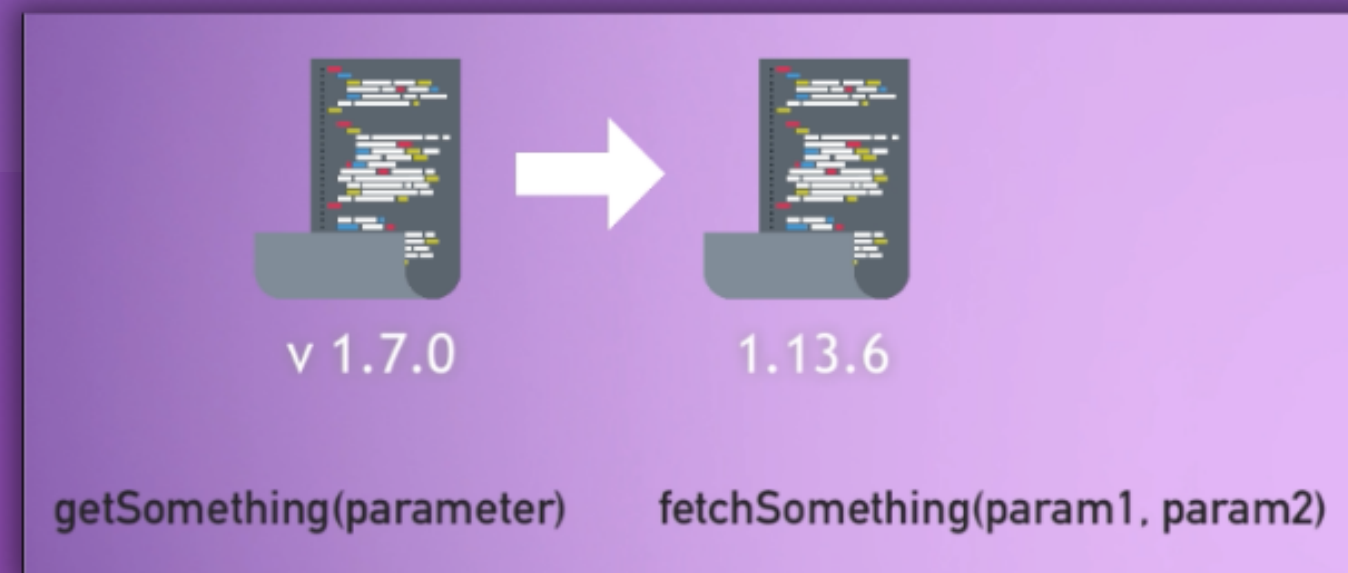
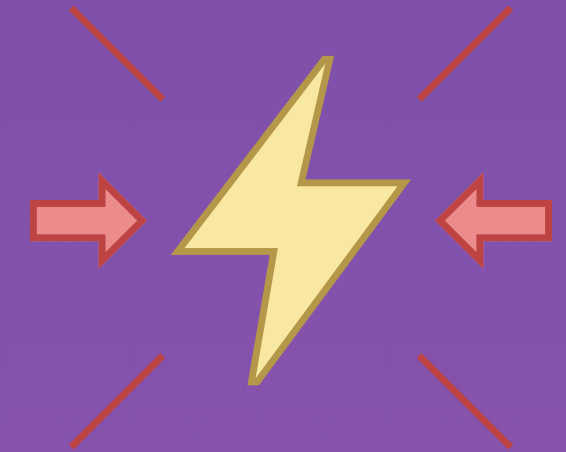
Updating Library Versions in general



Looks like an easy fix of just updating library version, but it's not...

Incompatibility issues

- New library version may be **incompatible** with our application code
- In that case developers may need to **update the actual code**, not just update the version



Note on Updating Library Versions

- This is another reason why developers are responsible for fixing security issues related to the application code itself
- As DevSecOps engineers we make developers aware of the known vulnerabilities
- Developers know the code the best



Especially major updates are critical and must be done with care.
Extensive testing necessary to ensure nothing is broken



Status of DevSecOps Pipeline

Note on Updating Library Versions

Tests different aspects of our application

- ✓ Hard-coded secrets
- ✓ Code security
- ✓ Vulnerabilities in dependencies

