Vulnerable Components



Laurie Williams Laurie_williams@ncsu.edu

Yingyaipumi / stock.adobe.com

Agenda

Overview

- Three supply chain-specific attacks
 - Typosquatting
 - Dependency confusion
 - New: manifest confusion
- Making good component choices
- Identifying vulnerabilities in components

Exercise

Taxonomy of malicious commit attack vector

Exercise

Oops! Accidental dependency vulnerability

#

Q

CBS NEWS

NEWS ${\scriptstyle \vee}$ Shows ${\scriptstyle \vee}$. In Live ${\scriptstyle \vee}$. Local ${\scriptstyle \vee}$

Login

Nightmare before Christmas: What to know about the Log4j vulnerability

BY NICOLE SGANGA UPDATED ON: DECEMBER 17, 2021 / 12:44 PM / CBS NEWS

f 🎔 🖬





Seventyfour i/stock.adobe.com Yingyaipumi/stock.adobe.com

Code dependencies as an attack vector Code dependencies as a weapon

FORTUNE RANKINGS ~

MAGAZINE NEWSLETTERS PODCASTS MORE ~

SEARCH SIGN IN



Russia's ruble has almost totally recovered. Does that mean sanctions aren't working?





Study finds ivermectin, the horse drug Joe Rogan championed as a COVID treatment, does nothing to cure the virus



accumulated as much wealth as Mark Zuckerberg in a quarter the time, explains how it feels to become unfathomably rich virtually overnight

Subscribe Now



INTERNATIONAL • UKRAINE INVASION

Russia's largest bank tells its clients to delay downloading software updates after 'protestware' attacks target Russian users

BY NICHOLAS GORDON March 22, 2022 7:07 AM EDT

node-ipc 🛛 🗖

11.1.0 • Public • Published 24 days ago



Siridhata/stock.adobe.com <u>Sergev Nivens</u>/stock.adobe.co





A6: Vulnerable and Outdated Components

- Components used in an application are outdated or have a vulnerability
- At the root of software supply chain attacks (think: Executive Order 14028).
- Notable CWEs:
 - 1104: Use of unmaintained thirdparty components
 - 1035: Using components with known vulnerabilities





Mindset shift required

"Some might argue that it's almost too easy to introduce a new dependency into your software systems. I'm definitely guilty of this in my previous life as an engineer. I remember pulling in random Python packages when building my own websites and not putting any thought into security. It should be fine if so many other people are using the same package, right?"

-- Kim Lewandowski, [Google Product Manager, founder Chainguard] and every other developer alive



https://openssf.org/blog/ #3 Choosing dependencies



* Snyk: State of Open Source Dependencies 2020



https://www.explainxkcd.com/wiki/index.php/2347:_Dependency

Ponder this ...

When you bring a third-party component into your project, it's like you are adding the developers of that component to your team. Do you trust them?

How about the development teams for all the transitive dependencies?

Kind of mind blowing



Agenda

Overview

- Three supply chain-specific attacks
 - Typosquatting
 - Dependency confusion
 - New: manifest confusion
- Making good component choices
- Identifying vulnerabilities in components

Exercise

Taxonomy of malicious commit attack vector

Exercise



Dependency Confusion

Most common attack 2021 49% of organizations are vulnerable

- Early build step download source and dependencies from approved source and artifact repos
 - Anyone can freely upload code
- Install dependencies: Node has npm; Python's pip uses PyPi; RubyGems pip install package_name
- Typosquatting leverages typo'd versions of popular package names
- Dependency confusion: a software installer is tricked into a pulling a malicious code file from a public repository instead of the intended file from an internal repository

Dependency Confusion - 2



- Public package contains higher version compared to private package
- If package indexing not done properly, it will automatically pull the higher version from the public registry

Dependency confusion - 3

- Finding private/internal packages (NPM)
 - Look at the package.json file

운 45ce538c8d - flipper / desktop / eslint-plugin-flipper / package.json
dependabot Bump @typescript-eslint/parser from 4.28.5 to 4.29.1 in /desktop (#2700)
A 4 contributors 🧌 🔁 👰 🍪
41 lines (41 sloc) 1017 Bytes
1 {
<pre>2 "name": "eslint-plugin-flipper",</pre>
3 "version": "0.0.0",
4 "private": true,
5 "description": "Custom ESLint rules for Flipper",
6 "repository": "facebook/flipper",
7 "main": "lib/index.js",
<pre>8 "flipperBundlerEntry": "src",</pre>
<pre>9 "types": "lib/index.d.ts",</pre>
10 "license": "MIT",
<pre>11 "bugs": "https://github.com/facebook/flipper/issues",</pre>
12 "dependencies": {
<pre>13 "@typescript-eslint/experimental-utils": "^4.28.5",</pre>
14 "fs-extra": "^10.0.0"
15 },
16 "devDependencies": {
eduum com/walex birsan/dependency-confusion-4a5d6(tec61()

Pinning dependencies

- Specify an exact version, under version control
- Example:
 - Npm lockfiles that list fix versions for all dependencies (direct and transitive)

🎁 snyk Advisor

pin-dependencies-checker

V1.0.6

	POPULARITY	SMALL
Package Health Score	MAINTENANCE	INACTIVE
48/100	SECURITY	NO KNOWN SECURITY ISSUES
904.403	COMMUNITY	LIMITED

Manifest confusion (npm)

occurs when there is an inconsistency between a package's manifest information presented on the npm registry and the actual 'package.json' file in the tarball of the published npm package used when the package is installed.

Nasal Piercings Manipulator	Pro Teams Pricing Documental
Q Search packages	Search Sign Up Sign
rcyclarke-manifest-pkg .15 • Public • Published an hour ago	
Readme Code Beta © 0 Dependencies	🗞 0 Dependents 💿 16 Versions
darcyclarke-manifest-p.kg / package.json	Install
Back 12 LOC	> npm i darcyclarke-manifest-pkg
	Version License
<pre>"name": "express", "version": "3.0.0",</pre>	2.1.15 none
<pre>"name": "express", "version": "3.0.0", "main": "index.js", "scripts": {</pre>	Unpacked Size Total Files
<pre>"name": "express", "version": "3.0.0", "main": "index.js", "scripts": { "install": "touch ./bad-pkg-write && echo \"bad pkg exec!\"\n" },</pre>	2.1.15noneUnpacked SizeTotal Files248 B2
<pre>{ "name": "express", "version": "3.0.0", "main": "index.js", "scripts": { "install": "touch ./bad-pkg-write && echo \"bad pkg exec!\"\n" }, "license": "ISC", "dependencies": { </pre>	2.1.15 none Unpacked Size Total Files 248 B 2 Last publish
<pre>{ "name": "express", "version": "3.0.0", "main": "index.js", "scripts": { "install": "touch ./bad-pkg-write && echo \"bad pkg exec!\"\n" }, "license": "ISC", "dependencies": { "sleepover": "*" } </pre>	2.1.15 none Unpacked Size Total Files 248 B 2 Last publish an hour ago

Risks of Manifest Confusion

- installation of unknown dependencies that won't show upon security tools
- execution of unknown scripts, and
- potentially also downgrade attacks

Needed action:

- Developers should manually read the package.json to determine version numbers, what dependencies will be installed, and what scripts will be executed
- Tools emerging

Agenda

Overview

- Three supply chain-specific attacks
 - Typosquatting
 - Dependency confusion
 - New: manifest confusion
- Making good component choices
- Identifying vulnerabilities in components

Exercise

Taxonomy of malicious commit attack vector

Exercise

Potential weak links

... may increase risk of supply chain attack



People are often the weak link

Kirsty Pargeter

(npm) Weak link signals



Install Scripts

93.9% of malicious packages use install scripts.

Unmaintained Packages

58.6% packages & **44.3%** maintainers are inactive.

Too many Maintainers

421 popular packages have 14,566 maintainers.

Too many Contributors

45 maintainers supervise **2,780** contributors in **23** popular packages. Overloaded Maintainers

4,743 maintainers own 52.4% packages in npm.



Nusrat@Ma	acBook-Pro-6 ~ %	scorecard	repo=github	.com/mochajs/mocha	checks	Dangerous-Workflow	show-deta
Starting	[Dangerous-Workf	low]					
Finished	[Dangerous-Workf	low]					

RESULTS

Aggregate score: 0.0 / 10

Check scores:

SCORE	NAME	REASON	DETAILS	
	Dangerous-Workflow	dangerous workflow patterns detected	<pre>Warn: untrusted code checkout '\${{ github.event.pull_request.head.sha })': .github/workflows/browser-test.yml:18 Warn: secret accessible to pull requests '\${{secrets.SAUCE_USERNAME}}': .github/workflows/browser-test.yml:33 Warn: secret accessible to pull requests '\${{secrets.SAUCE_ACCESS_KEY}}': .github/workflows/browser-test.yml:34 Warn: secret accessible to pull requests '\${secrets.GITHUB_TOKEN}': .github/workflows/browser-test.yml:34 Warn: secret accessible to pull requests '\${secrets.GITHUB_TOKEN}': .github/workflows/browser-test.yml:34 Warn: secret accessible to pull requests '\${secrets.SAUCE_ACCESS_KEY}}: .github/workflows/mocha.yml:160 Warn: secret accessible to pull requests '\${{ secrets.SAUCE_ACCESS_KEY}': .github/workflows/mocha.yml:161 Warn: secret accessible to pull requests '\${ secrets.GITHUB_TOKEN}': .github/workflows/mocha.yml:161 Warn: secret accessible to pull requests '\${ secrets.SAUCE_ACCESS_KEY}': .github/workflows/mocha.yml:161 Warn: secret accessible to pull requests '\${ secrets.GITHUB_TOKEN}': .github/workflows/mocha.yml:134 warn: secret accessible to pull requests '\${ secrets.GITHUB_TOKEN}': .github/workflows/mocha.yml:132 } }</pre>	htt

RESULTS

Aggregate score: 5.6 / 10

Check scores:

I	L	1	
SCORE	I NAME	REASON	
10 / 10	Binary-Artifacts	no binaries found in the repo	htt
0 / 10	 Branch-Protection 	branch protection not enabled on development/release branches	htt
10 / 10	CI-Tests 	28 out of 28 merged PRs checked by a CI test score normalized to 10	htt
0 / 10	CII-Best-Practices	no badge detected	htt
9 / 10	 Code-Review 	GitHub code reviews found for 28 commits out of the last 30 score normalized to 9	htt
10 / 10	Contributors	65 different companies found score normalized to 10	htt
0 / 10	 Dangerous-Workflow 	dangerous workflow patterns detected	htt
10 / 10	Dependency-Update-Tool	update tool detected	htt
0 / 10	 Fuzzing	project is not fuzzed	htt
10 / 10	 License	license file detected	htt
10 / 10 	 Maintained 	30 commit(s) out of 30 and 26 issue activity out of 30 found in the last 90 days score normalized to 10	htt
?	Packaging	no published package detected	htt
6 / 10	 Pinned-Dependencies 	dependency not pinned by hash detected score normalized to 6	htt
0 / 10	SAST	SAST tool is not run on all commits score normalized to 0	htt
10 / 10	Security-Policy	security policy file detected	htt
?	Signed-Releases	no releases found	htt
0 / 10	 Token-Permissions 	non read-only tokens detected in GitHub workflows	htt
10 / 10	Vulnerabilities	no vulnerabilities detected	htt

Dep.dev

io.fabric8:kubernetes-model-core • 6.1.1 -

verview Dependencies Dependents	Compare Versions		
Security Advisories			Published September 1, 2022 Description
Licenses Learn more about license information. LICENSES Apache-2.0 DEPENDENCY LICENSES Apache-2.0 EPL-2.0 MIT EPL-1.0 non-standard		5	Java client for Kubernetes and OpenShift Links ORIGIN https://search.maven.org/artifact/io.fabric8/kubernetes-model-core/6.1.1/jar HOMEPAGE http://fabric8.io/ REPO https://github.com/fabric8io/kubernetes-client Projects
Dependencies Direct Indirect	13 26 View all dependencies	39	fabric8io/kubernetes-client GitHub Java client for Kubernetes & OpenShift Y* 1k forks ★ 3k stars
Dependents Direct Indirect	63 118 View dependents	181	OpenSSF scorecard The Open Source Security Foundation is a cross-industry collaboration to improve the security of open source software (OSS). The Scorecard provides security health metrics for open source projects. View information about checks and how to fix failures. score 7/10

Mean Time to Update (MTTU)

Time to Remediate Known OSS Vulnerabilities After Detection



2020 Sonotype State of the Software Supply Chain Security

What other weak links can you think of?

If you want to make a good component choice, what should be consider?

Agenda

Overview

- Three supply chain-specific attacks
 - Typosquatting
 - Dependency confusion
 - New: manifest confusion
- Making good component choices
- Identifying vulnerabilities in components

Exercise

Taxonomy of malicious commit attack vector

Exercise

Software Component Analysis (SCA) Tools

Dependabot testSecurityAlert O Unwatch - 1 ¥ Fork 0 🗙 Star 0 username 💷 Wiki <> Code ① Issues 0 ① Pull requests 4 ① Actions III Projects 0 Security 🔄 🖞 Insights 🔅 Settings We found a vulnerable dependency in a repository you have security alert access to. waitress Dismiss - Alerts A fix has already been started A Open GitHub opened this alert 11 days ago Advisories reponame No bandwidth to fix this Policy Risk is tolerable to this project Known high severity security vulnerability detected in 1 Bump waitress from 1.3.0 to 1.4.3 dependencies This alert is inaccurate or incorrect swagger-ui < 3.23.11 defined in package-lock.json.</pre> #2 opened 11 days ago by dependabot bot Vulnerable code is not actually used package-lock.json update suggested: 5 waitress vulnerabilities found in requirements.txt 11 days ago swagger-ui ~> 3.23.11. Always verify the validity and compatibility of suggestions with your Remediation codebase. Upgrade waitress to version 1.4.2 or later. For example: waitress>=1.4.2 Always verify the validity and compatibility of suggestions with your codebase. **Review vulnerable dependency** Details GHSA-968f-66r5-5v74 high severity Vulnerable versions: < 1.4.2 Patched version: 1.4.2 SRC DNTRAST \\ 🕘 WhiteSource 🗑 snyk DEPENDENCY-CHECK SECURITY and BLACKDUCK more....

Software Component Analysis (SCA)

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]		
Ca	tenory									Total	Distribution
C		Destriction	-60	a a suith in A		da				rotai	
CV	/E-119 Improper	Restriction	or Operatio	ns within t	ne Bound	ds of a r	Memory E	surrer		5	
CV	/E-134 Uncontroll	led Format	String							1	
CV	/E-189 Numeric E	rrors								2	
CV	/E-20 Improper I	nput Valida	tion							7	
CV	/E-200 Informatio	on Exposure	<u>e</u>							5	
CV	/E-22 Improper L	imitation of	f a Pathnam	e to a Res	tricted D	irectory	('Path Tr	raversal	(')	4	
CV	E-264 Permission	ns, Privilege	es, and Acc	ess Contro	ls					4	
CV	/E-287 Improper	Authenticat	tion							2	
CV	/E-310 Cryptogra	phic Issues								2	
CV	/E-399 Resource	Manageme	nt Errors							7	
CV	/E-59 Improper L	ink Resolut	ion Before l	File Access	('Link Fo	ollowing	<u>')</u>			4	
CWE-79 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')						cripting')	14				
CV	E-89 Improper N	Neutralizatio	on of Specia	I Elements	used in	an SQL	Comman	nd ('SQL	Injection')	2	
CV	CWE-94 Improper Control of Generation of Code ('Code Injection')							10			
То	tal									153	

https://www.vojtechruzicka.com/detecting-dependencies-known-vulnerabilities/

Tool	Alert	Unique Dependency	Unique Package	Unique Vulnerability	CVE	Non-CVE	Scan Time (Minutes)
	0.0000-0.000	Total (Median	per project	t)			()
OWASP DC	12,466 (254.0)	332 (38.0)	149 (36.0)	313 (117.0)	289	24	14.4
Snyk	4,902 (66.0)	96 (6.0)	46 (6.0)	189 (23.0)	178	11	15.1
Dependabot	136 (0.0)	20 (0.0)	11 (0.0)	61 (0.0)	61	0	NA
MSV	3,197 (58.0)	36 (12.0)	14 (12.0)	36 (22.0)	36	0	3.4
Steady	2,489 (51.0)	91 (20.0)	39 (19.0)	97 (41.0)	89	8	385.0
WhiteSource	434 (0.0)	76 (0.0)	44 (0.0)	146 (0.0)	127	19	NA
Commercial A	2,998 (70.0)	107 (24.0)	53 (24.0)	208 (70.0)	187	21	NA
Commercial B	205	35	35	127	127	0	NA

Table 2: Vulnerable Dependencies for Maven (Java) projects

Table 3: Vulnerable Dependencies for npm (JavaScript) projects

Tool	Alert	Unique Dependency Path	Unique Dependency	Unique Package	Unique Vulnerability	CVE	Non- CVE	Scan Time (Minutes)	
Total (Median per project)									
OWASP DC	1,379 (208.0)	498 (72.0)	239 (71.0)	160 (57.0)	234 (71.0)	78	156	4.4	
Snyk	2,210 (135.0)	1,004 (44.0)	90 (20.0)	54 (17.0)	121 (26.0)	79	42	1.0	
Dependabot	97 (8.0)	NA	32 (1.0)	30 (1.0)	45 (4.0)	29	16	NA	
npm audit	1,266 (37.0)	852 (28.0)	58 (12.0)	45 (12.0)	62 (16.0)	31	31	0.1	
WhiteSource	205 (32.0)	205 (32.0)	89 (14.0)	55 (9.0)	96 (18.0)	58	38	NA	

Imtiaz, Thorn, Williams, A comparative study of vulnerability reporting by software composition analysis tools, ESEM 2021

Overlap in finding same vulnerable components



(a) Overlap ratios for Maven vulnerable dependencies



npm vulnerable dependencies

(b) Overlap ratios for npm vulnerable dependencies

Imtiaz, Thorn, Williams, A comparative study of vulnerability reporting by software composition analysis tools, ESEM 2021

OWASP Juice Shop

modern and sophisticated insecure web application



https://owasp.org/www-project-juice-shop/

OWASP Dependency Check

- Look at JuiceShop report: <u>https://tinyurl.com/3yev9jt2</u>
- Pick a high severity/high confidence vulnerability. Go to the National Vulnerability Database (NVD) Common Vulnerability Enumeration (CVE) and summarize the vulnerability



Project: WolfpackShop

Scan Information (show all):

- dependency-check version: 8.3.1
- Report Generated On: Sat, 8 Jul 2023 08:08:01 -0400
- Dependencies Scanned: 25057 (19009 unique)
- Vulnerable Dependencies: 42
- Vulnerabilities Found: 90
- Vulnerabilities Suppressed: 0
- ...

Analysis Exceptions

Unable to read yarn audit output.

Summary

Display: Showing Vulnerable Dependencies (click to show all)

Dependency	Vulnerability IDs	Package	Highest Severity	CVE Count	Confidence	Evidence Count
bench.js		pkg;javascript/underscore.js@1.7.0	HIGH	1		3
crypto-js:3.3.0	cpe:2.3;a;crypto-js_project;crypto-js;3.3;0;*;*;*;*;*;*	pkg:npm/crypto-js@3.3.0	MEDIUM	1	Highest	8
dottie:2.0.3	cpe:2.3;a:dottie_project:dottie:2.0.3;*;*;*;*;*;*	pkg:npm/dottie@2.0.3	HIGH	2	Highest	6
ecstatic: 3.3.2	cpe:2.3:a:ecstatic_project:ecstatic:3.3.2:*********	pkg:npm/ecstatic@3.3.2	HIGH	1	Highest	7
engine.io:4.1.2	cpe:2.3;a;socket:engine.io;4.1,2;********	pkg:npm/engine.io@4.1.2	MEDIUM	1	Highest	7
express-jwt:0.1.3	cpe:2.3:a:auth0:express-jwt:0.1.3:*********	pkg:npm/express-jwt@0.1.3	CRITICAL	2	Highest	9
file-type:11.1.0	cpe:2.3:a:file-type_project:file-type:11.1.0:*********	pkg:npm/file-type@11.1.0	MEDIUM	1	Highest	8

Secure Repository Process Flow



Updating vulnerable dependencies

To update or not to update?



from deps.dev

Agenda

Overview

- Three supply chain-specific attacks
 - Typosquatting
 - Dependency confusion
 - New: manifest confusion
- Making good component choices
- Identifying vulnerabilities in components

Exercise

Taxonomy of malicious commit attack vector

Exercise

SoK: Taxonomy of Attacks on Open-Source Software Supply Chains

Piergiorgio Ladisa SAP Security Research, Université de Rennes 1 piergiorgio.ladisa@sap.com, piergiorgio.ladisa@irisa.fr Henrik Plate SAP Security Research* henrik@endor.ai *now at Endor Labs Matias Martinez Université Polytechnique Hauts-de-France* matias.martinez@upc.edu *now at Universitat Politècnica de Catalunya-BarcelonaTech Olivier Barais Université de Rennes 1, INRIA/IRISA olivier.barais@irisa.fr









Methodology

- Systematic Literature Review and Grey Literature Review to collect
 - All known attack vectors
 - Associated safeguards
- Model the attack vectors in an attack tree and map the safeguards to each vector
- Conduct two user surveys to assess both the taxonomy and utility/cost of safeguards
 - 17 experts
 - 134 developers



Results: Taxonomy of Open-Source Software Supply Chain Attacks

The proposed taxonomy :

- Attacker's perspective
- Positively assessed by 17 experts
- The taxonomy, safeguards and references can be explored online using the Risk Explorer for Software Supply Chan [1]

33

Unique attack vectors

🖀 117

Unique high-level safeguards

[1] https://sap.github.io/risk-explorer-for-software-supply-chains/

Scientific and grey literature references

Risk Explorer for Software Supply Chain



Available online and open-source: https://sap.github.io/risk-explorer-for-software-supply-chains/

Exercise



- Go to the Risk Explorer
 - https://sap.github.io/risk-explorer-for-software-supply-chains
 - https://tinyurl.com/ymz63597
- Go three levels deep in the tree
 - Summarize the attack and possible safeguards

Cost matters: Safeguards Utility & Cost Assessment

Safeguard J_{2}	Safeguard J	Safeguard Junction	Safeguard J	Steguard A<		Safeguard					
Protect production branch 4.2 4.0 2.0 2.0 2 Remove un-used dependencies 4.3 5.0 2.1 2.0 2 1 1 2 2 2 2 1 1 2 3	Protect production branch 42 4.0 2.0 2.0 2 Remove un-used dependencies 4.3 5.0 2.1 2.0 2 1 1 2 2 2 2 1 1 1 2 2 2 2 1	Protect production branch 4.2 4.0 2.0 <th>Protect production branch 42 40 20 20 2 Remove un-used dependencies 43 5.0 2.1 20 2 Version pinning [74] [72] 3.7 3.0 1 2.2 2.0 1 Dependency resolution nules 3.7 3.0 1 2.2 2.0 1 1 User accontinuagement 3.9 4.0 2.6 3.0 1 1 1 4.0 2.6 3.0 1 1 User accontinuout, token protection 1.5 5.0 3.5 2.9 3.0 1 1 1 1.0</th> <th>Protect production branch 4.2 4.0 2.0 <</th> <th></th> <th></th> <th></th> <th>Mean</th> <th>Median</th> <th></th> <th>Mean</th>	Protect production branch 42 40 20 20 2 Remove un-used dependencies 43 5.0 2.1 20 2 Version pinning [74] [72] 3.7 3.0 1 2.2 2.0 1 Dependency resolution nules 3.7 3.0 1 2.2 2.0 1 1 User accontinuagement 3.9 4.0 2.6 3.0 1 1 1 4.0 2.6 3.0 1 1 User accontinuout, token protection 1.5 5.0 3.5 2.9 3.0 1 1 1 1.0	Protect production branch 4.2 4.0 2.0 <				Mean	Median		Mean
Remove un-sued dependencies 4.3 5.0 2.1 2.0 1 1.2 Version pinning [$\frac{64}{2}$] [$\frac{7}{2}$] 3.7 3.0 1 2.2 2.0 1 1.1 Uperentary resulting nelse 1.1 4.0 2.6 3.0 1 1.1 Use of security, quality and beath metrics 3.5 4.0 2.6 3.0 1 1.1 Use of security, quality and beath metrics 3.5 4.0 -4 2.6 3.0 1 1.1 Use of security, quality and beath metrics 3.5 4.0 -4 2.6 3.0 -1 1.1 Use of security, quality and beath metrics 3.5 4.0 -4 2.6 3.0 -1 1.1 Use minimal etc function [5], [76] 1.3 3.0 -2.5 2.0 -1 1.1 Use minimal etc alied SBOM [8] and perform Castion the relases [96] 3.6 3.0 -1 1.2 Maintain detailed SBOM [8] and perform Castion the relating etc [10] 3.6 5.0 -3.3 3.0 -1 1.2 Palemetric excess to system resources of code cascuat during a	Remove un-used dependencies 4.3 5.0 2.1 2.0 2.0 1 1 Version pinning [4] [72] 3.7 3.0 1.0 2.6 3.0 1 1.2 Dependency resolution rules 4.1 4.0 3.0 4.0 2.6 3.0 1 1.2 Secure authencion (e.g., MFA, passoor 4.3 5.0 4.0 2.6 3.0 1 1.1 Secure authencion (e.g., MFA, passoor 4.3 5.0 4.0 2.6 3.0 1 1.1 Use of security, quality and health metrics 3.5 4.0 -4 2.6 3.0 -4 1.1 Use of security, pachicy and health metrics 3.5 4.0 -4 2.6 3.0 -4 1.1 Type metric [50] [76] 3.1 4.0 -4 3.0 -4 1.1 Type metric [51] [76] 3.3 3.0 -2 2.0 -1 1.1 Type metric [50] [76] 3.3 3.0 -2 3.0 -1 1.1 Type metric [50] [76] 3.3 3.0 -1 3	Remove un-used dependencies 4.3 5.0 2.1 2.0 2.0 Version pinning $[\frac{74}{4}]$ [$\frac{72}{2}$] 3.7 3.0 1 2.2 2.0 1.0 Dependency resolution rules 4.1 4.0 2.6 3.0 1.1 2.6 3.0 1.1 1.2 Dependency resolution rules 3.3 4.0 2.6 3.0 1.1 <td>Remove un- and dependencies 4.3 5.0 21 20 2.0 20 Version pinning [2] [7] 3.7 3.0 1 2.2 2.0 1<!--</td--><td>Remove un usued dependencies 4.3 5.0 2.1 2.0 2.1 2.0 2.1 2.0 2.1 2.0 1.1 Dependency resolution rules 3.1 4.0 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1</td><td>Developers</td><td>Protect production branch</td><td>-</td><td>4.2</td><td>4.0</td><td>ad.</td><td>2.0</td></td>	Remove un- and dependencies 4.3 5.0 21 20 2.0 20 Version pinning [2] [7] 3.7 3.0 1 2.2 2.0 1 </td <td>Remove un usued dependencies 4.3 5.0 2.1 2.0 2.1 2.0 2.1 2.0 2.1 2.0 1.1 Dependency resolution rules 3.1 4.0 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1</td> <td>Developers</td> <td>Protect production branch</td> <td>-</td> <td>4.2</td> <td>4.0</td> <td>ad.</td> <td>2.0</td>	Remove un usued dependencies 4.3 5.0 2.1 2.0 2.1 2.0 2.1 2.0 2.1 2.0 1.1 Dependency resolution rules 3.1 4.0 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1	Developers	Protect production branch	-	4.2	4.0	ad.	2.0
Version pinning [74] [72] 3.7 3.0 1.0 2.2 2.0 Dependency resolution nules 1.1 4.0 2.6 3.0 Scoure anthencication (e.g., MFA, password 3.3 5.0 2.6 3.0 Scoure anthencication (e.g., MFA, password 3.5 4.0 2.6 3.0 Use of security, quality and health metrics 3.5 4.0 2.6 3.0 Typo gazd/Typo detection [15], [76] 3.9 4.0 3.1 3.0 4.0 Lie minimal set of trusted build dependencies in through trypophic bashes @0 3.3 3.0 2.5 2.0 3.4 Maintain detailed SBOM (8) and perform 3.2 5.0 3.4 4.0 4.0 4.0 Pheneral build environment (9) 3.6 3.0 4.0 3.3 3.0 4.0 Restrict access to system resources of code cascut during sub build graption graptic transfer grapt 3.7 4.0 3.8 4.0 3.8 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0	Version pinning [$\overline{72}$] 3.7 3.0 1 2.2 2.0 Dependency resolution rules 3.9 4.0 2.6 3.0 User account inducet, MEA password 3.3 5.0 2.6 3.0 Secure aubentication (e.g., MEA password 3.3 5.0 2.6 3.0 Use of security, quality and beatilt metrics 3.5 4.0 2.6 3.0 Type gars/Type detection (15) (76) 3.9 4.0 2.9 4.0 Use of security, quality and beatilt metrics 3.3 3.0 2.5 2.0 Integrity check of dependencies in the relates (b) [11] 3.3 3.0 2.5 2.0 Minitian detailed SBOM (8) and perform 4.2 5.0 3.4 4.0 3.3 3.0 <td>Version pinning [74] [72] 3.7 3.0 1 2.2 2.0 Dependency resolution rules 4.1 4.0 2.6 3.0 User acount management 3.9 4.0 2.6 3.0 Score authentication c_E, MFA, posseed 4.3 5.0 2.6 3.0 User acount management 3.3 5.0 2.6 3.0 Dye part/Pyo detcinin (5) 3.5 4.0 2.6 3.0 Use of accurity, quality and health metrics 3.5 4.0 2.6 3.0 Type part/Pyo detcinin [5], [5], [5] 3.4 4.0 3.1 3.0 Itage in the relaxes (9) [5] 3.3 3.0 2.5 2.0 Maintain detailed SBOM [8] and perform SCA 3.6 3.0 3.0 3.0 3.0 Integrity check of dependancies through [5] 3.6 3.0 3.0 3.0 3.0 3.0 3.0 3.0 Integrity check of dependancies through [5] 3.6 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0</td> <td>Version pinning [74] [72] 3.7 3.0 1 2.2 2.0 Dependency resolution rules 3.4 4.0 2.6 3.0 User accuration industication (e.g., MYA, password; recycle, seisin indust, token posteriori 3.5 3.0 2.6 3.0 Secure authentication (e.g., MYA, password; recycle, seisin indust, token posteriori 3.5 4.0 2.6 3.0 Use of security, quality and health metric 15.5 4.0 2.6 3.0 4.0 Typo gaus/Typo detection [15], [76] 1.0 3.0 3.0 2.5 2.0 3.0 Use of security, quality and health metric 13.3 3.0 2.5 2.0 3.0 Typo gaus/Typo detection [15], [76] 1.0 3.3 3.0 2.5 2.0 3.1 Integrity check of dependencies through ergotographic hashe [0] 3.3 3.0 2.2 3.0 3.0 3.0 Prevent sergits execution 3.7 3.0 3.0 3.0 3.0 3.0 3.0 3.0 Code signing 3.7 4.0 3.3 3.0 3.0 3.0 3.0 3.0 3.0</td> <td>Version pinning [72] 3.7 3.0 1 2.2 2.0 Pependency recolution rules 3.4 4.0 3.6 3.3 User account management 3.5 4.0 2.6 3.0 Scener authentication (e.g., MFA, password 3.5 4.0 2.6 3.0 User account management 3.5 4.0 2.6 3.0 User account management 3.5 4.0 2.6 3.0 User of security, quality and bauth metrics 3.5 4.0 2.6 3.0 Typo guard/Typo detection [15], [76] 3.3 4.0 2.2 2.0 3.1 Use of security, quality and bauth metrics 3.3 3.0 2.2 2.0 3.0 Integrity check of dependencies through crassing battering incompetition batters (0) 3.3 3.0</td> <td></td> <td>Remove un-used dependencies</td> <td></td> <td>4.3</td> <td>5.0</td> <td></td> <td>2.1</td>	Version pinning [74] [72] 3.7 3.0 1 2.2 2.0 Dependency resolution rules 4.1 4.0 2.6 3.0 User acount management 3.9 4.0 2.6 3.0 Score authentication c_E , MFA, posseed 4.3 5.0 2.6 3.0 User acount management 3.3 5.0 2.6 3.0 Dye part/Pyo detcinin (5) 3.5 4.0 2.6 3.0 Use of accurity, quality and health metrics 3.5 4.0 2.6 3.0 Type part/Pyo detcinin [5], [5], [5] 3.4 4.0 3.1 3.0 Itage in the relaxes (9) [5] 3.3 3.0 2.5 2.0 Maintain detailed SBOM [8] and perform SCA 3.6 3.0 3.0 3.0 3.0 Integrity check of dependancies through [5] 3.6 3.0 3.0 3.0 3.0 3.0 3.0 3.0 Integrity check of dependancies through [5] 3.6 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0	Version pinning [74] [72] 3.7 3.0 1 2.2 2.0 Dependency resolution rules 3.4 4.0 2.6 3.0 User accuration industication (e.g., MYA, password; recycle, seisin indust, token posteriori 3.5 3.0 2.6 3.0 Secure authentication (e.g., MYA, password; recycle, seisin indust, token posteriori 3.5 4.0 2.6 3.0 Use of security, quality and health metric 15.5 4.0 2.6 3.0 4.0 Typo gaus/Typo detection [15], [76] 1.0 3.0 3.0 2.5 2.0 3.0 Use of security, quality and health metric 13.3 3.0 2.5 2.0 3.0 Typo gaus/Typo detection [15], [76] 1.0 3.3 3.0 2.5 2.0 3.1 Integrity check of dependencies through ergotographic hashe [0] 3.3 3.0 2.2 3.0 3.0 3.0 Prevent sergits execution 3.7 3.0 3.0 3.0 3.0 3.0 3.0 3.0 Code signing 3.7 4.0 3.3 3.0 3.0 3.0 3.0 3.0 3.0	Version pinning [72] 3.7 3.0 1 2.2 2.0 Pependency recolution rules 3.4 4.0 3.6 3.3 User account management 3.5 4.0 2.6 3.0 Scener authentication (e.g., MFA, password 3.5 4.0 2.6 3.0 User account management 3.5 4.0 2.6 3.0 User account management 3.5 4.0 2.6 3.0 User of security, quality and bauth metrics 3.5 4.0 2.6 3.0 Typo guard/Typo detection [15], [76] 3.3 4.0 2.2 2.0 3.1 Use of security, quality and bauth metrics 3.3 3.0 2.2 2.0 3.0 Integrity check of dependencies through crassing battering incompetition batters (0) 3.3 3.0		Remove un-used dependencies		4.3	5.0		2.1
Dependency resolution nets 4.1 4.0 -1 2.6 3.0 User account management recycle, assistic mitters, the protection) 3.9 4.0 -1 2.6 3.0 User account management recycle, assistic mitters, the protection) 3.5 4.0 -1 2.6 3.0 Use of security, quality and health metrics: 3.5 4.0 -1 2.6 3.0 Type guard/Type detection [15]. [76] Use minimal set of trasted baild dependen- cies in the release [96] [20] 3.9 4.0 -1 2.9 4.0 Maintain detailed SBOM [6] and perform SCA 3.3 3.0 2.2 3.0 Protental baild environment [9] 3.6 3.0 3.3 3.0 Protental baild environment [9] 3.7 4.0 -1 3.3 3.0 Protental baild environment [9] 3.7 4.0 -1 3.3 3.0 Protental baild environment [9] 3.7 4.0 -1 3.3 3.0 Code signing 3.7 4.0 -1 3.3 3.0 -1	Dependency resolution rules User account familing and Secure authentication (e.g., MEA, passwort recycle, securitioned, tokan protection Use of security, quality and health metrics. 3.9 4.0 2.6 3.0 View of the security of the security of the security user of the security of the security of the security of the security is in the release job [5]. 3.9 4.0 2.6 3.0 Type guard/Type detection [15]. [76] Use minimal set of transfer baild dependen- ies in the release job [5]. 3.3 3.0 2.2 2.0 Maintain detailed SBOM [6] and perform SCA. 3.3 3.0 2.2 2.0 3.0 Prevent stript execution account during each build step [5]. 3.6 3.8 4.0 3.3 3.0 4.0 Code signing 3.7 4.0 3.3 3.0 4.0 3.3 3.0 Code signing 3.7 4.0 3.3 3.0 3.3 3.0 3.3 3.0 Integrat Open Source valuerability samer uso CCD projection 3.6 4.0 3.3 3.0 3.3 3.0 Code signing 3.4 4.0 3.3 3.0 3.3 3.0 Preve	Dependency resolution nules User account magnetistic Secure authentication (c_2 , MFA, password recycle, seison innocut, skep program innocut, skep program verse, set innocut, skep program istricture, authentication (c_2 , MFA, password recycle, setsion innocut, skep program istricture, authentication (c_1), $\overline{f(5)}$ Use of security, quality and health metrics 3.5 4.0 2.6 3.0 Type pauel/Type detection $[\overline{15}], [\overline{76}]$ Use minimal set of trusted build dependen- cies in the release (bb [3] 3.9 4.0 2.5 2.0 Maintain detailed SBOM [8] and perform SCA 3.3 3.0 2.2 3.0 Prevent script excession SCA 3.6 3.0 4.0 3.3 3.0 Difference excuted during each build step [5] 3.6 3.0 3.8 4.0 Difference excuted during each build step [6] 3.7 4.0 3.3 3.0 Code signing 3.7 4.0 3.3 3.0 Integrate Open-Source vulnerability scamer into CLCD pipeline from CLCD pipeline sessment, volnerability accounter ($\overline{10}$) 3.4 4.0 3.3 3.0 Prevents traiget the relaxed peakees Addi, scearity assessment, volnerability accounter ($\overline{10}$) 3.4 4.0 4.1 4.0	Dependency resolution magnetic 31 40 26 30 Secure authentication (e.g., MFA, passwort recycle, section intens), tokan protection 35 40 26 30 Secure authentication (e.g., MFA, passwort recycle, section intens), tokan protection 35 40 26 30 Use of security, quality and health metrics 35 40 26 30 30 Type grauf/typo detection [15], [76] 39 40 31 30 30 30 Use of security, quality and health metrics 33 30 22 30 33 30 <	$ \begin{array}{c c c c c c c c c c c c c c c c c c c $		Version pinning [74] [72]		3.7	3.0	44	2.2
Secure authentication (e.g., MPA, password) 4.3 5.0 2.0 3.0 Use of security, quality and health metrics 3.5 4.0 2.6 3.0 Type garent/ype detection [15], [76] 3.9 4.0 2.0 3.0 Use of security, quality and health metrics 3.5 3.0 2.2 4.0 Type garent/ype detection [15], [76] 3.9 4.0 3.1 2.0 3.0 Use of security, quality and health metrics 3.3 3.0 3.0 2.2 2.0 Use minimal set of trusted build dependen- cies in the relates [ob [3]) 3.3 3.0 3.0 2.5 2.0 Minimin detailed SBOM [8] and perform 4.2 5.0 -4 3.4 4.0 SCA Ephemeral build environment [9] 3.6 3.0 4.0 3.8 4.0 Publicing request treise 3.6 1.4 4.0 3.8 4.0 Publicing request treise 3.7 4.0 3.3 3.0 Code signing 3.7 4.0 3.3 3.0 Code signing 3.7 4.0 3.3 3.0 Use of declated build service [7] 3.6 4.0 3.3 3.0 Code signing 1.5 4.0 3.3 </td <td>Scoure authentication (e.g., MFA, password) 4.3 5.0 -4 2.0 3.0 Use of security, quality and health metrics 3.5 4.0 -4 2.0 3.0 Type guard/Typo detection [15], [6], Use minimal set of tracked build dependen- cies in the relaxes job [3]. 3.0 -4 2.0 3.0 Integrity check of dependencies through eryptographic hashes [9]. 3.3 3.0 - 2.5 2.0 Mannin detailed SBOM [8] and perform SCA 3.0 3.0 - 2.5 2.0 Minimin detailed SBOM [8] and perform SCA 3.0 3.0 - 2.5 2.0 Minimin detailed SBOM [8] and perform SCA 3.0 3.0 - 2.5 3.0 Prevent script execution in Circle pipetire mestores 7.7 3.0 - 3.3 3.0 3.3 3.0 Code signing 3.7 4.0 - 3.3 3.0 - 3.3 3.0 Code signing 3.7 4.0 - 3.3 3.0 - 3.3 3.0 - 3.3 <td< td=""><td>Secure anthenicianics (c.g., MEA, password) 4.3 5.0 2.9 3.0 Use of security, quality and health metrics 3.5 4.0 4 2.6 3.0 Type part/Pype detection [55] 7.0 4.3 5.0 4.0 2.6 3.0 Type part/Pype detection [55] 7.0 4.1 4.0 2.5 2.0 Maintain detailed SBOM [8] and perform 4.1 4.0 2.5 2.0 Maintain detailed SBOM [8] and perform 4.2 5.0 3.4 4.0 SCA SCA 4.0 4.0 3.3 3.0 2.9 3.0 Maintain detailed SBOM [8] and perform 4.2 5.0 3.4 4.0 3.3 3.0 2.9 3.0 Prevent scipt execution 3.7 5.0 3.3 3.0 <td< td=""><td>Secure anthentication (e.g., MFA, password) 4.3 5.0 -4 2.0 30 Ure of security, quality and health metrics 3.5 4.0 -4 2.6 39 Type guard/Ypo detection [15], [76] 3.9 4.0 -4 2.6 30 Ure of security, quality and health metrics 3.5 4.0 -4 2.4 30 Type guard/Ypo detection [15], [76] 3.3 4.0 -4 2.5 20 Use in mininal set of trutted build dependen- cies in the etalse job [51] 3.3 3.0 -5 2.0 Miniania detailed SBOM [8] and perform ACA 4.2 5.0 -4 3.4 4.0 Prevent seriptic bashes (9) 3.6 3.0 -5 2.0 3.0 Prevent seriptic securition 3.6 3.0 -6 3.5 4.0 Reproducible baytim resources of cells 5.0 -4 3.3 3.0 Code signing 5.7 4.0 -6 3.3 3.0 Code signing 3.7 4.0 -6 3.3 3.0 Code signing 3.3 4.0 -6 3.3 3.0 Prevents exignating the released packages Addis security sessmetric hyterine sources of cells at 3.3 3.0 -6 3.3</td><td>$\begin{array}{c cccc} Secure authentication (e.g., MFA, password) & 4.3 & 5.0 & -1 & 2.0 & 3.0 \\ recycle, sexion innovat, tacker protection \\ recycle, sexion innovat, tacker protection \\ Ue of security, quality and health metrics & 2.5 & 4.0 & -1 & 2.6 & 3.0 \\ \hline Ue of security, quality and health metrics & 2.5 & 4.0 & -1 & 2.4 & 3.0 \\ \hline Ue minimal set of trusted build dependen-cies in the relates job [51] \\ Integrity check of dependencies through & 3.3 & 3.0 & -2 & 2.5 \\ Prevent serie tackers in the second of the second adminimal dealed SBOM [6] and perform & 4.2 & 5.0 & -1 & 3.4 & 4.0 \\ SCA & Ephemeral build environment [9] & 3.6 & 3.0 & -1 & 3.0 & 3.0 \\ Prevent script excaption & 3.7 & 3.0 & -1 & 3.0 & 3.0 \\ Public graphic graphic second adminimation of the second adminimation and the second adminimation admini$</td><td></td><td>Dependency resolution rules User account management</td><td></td><td>4.1 3.9</td><td>4.0 4.0</td><td>1</td><td>2.6</td></td<></td></td<></td>	Scoure authentication (e.g., MFA, password) 4.3 5.0 -4 2.0 3.0 Use of security, quality and health metrics 3.5 4.0 -4 2.0 3.0 Type guard/Typo detection [15], [6], Use minimal set of tracked build dependen- cies in the relaxes job [3]. 3.0 -4 2.0 3.0 Integrity check of dependencies through eryptographic hashes [9]. 3.3 3.0 - 2.5 2.0 Mannin detailed SBOM [8] and perform SCA 3.0 3.0 - 2.5 2.0 Minimin detailed SBOM [8] and perform SCA 3.0 3.0 - 2.5 2.0 Minimin detailed SBOM [8] and perform SCA 3.0 3.0 - 2.5 3.0 Prevent script execution in Circle pipetire mestores 7.7 3.0 - 3.3 3.0 3.3 3.0 Code signing 3.7 4.0 - 3.3 3.0 - 3.3 3.0 Code signing 3.7 4.0 - 3.3 3.0 - 3.3 3.0 - 3.3 <td< td=""><td>Secure anthenicianics (c.g., MEA, password) 4.3 5.0 2.9 3.0 Use of security, quality and health metrics 3.5 4.0 4 2.6 3.0 Type part/Pype detection [55] 7.0 4.3 5.0 4.0 2.6 3.0 Type part/Pype detection [55] 7.0 4.1 4.0 2.5 2.0 Maintain detailed SBOM [8] and perform 4.1 4.0 2.5 2.0 Maintain detailed SBOM [8] and perform 4.2 5.0 3.4 4.0 SCA SCA 4.0 4.0 3.3 3.0 2.9 3.0 Maintain detailed SBOM [8] and perform 4.2 5.0 3.4 4.0 3.3 3.0 2.9 3.0 Prevent scipt execution 3.7 5.0 3.3 3.0 <td< td=""><td>Secure anthentication (e.g., MFA, password) 4.3 5.0 -4 2.0 30 Ure of security, quality and health metrics 3.5 4.0 -4 2.6 39 Type guard/Ypo detection [15], [76] 3.9 4.0 -4 2.6 30 Ure of security, quality and health metrics 3.5 4.0 -4 2.4 30 Type guard/Ypo detection [15], [76] 3.3 4.0 -4 2.5 20 Use in mininal set of trutted build dependen- cies in the etalse job [51] 3.3 3.0 -5 2.0 Miniania detailed SBOM [8] and perform ACA 4.2 5.0 -4 3.4 4.0 Prevent seriptic bashes (9) 3.6 3.0 -5 2.0 3.0 Prevent seriptic securition 3.6 3.0 -6 3.5 4.0 Reproducible baytim resources of cells 5.0 -4 3.3 3.0 Code signing 5.7 4.0 -6 3.3 3.0 Code signing 3.7 4.0 -6 3.3 3.0 Code signing 3.3 4.0 -6 3.3 3.0 Prevents exignating the released packages Addis security sessmetric hyterine sources of cells at 3.3 3.0 -6 3.3</td><td>$\begin{array}{c cccc} Secure authentication (e.g., MFA, password) & 4.3 & 5.0 & -1 & 2.0 & 3.0 \\ recycle, sexion innovat, tacker protection \\ recycle, sexion innovat, tacker protection \\ Ue of security, quality and health metrics & 2.5 & 4.0 & -1 & 2.6 & 3.0 \\ \hline Ue of security, quality and health metrics & 2.5 & 4.0 & -1 & 2.4 & 3.0 \\ \hline Ue minimal set of trusted build dependen-cies in the relates job [51] \\ Integrity check of dependencies through & 3.3 & 3.0 & -2 & 2.5 \\ Prevent serie tackers in the second of the second adminimal dealed SBOM [6] and perform & 4.2 & 5.0 & -1 & 3.4 & 4.0 \\ SCA & Ephemeral build environment [9] & 3.6 & 3.0 & -1 & 3.0 & 3.0 \\ Prevent script excaption & 3.7 & 3.0 & -1 & 3.0 & 3.0 \\ Public graphic graphic second adminimation of the second adminimation and the second adminimation admini$</td><td></td><td>Dependency resolution rules User account management</td><td></td><td>4.1 3.9</td><td>4.0 4.0</td><td>1</td><td>2.6</td></td<></td></td<>	Secure anthenicianics (c.g., MEA, password) 4.3 5.0 2.9 3.0 Use of security, quality and health metrics 3.5 4.0 4 2.6 3.0 Type part/Pype detection [55] 7.0 4.3 5.0 4.0 2.6 3.0 Type part/Pype detection [55] 7.0 4.1 4.0 2.5 2.0 Maintain detailed SBOM [8] and perform 4.1 4.0 2.5 2.0 Maintain detailed SBOM [8] and perform 4.2 5.0 3.4 4.0 SCA SCA 4.0 4.0 3.3 3.0 2.9 3.0 Maintain detailed SBOM [8] and perform 4.2 5.0 3.4 4.0 3.3 3.0 2.9 3.0 Prevent scipt execution 3.7 5.0 3.3 3.0 <td< td=""><td>Secure anthentication (e.g., MFA, password) 4.3 5.0 -4 2.0 30 Ure of security, quality and health metrics 3.5 4.0 -4 2.6 39 Type guard/Ypo detection [15], [76] 3.9 4.0 -4 2.6 30 Ure of security, quality and health metrics 3.5 4.0 -4 2.4 30 Type guard/Ypo detection [15], [76] 3.3 4.0 -4 2.5 20 Use in mininal set of trutted build dependen- cies in the etalse job [51] 3.3 3.0 -5 2.0 Miniania detailed SBOM [8] and perform ACA 4.2 5.0 -4 3.4 4.0 Prevent seriptic bashes (9) 3.6 3.0 -5 2.0 3.0 Prevent seriptic securition 3.6 3.0 -6 3.5 4.0 Reproducible baytim resources of cells 5.0 -4 3.3 3.0 Code signing 5.7 4.0 -6 3.3 3.0 Code signing 3.7 4.0 -6 3.3 3.0 Code signing 3.3 4.0 -6 3.3 3.0 Prevents exignating the released packages Addis security sessmetric hyterine sources of cells at 3.3 3.0 -6 3.3</td><td>$\begin{array}{c cccc} Secure authentication (e.g., MFA, password) & 4.3 & 5.0 & -1 & 2.0 & 3.0 \\ recycle, sexion innovat, tacker protection \\ recycle, sexion innovat, tacker protection \\ Ue of security, quality and health metrics & 2.5 & 4.0 & -1 & 2.6 & 3.0 \\ \hline Ue of security, quality and health metrics & 2.5 & 4.0 & -1 & 2.4 & 3.0 \\ \hline Ue minimal set of trusted build dependen-cies in the relates job [51] \\ Integrity check of dependencies through & 3.3 & 3.0 & -2 & 2.5 \\ Prevent serie tackers in the second of the second adminimal dealed SBOM [6] and perform & 4.2 & 5.0 & -1 & 3.4 & 4.0 \\ SCA & Ephemeral build environment [9] & 3.6 & 3.0 & -1 & 3.0 & 3.0 \\ Prevent script excaption & 3.7 & 3.0 & -1 & 3.0 & 3.0 \\ Public graphic graphic second adminimation of the second adminimation and the second adminimation admini$</td><td></td><td>Dependency resolution rules User account management</td><td></td><td>4.1 3.9</td><td>4.0 4.0</td><td>1</td><td>2.6</td></td<>	Secure anthentication (e.g., MFA, password) 4.3 5.0 -4 2.0 30 Ure of security, quality and health metrics 3.5 4.0 -4 2.6 39 Type guard/Ypo detection [15], [76] 3.9 4.0 -4 2.6 30 Ure of security, quality and health metrics 3.5 4.0 -4 2.4 30 Type guard/Ypo detection [15], [76] 3.3 4.0 -4 2.5 20 Use in mininal set of trutted build dependen- cies in the etalse job [51] 3.3 3.0 -5 2.0 Miniania detailed SBOM [8] and perform ACA 4.2 5.0 -4 3.4 4.0 Prevent seriptic bashes (9) 3.6 3.0 -5 2.0 3.0 Prevent seriptic securition 3.6 3.0 -6 3.5 4.0 Reproducible baytim resources of cells 5.0 -4 3.3 3.0 Code signing 5.7 4.0 -6 3.3 3.0 Code signing 3.7 4.0 -6 3.3 3.0 Code signing 3.3 4.0 -6 3.3 3.0 Prevents exignating the released packages Addis security sessmetric hyterine sources of cells at 3.3 3.0 -6 3.3	$ \begin{array}{c cccc} Secure authentication (e.g., MFA, password) & 4.3 & 5.0 & -1 & 2.0 & 3.0 \\ recycle, sexion innovat, tacker protection \\ recycle, sexion innovat, tacker protection \\ Ue of security, quality and health metrics & 2.5 & 4.0 & -1 & 2.6 & 3.0 \\ \hline Ue of security, quality and health metrics & 2.5 & 4.0 & -1 & 2.4 & 3.0 \\ \hline Ue minimal set of trusted build dependen-cies in the relates job [51] \\ Integrity check of dependencies through & 3.3 & 3.0 & -2 & 2.5 \\ Prevent serie tackers in the second of the second adminimal dealed SBOM [6] and perform & 4.2 & 5.0 & -1 & 3.4 & 4.0 \\ SCA & Ephemeral build environment [9] & 3.6 & 3.0 & -1 & 3.0 & 3.0 \\ Prevent script excaption & 3.7 & 3.0 & -1 & 3.0 & 3.0 \\ Public graphic graphic second adminimation of the second adminimation and the second adminimation admini$		Dependency resolution rules User account management		4.1 3.9	4.0 4.0	1	2.6
Use of security, quality and health metrics 3.5 4.0 -1 2.6 3.0 Type guad/Type detection [15], [76], Use minimal set of trusted build dependen- cies in the release job [31], Trustegrity check of dependencies through 3.3 3.0 2.9 4.0 Integrity check of dependencies through Crystoproprinc hashes [20] 3.3 3.0 2.5 2.0 Maintain detailed SBOM [8] and perform AC Papermet build environment [9] 3.6 3.0 2.9 3.0 Prevent scipt execution Paul/Merge request review caused during and build were [9]] 3.6 5.0 3.8 4.0 Code signing Integrate Open-Source Use of dedicated build ervice [9] Prevent sequenting the review Use of dedicated build ervice [9] Prevent sequenting the review Prevent sequenting the review [9] 3.8 4.0 3.3 3.0 Code signing Integrate open-Source 3.7 4.0 3.3 3.0 Use of dedicated build ervice [9] Prevent sequenting the review of [9] 3.6 4.0 3.3 3.0 Use of dedicated build ervice [9] Prevent sequenting the review of [9] 3.6 4.0 3.3 3.0	Use of security, quality and health metric 55 4.0 -4 2.6 3.0 Type guard/type detection [15], [6] 3.9 4.0 -4 2.9 4.0 Use minimal set of trusted build dependencies in the relates job [5] 3.3 3.0 - 2.5 20 Integrity check of dependencies through regarding thatse [9] 3.3 3.0 - 2.5 20 Maintain detailed SBOM [8] and perform SCA 3.0 3.0 - 2.9 3.0 Prevent script cexecution region regarding the resource of role exceuted during each build steps [5] 3.7 3.0 3.8 4.0 Code signing 3.7 4.0 - 3.3 3.0 - 3.3 3.0 - 3.3 3.0	Use of security, quality and health metrics 3.5 4.0 -1 2.6 3.0 Type pare/Type detection [5]. [76] 3.9 4.0 -1 2.4 4.0 Use minimal set of mested build dependen. 3.1 4.0 -1 2.5 2.0 Integrity check of dependencies through 3.3 3.0 -2.5 2.0 Minimal set alled SBOM [8] and perform 4.2 5.0 -1 3.4 4.0 Spenneral build environment [9] 3.6 3.0 -2.2 3.0 Prevent script execution 3.7 1.0 -10.0 3.0 TolkiMorge request houring security of the script environment [9] -3.6 3.0 -2.9 3.0 Octor signing 3.7 4.0 -1 3.3 3.0 Torder signing 3.7 4.0 -3.3 3.0 Torder signing 3.7 4.0 -3.3 3.0 Torder signing 3.4 4.0 -3.3 3.0 Torder signing 3.4 4.0 -3.3	Use of security, quality and health metrics 5.5 4.0 -1 2.6 3.0 Type guard/Type detection [15], [76] 3.9 4.0 -1 2.9 4.0 Use in mininal set of much build dependen- cies in the relation [36] 3.0 -1 2.9 4.0 Integrity check of dependencies through Cryptographic hashe [9] 3.3 3.0 -2 2.0 Maintain detailed SBOM [8] and perform CA. 3.0 1 2.0 3.0 -2 2.0 Prevent sergits execution Prevent sergits execution 3.0 1 2.0 3.0 -3 3.0	Use of security, quality and health metrics 3.8 4.0 -1 2.6 3.0 Typo guard/Typo detection [15], [76], Use minual set of trated baild dependen- cies in the release job [31] 3.9 4.0 -1 2.9 4.0 Introprogrammed bases [30] Introprogrammed bases [30] 3.3 3.0 -2 2.0 Minimia detailed SBOM [3] and perform SCA. 4.2 5.0 -1 3.4 4.0 Code Signing 3.7 3.0 -2 2.9 3.0 Puellement build environment [9] 3.6 5.0 -1 3.3 3.0 Puellement stuid environment [9] 3.7 4.0 -1 3.3 3.0 Puellement stuid environment [9] 3.6 4.0 -1 3.3 3.0 Puellement stuid environment [9] 3.7 4.0 -1 3.3 3.0 Puellement stuid environment [9] 3.6 4.0 -1 3.3 3.0 Use of deticated build stars [11] 3.7 4.0 -1 3.3 3.0 Use of deticated build sta		Secure authentication (e.g., MFA, passw recycle, session timeout, token protecti	vord on)	4.3	5.0	-	2.9
Type guard/Type detection $[15], [76]$ Use minimal set of trusted Faild dependen- is in the relases $[051]$ 3.9 4.0 3.1 2.9 6.0 Integrity check of dependencies through Corporaptic handles $[33]$ 3.0 2.5 20 Minitain detailed SBOM (§) and perform 3.3 3.0 2.5 20 Minitain detailed SBOM (§) and perform 3.6 3.6 3.6 2.9 3.0 Provent script execution 3.6 3.0 4.0 3.6 3.6 2.9 3.0 Publicharge request review 3.6 4.0 4.0 4.0 3.8 4.0 Code signing 3.7 4.0 4.0 4.0 3.3 3.0 Code signing 3.7 4.0 4.0 4.0 4.0 3.3 3.0 Code signing 3.7 4.0 -10 3.3 3.0 Code signing 3.7 4.0 -10 3.3 3.0 Use of defined build service $[9]$ 5.0 4.0	Typo guard/Typo detection [15], [76] Use minimal set of trusted bial dependen- is in the relases [16] 3.9 4.0 3.1 3.0 Integrity check of dependencies through Typopraphic handle 3.3 3.0 2.5 20 Minimal dealled SBOM [8] and perform SCA 3.0 3.0 3.2 2.9 3.0 Prevent script execution 3.6 3.0 4.0 3.4 4.0 SCA Ephement build environment [9] 3.6 3.0 4.0 3.3 3.0 Prevent script execution 3.7 4.0 4.0 3.3 3.0 3.3 3.0 Code signing 3.7 4.0 3.3 3.0 3.3 3.0 Code signing 3.6 4.0 3.3 3.0 3.3 3.0 <td< td=""><td>Typo guard/Typo detection [15], [6] 3.9 4.0 3.1 3.0 Use minimal set of trasted biald dependen- tion there leads 05 (1) 1.1 4.0 3.1 3.0 Integrity check of dependencies through propergraphic hashes (9) 3.3 3.0 2.5 20 Maintain detailed SBOM (8) and perform SCA 3.6 3.0 4.0 3.4 4.0 Prevent script execution 3.7 3.6 3.0 4.0 3.3 3.0 2.9 3.0 Prevent script execution 3.7 3.6 3.0 4.0 3.3 3.0</td><td>Typo guard/Typo detection [15], [76] Use minimal set of trained build dependen- is in the relaxed boild dependen- sin the relaxed boild dependen- tion the relaxed boild dependen- boild dependencies through Maintain detailed SBOM [8] and perform CA Ephemeral build environment [9] 3.0 2.2 20 Maintain detailed SBOM [8] and perform CA Ephemeral build environment [9] 3.6 3.1 2.2 3.0 Phil/Merge ruppest review CA executed during each build steps [51] 3.7 4.0 3.8 4.0 Code signing Linggrato Open-Scorece ulterability scame Auf, secutiry assement, vulterability ass</td><td>$\begin{array}{cccccccccccccccccccccccccccccccccccc$</td><td></td><td>Use of security, quality and health met</td><td>rics</td><td>3.5</td><td>4.0</td><td></td><td>2.6</td></td<>	Typo guard/Typo detection [15], [6] 3.9 4.0 3.1 3.0 Use minimal set of trasted biald dependen- tion there leads 05 (1) 1.1 4.0 3.1 3.0 Integrity check of dependencies through propergraphic hashes (9) 3.3 3.0 2.5 20 Maintain detailed SBOM (8) and perform SCA 3.6 3.0 4.0 3.4 4.0 Prevent script execution 3.7 3.6 3.0 4.0 3.3 3.0 2.9 3.0 Prevent script execution 3.7 3.6 3.0 4.0 3.3 3.0	Typo guard/Typo detection [15], [76] Use minimal set of trained build dependen- is in the relaxed boild dependen- sin the relaxed boild dependen- tion the relaxed boild dependen- boild dependencies through Maintain detailed SBOM [8] and perform CA Ephemeral build environment [9] 3.0 2.2 20 Maintain detailed SBOM [8] and perform CA Ephemeral build environment [9] 3.6 3.1 2.2 3.0 Phil/Merge ruppest review CA executed during each build steps [51] 3.7 4.0 3.8 4.0 Code signing Linggrato Open-Scorece ulterability scame Auf, secutiry assement, vulterability ass	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$		Use of security, quality and health met	rics	3.5	4.0		2.6
Use minimal set of truited ball dependents 4.1 4.0 3.1 3.0 Integrity check of dependencies through cryptographic handle (P) 3.3 3.0 3.4 4.0 3.4 Maintain detailed SBOM (B) and perform SCA 2.5 2.0 3.4 4.0 3.4 Color and the detailed SBOM (B) and perform SCA 3.6 3.0 3.4 4.0 3.4 Provest series resources of code scenario dring such build steps (SI) 3.6 5.0 3.3 4.0 3.3 3.0 3.4 Code signing 3.7 4.0 4.0 3.3 3.0 3.4 3.0 3.4 3.0 3.4 3.0	Use minimal set of frusted build dependen. 4.1 4.0 3.1 3.0 3.0 Integrity check of dependencies through corporagriths handle (P) 3.3 3.0 3.3 3.0 2.5 2.0 Maintain detailed SBOM (§) and perform 4.2 5.0 3.4 4.0 4.1 4.0 4.1 4.0 4.1	Use minimal set of most build dependent 4.1 4.0 3.1 3.0 Integrity check of dependencies through cryptographic hashes (9) 3.3 3.0 2.2 2.0 Minimal actual of SBOM (6) and perform 4.2 5.0 3.4 4.0 3.5 3.0 2.5 2.0 Minimal actual of SBOM (6) and perform 4.2 5.0 3.4 4.0 3.6 3.0 2.9 3.0 3.0 Prevent script execution 3.7 3.0	Use minimal set of mixed build dependen- ior in the rotation job [5] 4.1 4.0 3.3 3.0 Integrity check of dependencies through Cryptographic hashes (9) 3.3 3.0 2.5 2.0 Maintain detailed SBOM [8] and perform SCA 4.2 5.0 3.4 4.0 3.4 Prevent script execution 3.7 3.0 3.0 3.5 3.0 Prevent script execution 3.7 4.0 3.3 3.0 3.5 4.0 Difference is bystem resources of code executed during each build steps [5] 3.7 4.0 3.3 3.0 <td>Use minimal set of mosteb build dependent: 4.1 4.0 3.1 3.0 Integrity check of dependenties through cryotographic basks (0) 3.3 3.0 2.2 2.0 Maintain detailed SBOM [8] and perform SCA 3.6 3.0 3.4 4.0 3.4 Agentation detailed SBOM [8] and perform SCA 3.6 3.0 3.0 3.4 4.0 3.4 Present scatter 3.6 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 Pull/Merge request review concert of code scatted diaring each build aspin [51] 3.7 4.0 4.0 3.3 3.0</td> <td></td> <td>Typo guard/Typo detection [15], [76]</td> <td></td> <td>3.9</td> <td>4.0</td> <td>-1</td> <td>1 2.9</td>	Use minimal set of mosteb build dependent: 4.1 4.0 3.1 3.0 Integrity check of dependenties through cryotographic basks (0) 3.3 3.0 2.2 2.0 Maintain detailed SBOM [8] and perform SCA 3.6 3.0 3.4 4.0 3.4 Agentation detailed SBOM [8] and perform SCA 3.6 3.0 3.0 3.4 4.0 3.4 Present scatter 3.6 3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0 Pull/Merge request review concert of code scatted diaring each build aspin [51] 3.7 4.0 4.0 3.3 3.0		Typo guard/Typo detection [15], [76]		3.9	4.0	-1	1 2.9
Integrity check of dependencies through cryptographic hashe (0) 3.3 3.0 2.5 2.0 1 Maintain detailed SBOM [8] and perform SCA. 4.2 5.0 1 3.4 4.0 -d. SCA. Fighemeral build environment [9] 3.6 3.0 12 2.9 3.0 -d. Perotent services 3.7 3.0 10 3.0 -d. -d. Retrict receives on first execution of straining and build steps [5]] 3.7 4.0 -d. 3.3 3.0 -d. Code signing 3.7 4.0 -d. 3.3 3.0 -d. -d. Use of declarate build steps [6] 3.6 4.0 -d. 3.3 3.0 -d. Use of declarate build steps [6] 3.6 4.0 -d. 3.3 3.0 -d. Use of declarate build steps [6] 3.6 4.0 -d. 3.3 3.0 -d. Provents equining the charded packing 3.8 4.0 -d. 3.3 3.0 -d.	Integrity check of dependencies through cryptographic basics [0] 3.3 3.0 2.5 2.0 J Minimiza detailed SBOM [8] and perform SCA 4.2 5.0 J 3.4 4.0 J SCA SCA 3.0 J 2.9 3.0 J J J Prevent script execution 3.7 3.0 J 2.9 3.0 J	Integrity check of dependencies through 3.3 3.0 2.5 2.0	Integrity check of dependencies through cryptopriptic lastic [0] 33 30 25 20 1 Minimin detailed SBOM [6] and perform SCA 36 30 29 30 40 4 SCA 30 30 29 30 1 30 30 40 4 Prevent script 37 30 40 40 40 33 30 40 30 40 30 40 30 40 30 40 30 40	Integrity check of dependencies through 3.3 3.0 \sim 2.5 2.0 \rightarrow Maintain detailed SBOM (§) and perform 4.2 5.0 \rightarrow 3.4 4.0 \rightarrow SCA Epheneral build environment (9) 3.6 3.0 \rightarrow 2.9 3.0 \rightarrow Prevent script execution 3.7 3.0 \rightarrow 3.8 4.0 \rightarrow Pull/Merge request review 4.6 5.0 \rightarrow 3.8 4.0 \rightarrow Code signing 3.7 4.0 \rightarrow \rightarrow 3.3 3.0 \rightarrow Use of dedicated build service (7) 3.6 4.0 \rightarrow 3.3 3.0 \rightarrow Use of dedicated build service (7) 3.6 4.0 \rightarrow 3.3 3.0 \rightarrow Reproducible builds 4.2 5.0 \rightarrow 4.1 4.0 \rightarrow Reproducible builds (51) 2.9 3.0 \rightarrow \rightarrow 3.3 3.0 \rightarrow		Use minimal set of trusted build depend cies in the release job [51]	den-	4.1	4.0		3.1
Maintain detailed SBOM [8] and perform 4.2 5.0 3.4 4.0 SCA Ca Epheneral build environment [9] 3.6 30 3.7 3.0 3.8 4.0 Powers rejet execution 3.7 4.0 4.0 3.3 3.0 Poll/Mage request review 6.40 5.0 3.3 3.0 Code signing 3.7 4.0 3.3 3.0 Integrate Open-Source vulnerability scanner into CICD profile 3.6 4.0 3.3 3.0 Use of dedicate build strive [7] 3.6 4.0 3.3 3.0	Mainaina deallad SBOM [8] and perform 42 50 34 40 adl SCA SCA 30 40 34 40 adl Ephernarab baild environment [9] 36 30 40 30 30 40 Prevent script execution 37 300 40 40 40 Retirt access to yatem resource of code executed during each baid steps [6] 40 40 40 40 40 Code signing 37 40 40 31 30 40 Integrate Open-Source vulnenbility scamer into (CLC) Deptine 35 50 43 30 41 Preventive squitting the relaxed processource [9] 31 30 41 40 40	Maintini detailed SBOM (ii) and perform 4.2 5.0 3.4 4.0 4.1 SCA Ephennel build environment (ii) 3.6 3.0 4.2 3.0 3.0 Prevent script execution 3.7 3.0 4.0 3.0 3.0 3.0 Prevent script execution 3.7 3.0 4.0 4.0 3.3 3.0 Restrict access to system resources of code signing 3.7 4.0 4.3 3.0 4.0 Inforgent Cyres-Source viborrability scamer 3.8 4.0 4.0 3.3 3.0 Inforgent Cyres-Source viborrability scamer 3.8 4.0 4.0 3.3 3.0 Preventive squitting the reloased periods 3.6 4.0 4.1 4.0 Preventive squitting the reloased periods 3.4 4.0 4.1 4.0	Maintain detailed SBOM [8] and perform 4.2 5.0 -10 3.4 4.0 -11 SCA Ephenerati build environment [9] 3.0 30 12 2.9 3.0 -1 Prevent service textution 3.7 3.0 1.0 3.0 3.0 -1 Pull Merge request review 4.6 5.0 -10 3.3 3.0 -1 Reprint concess to system resources of code 4.0 4.0 4.0 -10 3.3 3.0 -1 Code signing 3.7 4.0 -10 3.3 3.0 -1 Integrate Open-Source valuerability scatter 7 3.8 4.0 -10 3.3 3.0 Use of dubates huid service [9] 5.6 4.0 4.0 2.0 3.0 Preventive sequating the released packages 3.1 3.0 -1 -1 3.3 3.0 Preventive sequating the released packages -13 3.0 -1 -1 -1 -1 Reproducible builds 5.1 3.0 -1 -1 -1 -1	Maintain detailed BROM [8] and perform 4.2 5.0		Integrity check of dependencies thro cryptographic hashes [9]	ugh	3.3	3.0	and the second sec	2.5
Ephemeral build environment [9] 3.6 3.0 -4.2 2.9 3.0 -4. Prevent script execution 3.7 3.0 -1. 3.0 3.0 3.0 -1. Pull-Regress request review executed during each build serpe [3] 4.0 4.0 -4.0 -3.3 3.0 -1. Code signing 3.7 3.0 -1. -3.3 3.0 -1. Integrate Open-Source vulnerability scamer 3.6 4.0 -1. 3.3 3.0 -1. Use of declased build servic [9] 2.6 4.0 -1. 3.3 3.0 -1. Use of declased build servic [9] 2.6 3.0 -1. 3.2 3.0 -1.	Ephemenal build environment [9] 3.6 3.0 4 2.9 3.0 4. Prevent script execution 3.7 3.0 4.0 3.0 3.0 4.0 3.0 3.0 4.0 3.0 3.0 4.0 3.0 4.0 4.0 3.0 3.0 4.0 4.0 3.0 3.0 4.0 4.0 3.0 3.0 4.0 4.0 3.0 3.0 4.0 4.0 3.0 3.0 4.0 4.0 4.0 3.0 3.0 4.0 4.0 3.0 3.0 4.0 4.0 3.0 4.0 </td <td>Epbeneral build environment (9) 3.6 3.0 4 2.9 3.0 Prevent script execution 3.7 3.0 4 3.0<td>Ephemeral build environment [9] 3.6 3.0 4.2 2.9 3.0 4.3 Prevent script execution 3.7 3.0 4.0 3.0 5.0 4.0 Phild/Mage regent review 4.6 5.0 4.8 4.0 4.0 4.0 3.3 3.0 4.0 Code signing 3.7 4.0 4.0 3.3 3.0 4.1 Integrate Open-Source vulnerability scamer into CICD pipeline 3.6 4.0 4.1 4.0 4.1 4.0 Use of doctacted build stropt (5] 3.3 3.0 4.1 4.0 4.1 4.0 Reproducible build stropt (5] 3.2 3.0 4.1 4.0 4.1 4.0</td><td>Ephemeral build environment (P) 3.6 3.0 4.2 2.9 3.0 4.1 Pervents script execution 3.7 3.0 4.0 3.0 3.0 4.0 PullMarge request review executed during each build steps [51] the of dedicated build service [9] 3.7 4.0 4.0 3.3 3.0 4.0 Lise of dedicated build service [9] 3.6 4.0 -1 3.3 3.0 4.0 Preventive scatting the revise [9] 3.6 4.0 -1 3.3 3.0 4.1 Preventive scatting the release plackages Audit, security assessment, vulnerability scatter 3.3 4.0 -1 3.3 3.0 -1 Reproducible builds 4.2 5.0 -1 4.1 4.0 -1 Reproducible builds 4.2 5.0 -1 4.1 4.0 -1</td><td></td><td>Maintain detailed SBOM [8] and perf</td><td>orm</td><td>4.2</td><td>5.0</td><td>- 10</td><td>3.4</td></td>	Epbeneral build environment (9) 3.6 3.0 4 2.9 3.0 Prevent script execution 3.7 3.0 4 3.0 <td>Ephemeral build environment [9] 3.6 3.0 4.2 2.9 3.0 4.3 Prevent script execution 3.7 3.0 4.0 3.0 5.0 4.0 Phild/Mage regent review 4.6 5.0 4.8 4.0 4.0 4.0 3.3 3.0 4.0 Code signing 3.7 4.0 4.0 3.3 3.0 4.1 Integrate Open-Source vulnerability scamer into CICD pipeline 3.6 4.0 4.1 4.0 4.1 4.0 Use of doctacted build stropt (5] 3.3 3.0 4.1 4.0 4.1 4.0 Reproducible build stropt (5] 3.2 3.0 4.1 4.0 4.1 4.0</td> <td>Ephemeral build environment (P) 3.6 3.0 4.2 2.9 3.0 4.1 Pervents script execution 3.7 3.0 4.0 3.0 3.0 4.0 PullMarge request review executed during each build steps [51] the of dedicated build service [9] 3.7 4.0 4.0 3.3 3.0 4.0 Lise of dedicated build service [9] 3.6 4.0 -1 3.3 3.0 4.0 Preventive scatting the revise [9] 3.6 4.0 -1 3.3 3.0 4.1 Preventive scatting the release plackages Audit, security assessment, vulnerability scatter 3.3 4.0 -1 3.3 3.0 -1 Reproducible builds 4.2 5.0 -1 4.1 4.0 -1 Reproducible builds 4.2 5.0 -1 4.1 4.0 -1</td> <td></td> <td>Maintain detailed SBOM [8] and perf</td> <td>orm</td> <td>4.2</td> <td>5.0</td> <td>- 10</td> <td>3.4</td>	Ephemeral build environment [9] 3.6 3.0 4.2 2.9 3.0 4.3 Prevent script execution 3.7 3.0 4.0 3.0 5.0 4.0 Phild/Mage regent review 4.6 5.0 4.8 4.0 4.0 4.0 3.3 3.0 4.0 Code signing 3.7 4.0 4.0 3.3 3.0 4.1 Integrate Open-Source vulnerability scamer into CICD pipeline 3.6 4.0 4.1 4.0 4.1 4.0 Use of doctacted build stropt (5] 3.3 3.0 4.1 4.0 4.1 4.0 Reproducible build stropt (5] 3.2 3.0 4.1 4.0 4.1 4.0	Ephemeral build environment (P) 3.6 3.0 4.2 2.9 3.0 4.1 Pervents script execution 3.7 3.0 4.0 3.0 3.0 4.0 PullMarge request review executed during each build steps [51] the of dedicated build service [9] 3.7 4.0 4.0 3.3 3.0 4.0 Lise of dedicated build service [9] 3.6 4.0 -1 3.3 3.0 4.0 Preventive scatting the revise [9] 3.6 4.0 -1 3.3 3.0 4.1 Preventive scatting the release plackages Audit, security assessment, vulnerability scatter 3.3 4.0 -1 3.3 3.0 -1 Reproducible builds 4.2 5.0 -1 4.1 4.0 -1 Reproducible builds 4.2 5.0 -1 4.1 4.0 -1		Maintain detailed SBOM [8] and perf	orm	4.2	5.0	- 10	3.4
Prevent script execution 3.7 3.0 1.0 3.0 3.0 1. 1. Pull/Marge request review 4.6 5.0 3.8 3.0 1. 1. Restrict access to system resources of code 4.0 4.0 4.0 3.3 3.0 1. 1. Code signing 3.7 4.0 4.0 3.1 3.0 1. 1. Integrate Open-Source vibrenbility scamer 3.8 4.0 - 3.3 3.0 1. 1. Use of scheme Use of scheme vibrenbility scamer 3.8 4.0 - 3.3 3.0 1. 1. Use of scheme built service [9] 3.6 4.0 1.2 3.0 1.0 1. Preventive separating the released packages 3.3 3.0 1.0 1.0 1.0 1.0	Prevent script execution 3.7 3.0 4.0 3.0 3.0 -1. 1. PulluMcgree project rociwe 4.6 5.0 -1. 3.3 3.0 -1. 1. Restrict access to system resources of code signing 4.0 4.0 4.0 -1. 3.3 3.0 -1. 1. Code signing 3.7 4.0 -1. 3.3 3.0 -1. 1. Integrate Open-Source outerability scamer 3.8 4.0 -1. 3.3 3.0 -1. 1. Preventive squating the released Piceback 3.1 3.6 -1. 3.3 3.0 -1. 1. Preventive squating the released Piceback 3.6 0. -1. 3.3 3.0 -1. 1. Sestemet, Predictive strike (Piceback 3.6 0. -1. 3.3 3.0 -1. 1. Sestemet, Predictive strike (Piceback 3.6 0. -1. 3.3 3.0 -1. 1. Sestemet, Predictive strike (Piceback 3.6 0. -1. 3.4. 0. -1. 1.	Prevent script execution 3.7 30 50 5.8 4.1 1. Restrict access to system resources of code executed during each build steps [5] 4.0 4.0 4.0 3.8 4.0 1.1 Code signing 3.7 4.0 4.0 3.3 3.0 4.1 1. Integrate Open-Source vulnerability scamer into CICD pipeline 3.7 4.0 4.0 3.3 3.0 4.1 1. Ver of dedicated builto revice [9] 3.6 4.0 4.0 3.3 3.0 4.1 1. Net of dedicated builto revice [9] 3.6 3.0 4.1 4.0 1.1 Restrict accell parameter the model packages accenter prevention the string builty accenter packages 3.3 3.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1 4.0 4.1	Prevent script execution 3.7 3.0 1.0 Restrict access to system resource of code executed during each build steps [S1] 3.7 4.0 4.0 3.1 3.0 -1 1. Integrate Open-Source vulnerability scamer into CICD pipeline 3.6 4.0 -1 3.3 3.0 -1 1. Use of dotacide build steps [S1] 3.1 3.0 -1 3.3 3.0 -1 1. Preventive squating the released packages Assessment, preservice [9] 3.6 4.0 -1 1. 1. 1. Resproducible builds 5.1 3.1 3.0 -1 1. 1. 1. 1. Sessment, prestration testing -1 -1 -1 1. 1. 1. 1. 1. 1.	Prevent stript execution 3.7 3.0 1.0 3.0 3.0 1.1 Pull/deep repeat review 4.6 5.0 3.3 3.0 1.0 1.0 Retrict acces to symme resources of order each build steps [S1] 0. 4.0 4.0 3.3 3.0 1.0 1.0 Code signing cash build steps [S1] 3.7 4.0 4.0 3.1 3.0 1.0 1.0 Integrate Open-Source vulnerability scamer 3.8 4.0 -10 3.3 3.0 -10 1.0 Use of deficient buils service [9] 3.6 4.0 -10 3.3 3.0 -10 1.0 Auti, scenarity insessment, indernethity as- sessment, pertertation testing 3.3 3.0 -11 1.1 Reproducible builds 4.2 5.0 -11 1.1 1.1 1.1 Ionlarge of build steps [S1] 3.3 3.0 -11 1.1 1.1 1.1 Lise of deficient build steps [S1] 3.3 3.0 -11 1.1 1.1 1.1 Auti, scenarity 1.3 3.0 -11 1.1		Ephemeral build environment [9]		3.6	3.0	also,	2.9
Pall/Merge request review 4.6 5.0 J 3.8 4.0 J Restrict access to system resources of code executed during each build steps [51] 4.0 4.0 J 3.3 3.0 JL JL Code signing 3.7 4.0 JL 3.1 3.0 JL JL Integrate Open-Source vulnerability scanner 3.8 4.0 JL 3.1 3.0 JL JL Use of dedicated build service [9] 3.1 3.0 JL 2.9 3.0 JL 2.9 3.0 JL 2.9 3.0 JL 3.0	PullMarge request review 4.6 5.0 J 3.8 4.0 J Restrict access to system resources of code executed during each build steps [51] 4.0 4.0 J 3.3 3.0 J J Code signing 3.7 4.0 J 3.1 3.0 J <td< td=""><td>Pall/Marge request review 4.6 5.0 3.8 4.0 1 Retrict access to system resources of code cascated during each build steps [S1] 3.7 4.0 3.8 3.0 1 Code signing 3.7 4.0 3.8 3.0 1 1 Integrate Open-Source vulnerability scamer into CLOP pipeline 3.6 4.0 3.3 3.0 1 1 View of declared built service [9] 3.6 4.0 2.9 3.0 1 1 Audit, accerts assumet, vulnerability sc 4.3 4.0</td><td>Pull/Merge request review 46 50 a 38 40 Restrict access to system resources of code executed during each build steps [5] 40 40 a 31 30 a Code signing 37 40 b 31 30 b 1 Integrate Open-Source vulnerability scamer 38 40 b 31 30 b 1 Use of deficiend build steps [6] 36 40 b 31 30 b 1 Use of deficiend build steps [6] 36 40 b 33 30 b 1 Verventive squatting the released packages Audi, security assessmert, ulternethity as- sessmert, penetration testing 33 30 b 41 40 a Reproducible builds 10 33 30 b 31 30 b</td><td>Pull/Marge request review 4.6 5.0 3.8 4.0 1.1 Retrict acces to system resources of ode executed during each build steps [5] 3.7 4.0 3.8 3.0 1.1 Code signing 3.7 4.0 3.1 3.0 1.1 1.1 Integrate Open Source vinterability scanner 3.8 4.0 -1 3.3 3.0 1.1 1.1 Integrate Open Source vinterability scanner 3.8 4.0 -1 3.3 3.0 1.1 1.1 Preventive squatting the released packages 3.3 3.0 1.1 2.0 3.0 1.1 1.1 Preventive squatting the released packages 4.3 4.0 1.4 4.0 1.1 1.1 Reproducible builds 4.2 5.0 1.4 4.0 1.1 1.1 1.1 1.1 1.1 1.1 2.9 3.0 1.1 2.0 3.0 1.1 1.1 2.0 3.0 1.1 2.0 3.0 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1</td><td></td><td>Prevent script execution</td><td></td><td>3.7</td><td>3.0</td><td>12</td><td>3.0</td></td<>	Pall/Marge request review 4.6 5.0 3.8 4.0 1 Retrict access to system resources of code cascated during each build steps [S1] 3.7 4.0 3.8 3.0 1 Code signing 3.7 4.0 3.8 3.0 1 1 Integrate Open-Source vulnerability scamer into CLOP pipeline 3.6 4.0 3.3 3.0 1 1 View of declared built service [9] 3.6 4.0 2.9 3.0 1 1 Audit, accerts assumet, vulnerability sc 4.3 4.0	Pull/Merge request review 46 50 a 38 40 Restrict access to system resources of code executed during each build steps [5] 40 40 a 31 30 a Code signing 37 40 b 31 30 b 1 Integrate Open-Source vulnerability scamer 38 40 b 31 30 b 1 Use of deficiend build steps [6] 36 40 b 31 30 b 1 Use of deficiend build steps [6] 36 40 b 33 30 b 1 Verventive squatting the released packages Audi, security assessmert, ulternethity as- sessmert, penetration testing 33 30 b 41 40 a Reproducible builds 10 33 30 b 31 30 b	Pull/Marge request review 4.6 5.0 3.8 4.0 1.1 Retrict acces to system resources of ode executed during each build steps [5] 3.7 4.0 3.8 3.0 1.1 Code signing 3.7 4.0 3.1 3.0 1.1 1.1 Integrate Open Source vinterability scanner 3.8 4.0 -1 3.3 3.0 1.1 1.1 Integrate Open Source vinterability scanner 3.8 4.0 -1 3.3 3.0 1.1 1.1 Preventive squatting the released packages 3.3 3.0 1.1 2.0 3.0 1.1 1.1 Preventive squatting the released packages 4.3 4.0 1.4 4.0 1.1 1.1 Reproducible builds 4.2 5.0 1.4 4.0 1.1 1.1 1.1 1.1 1.1 1.1 2.9 3.0 1.1 2.0 3.0 1.1 1.1 2.0 3.0 1.1 2.0 3.0 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1		Prevent script execution		3.7	3.0	12	3.0
ceccuted during each build steps [5] 57 4.0 -Im 3.1 3.0 -Im Code signing 5.7 4.0 -Im 3.3 3.0 -Im Integrate Open-Survey valuerability scame 3.8 4.0 - Im 3.3 3.0 -Im Use of dedicated build service 3.6 4.0 - Im 3.3 3.0 -Im Use of dedicated build service 3.5 3.0 - Im 2.0 3.0 - Im	Code signing 3.7 4.0 Jm 3.1 3.0 Jm Integrate Open-Source values/billy scamer 3.8 4.0 - m 3.3 3.0 Jm Integrate Open-Source values/billy scamer 3.8 4.0 - m 3.3 3.0 Jm Preventive squares/billy scamer 3.6 4.0 Jm 3.3 3.0 Jm Preventive squares/bills the relased projectages 33 3.0 Jm	Code signing 3.7 4.0 3.1 3.0 Integrate Oper-Source vulnerability scaner 3.8 4.0 - 3.3 3.0 Integrate Oper-Source vulnerability scaner 3.8 4.0 - 3.3 3.0 Use of dedicated build serve [0] 3.6 4.0 - 3.3 3.0 Preventive squaring the released point 3.1 3.0 - - 3.3 3.0 Sesment, phetration testing 3.1 3.0 - - 3.3 3.0 Preventive squaring the released point 3.1 3.0 - - 3.3 3.0 Brownburdte builds 4.1 4.0 - - - 3.3 3.0	Code signing 3.7 4.0 3.3 3.0 Integrate Open-Source vulnerability scamer 3.8 4.0 - 3.3 3.0 Integrate Open-Source vulnerability scamer 3.8 4.0 - 3.3 3.0 - Preventive separating the released packages 3.4 4.0 - 3.3 3.0 - Reproducible builds 5.1 3.0 - 4.1 4.0 -	Code signing 3.7 4.0 3.3 3.0 Integrate Open-Source vulnerability scame 3.8 4.0 - 3.3 3.0 Use of dedicated build service (7) 3.6 4.0 - 3.3 3.0 Presentive squaring the released packages 3.6 4.0 - 3.3 3.0 Reproducible build 4.3 4.0 - 4.1 4.0 Scooped paces [21] (74] 2.9 3.0 - -		Pull/Merge request review	ode	4.6	5.0	- 12	3.8
Code signing 3.7 4.0 Jm 3.1 3.0 .L. Integrate Open-Source vulnerability scamer 3.8 4.0 - Jm 3.3 3.0 - JL Use of dedicated built service (*) 3.6 4.0 - Jm 3.3 3.0 - JL Use of dedicated built service (*) 3.6 3.8 3.0 - JL 3.0 - JL Presentive quanting the released packages 3.8 3.0 - JL 2.9 3.0 - JL	Code signing 3.7 4.0 Jm 3.1 3.0 Jm Integrate Open-Source voltenbility scamer 3.8 4.0 - 3.3 3.0 Jm Integrate Open-Source voltenbility scamer 3.8 4.0 - 3.3 3.0 Jm Use of decisated built service [9] 3.6 4.0 Jm 3.3 3.0 Jm Preventive squaring the released prevention segurition testing 3.1 3.4 4.0 Jm 4.1 4.0 Jm	Code signing 3.7 4.0 3.1 3.0 1 Integrate Oper-Source vulnerability scamer into CUCD pipeline Use of dedicated build service [9] 3.6 4.0 - 3.3 3.0 - - 3.3 3.0 - - 3.3 3.0 - - 3.3 3.0 - - - 3.3 3.0 - - - 3.3 3.0 - - - 3.3 3.0 - - - 3.3 3.0 - - - 3.3 3.0 - - - 3.3 3.0 - - - 3.3 3.0 - - - 3.3 3.0 - - - 3.3 3.0 - - - 3.3 3.0 - - - 3.3 3.0 - - - - 3.0 - - - - 3.0 - - - - - - - <t< td=""><td>Code signing 3.7 4.0 Jan 3.1 3.0 Jan Integrate Open-Source vulnerability scamer 3.8 4.0 - 3.3 3.0 - Jan Use of dedicated build service [9] 3.6 4.0 - 3.3 3.0 - Jan Jan</td><td>Code signing 3.7 4.0 Jm 3.1 3.0 Jm Integrate Open Source vulnerability scamer into CLCD pipeline Use of dedicated build service (i) 3.8 4.0 Jm 3.3 3.0 Jm Use of dedicated build service (i) 3.6 4.0 Jm 3.3 3.0 Jm Preventive summing the released packages Audit, scentry assessment, vulnerability as- sessment, packages 3.0 Jm 3.0 Jm 3.0 Jm Reproducible build Scoled autores [51] 2.9 3.0 Jm 3.1 4.0 Jm Scoled autores [51] 2.9 3.0 Jm 3.1 4.1 4.0 Jm Scoled autores [51] 2.9 3.0 Jm 3.0 Jm Jm 3.0 Jm</td><td></td><td>executed during each build steps [51]</td><td>oue</td><td>4.0</td><td>4.0</td><td></td><td>3.3</td></t<>	Code signing 3.7 4.0 Jan 3.1 3.0 Jan Integrate Open-Source vulnerability scamer 3.8 4.0 - 3.3 3.0 - Jan Use of dedicated build service [9] 3.6 4.0 - 3.3 3.0 - Jan	Code signing 3.7 4.0 Jm 3.1 3.0 Jm Integrate Open Source vulnerability scamer into CLCD pipeline Use of dedicated build service (i) 3.8 4.0 Jm 3.3 3.0 Jm Use of dedicated build service (i) 3.6 4.0 Jm 3.3 3.0 Jm Preventive summing the released packages Audit, scentry assessment, vulnerability as- sessment, packages 3.0 Jm 3.0 Jm 3.0 Jm Reproducible build Scoled autores [51] 2.9 3.0 Jm 3.1 4.0 Jm Scoled autores [51] 2.9 3.0 Jm 3.1 4.1 4.0 Jm Scoled autores [51] 2.9 3.0 Jm 3.0 Jm Jm 3.0 Jm		executed during each build steps [51]	oue	4.0	4.0		3.3
Integrate Open-Source vulnerability scanner 3.8 4.0 - In 3.3 3.0 In into CI/CD pipeline Use of adelarate built service [9] 3.6 4.0 In 3.3 3.0 In Preventive squatting the released packages 3.1 3.0 In 4.2 9 3.0 In	Integrate Open-Score: voltamethility scanner 3.8 4.0 - - 3.3 3.0 - - - 1.0 <	Integrate Open-Source volume/hilly scamer 3.8 4.0 - 3.3 3.0 Use of declared build service [9] 3.6 4.0 3.3 3.0 Preventive squaring the released parts 3.1 3.6 .0 3.3 3.0 Audit, scenitry assessment, undershilly assessment, preteration testing 3.1 3.40 4.1 4.0 Browdwitche builds 4.2 5.0 4.1 4.0	Integrate Open-Source whereholding scatter 3.8 4.0 - 3.3 3.0 - Use of definition that wrise [P] 3.6 4.0 - 3.3 3.0 - - - 3.3 3.0 - - - 3.3 3.0 - - - 3.3 3.0 - - - 3.3 3.0 - - - 3.3 3.0 - - - - 3.3 3.0 - - - - - 3.3 3.0 -	Integrate Open-Source vulnerability scamer into CPCD pipeline 3.8 4.0 - 3.3 3.0 - Use of decisard built service [9] 3.6 4.0 - 3.3 3.0 - - 3.3 3.0 - - - 3.3 3.0 - - - 3.3 3.0 - - - 3.3 3.0 - - - - 3.3 3.0 - - - 3.3 3.0 - - - - 3.3 3.0 - - - 3.3 3.0 - - - 3.3 3.0 - - - - 3.3 3.0 - - - - 3.0 -		Code signing		3.7	4.0	.lte	3.1
into CUCD pipeline Use of deciated build service [9] 3.6 4.0 3.3 3.0 1.09 Preventive squatting the released packages 3.1 3.0 2.9 3.0 1.07 Audit complementation through the service of the	into CLCD pipeline Use of deficient buils service [9] 3.6 4.0 1.3 33 50 1. 1.09 Prevenitive spating the released packages 3.1 50 1.2 2.9 30 1. 107 Audit, security assessment, vulnerability as- sessment, protentiation testing	initio CROP pipeline 5.5 4.0 5.3 3.0 1.99 Use of declarated bid service (?) 5.5 4.0 2.9 3.0 1.99 Presentive squaring the released packages 33 3.0 1.09 2.9 3.0 1.01 Assesses and protection to call with released packages 3.4 4.0 1.4 1.05 Resented protection builds 4.3 4.0 4.1 4.0 1.02	into CLCD piceline 3.6 4.0 3.3 3.0 1.09 Diversitie squating the released packages 3.3 3.0 2.9 3.0 1.07 Addit, scenity assessment, vientenbility as- sessment, penetration isolitie 4.3 4.0 4.1 4.0 109 Reproducible builds 4.2 5.0 4.1 4.0 100 Ioadian of builds steps [5] 3.1 3.0 4.1 4.0 100	into CLCD piceline 3.6 4.0 3.3 5.0 1.09 Prevenite squating the released packages 3.3 3.0 2.9 5.0 1.07 Addit, security assessment, valuerability assessment, preteriation resting 4.3 4.0 4.1 4.0 1.08 Reproducible builds or p(51) 3.1 3.0 4.1 4.0 1.08 Iobation of builds or p(51) 3.1 3.0 4.1 4.0 1.08 Scoped packages [72], [72] 2.9 3.0 4.3 3.0 4.1 1.00		Integrate Open-Source vulnerability sca	nner	3.8	4.0	- 80	a 3.3
Preventive squatting the released packages 3.1 3.0 2.9 3.0 1.07	Preventive squatting the released packages 31 3.0 L 2.9 3.0 L 1.07 Audit, security assessment, vulnerability as- sessment, penetration testing	Preventive squating the released packages 3.1 3.0 L 2.9 3.0	Preventive separating the released packages 34 30 29 30 107 Audit, security assessment, ubreatrability as- sessment, prestration testing 4.3 4.0 1.07 Reproducible builds 4.2 5.0 4.1 4.0 1.02 Isolation of build seeps (51) 331 3.0 3.0 1.02	Preventive squatting the released packages 33 30 42 29 30 41 107 Audit, scentri, varia, sessment, uplenzbility as- sesment, pentration testing 43 40 41 40 105 Reproducible builds Isolation of build steps [51] 33 30 41 40 100 Scoped packages [72], [74] 29 30 41 100 100		into CI/CD pipeline Use of dedicated build service [9]	- 1	3.6	4.0	.104	3.3
	Addit, security assessment, vulnerability as- 4.5 4.0 4.1 4.0 1.05 sessment, penetration testing	Audit, security assessment, vulnerationity as- sessment, pretration resting	Audit, lecurity assessment, penetration testing 4.3 4.0 4.1 4.0 1.05 sessment, penetration testing Reproducible builds 4.2 5.0 4.1 4.0 4.1 1.02 Isolation of build seps [51] 3.1 3.0 4.3 3.0 4.1 1.02	Audit, locarity assessment, vimerationy as- 4.3 4.0 1.03 assessment, productible builds 42 5.0 1.41 4.0 1.03 Reproductible builds 42 5.0 1.41 4.0 1.03 Jobatian of build steps (5) 2.3 3.0 1.03 1.03 Scouge Daylese (72) (74) 2.9 3.0 1.03 1.04		Preventive squatting the released packa	ges	3.1	3.0	alle i	2.9
Reproducible builds 4.2 5.0 J 4.1 4.0 J 1 Isolation of build steps [51] 3.1 3.0 J 3.1 3.0 J 1 <td>Isolation of build steps [51] 2,1 (31 30 4 1 30 4 1 1 Scoped packages [72], [74] 2,9 3,0 4 2,9 3,0 4 1 1 Establish internal repository multiple [72] 4 3,0 4 3,7 4,0 4 0 0</td> <td>Scoped packages [72], [74] 2.9 3.0 5 2.9 3.0 5 1. Establish internal repository mirrors and ref- erence one private feed, not multiple [72]</td> <td>Establish internal repository mirrors and ref- erence one private feed, not multiple [72] 3.7 4.0 - 0.</td> <td></td> <td></td> <td>erence one private feed, not multiple [7</td> <td></td> <td>4.1</td> <td>4.0</td> <td>14</td> <td>4.3</td>	Isolation of build steps [51] 2,1 (31 30 4 1 30 4 1 1 Scoped packages [72], [74] 2,9 3,0 4 2,9 3,0 4 1 1 Establish internal repository multiple [72] 4 3,0 4 3,7 4,0 4 0 0	Scoped packages [72], [74] 2.9 3.0 5 2.9 3.0 5 1. Establish internal repository mirrors and ref- erence one private feed, not multiple [72]	Establish internal repository mirrors and ref- erence one private feed, not multiple [72] 3.7 4.0 - 0.			erence one private feed, not multiple [7		4.1	4.0	14	4.3
Reproducible builds 4.2 5.0	Induiting of Point 31 30 31 30	Scoped packages [72], [74] 29 30 29 30 10 Establish internal repository mirrors and ref- erence one private feed, not multiple [72] 30 29 30 10 02 Application Scenarity Testing 4.1 4.0 4.3 5.0 02	Establish internal repository mirrors and ref- erence one private feed, not multiple [72] Application Security Testing 4.1 4.0 H 4.3 5.0 - 00	Application Security Testing 4.1 4.0 🖬 4.3 5.0 0.		erence one private feed, not multiple [Application Security Testing			4.0	- 10	4.3
Reproducible builds 4.2 5.0 4.1 4.0 4.3 5.0 4.1 4.0 4.3 5.0 4.1 4.0 4.3 5.0 4.1	Institution of build steps [51] 3.1 3.0 1.4 3.1 3.0 1.0 Scoped packages [72], [74] 3.1 3.0 1.0 1.0 1.00 Batabilis interrel resolutory mirrors and ref- erence one private feed, nor multiple [72] 3.0 1.0 1.0 1.00 Application Security Testing 4.1 4.0 1.0 0.97 0.97 Enablishy situte provess for Open-Source 4.1 4.0 1.0 0.97	Scoped packages [72], [74] 20 50 - 29 50 - 29 50 - 100 Establish interrupt repository mirrors and ref- erence one private feed, nor multiple [72] Application Security Testing 4.1 4.0 4.3 5.0 - 0 99 Establish wirring mores for Done-Source 4.1 4.0 4.3 5.0 - 0 99	Establishi metana ferçosi yan maros nad ref- Application Security Testing 4.1 4.0 4.3 5.0 4.0 995 Fatablishi vertem provase for Open-Source 4.1 4.0 4.3 5.0 4.0 995	Application Security Testing 4.1 4.0 4.3 5.0 - 0.95 Feablish vetting process for Open-Source 4.1 4.0 4.3 5.0 - 0.95		erence one private feed, not multiple (Application Security Testing Establish vetting process for Open-So	urce	4.1			
Reproducible builds 4.2 5.0 1.4.1 4.0	Isolation of build streps [51] 31 30 1 31 30 1 100 Scoped package (72), [72] 9 30 1 31 30 1 100 Establish internal repository mirrors and ref- erence one private feed, nor multiple [72] 36 30 1 37 4.0 0 97 Application Security Testing 4.1 4.0 1 4.3 5.0 0.95 Establish verting process for Open-Source components bound in internal/public repos- components bound in internal/public repos- 4.1 4.0 1 4.3 5.0 0 95	Scored packages [72], [74] Establish internal propository mirrors and ref- rence one private feed, not multiple [72] Application Security Testing Establish vetting process for Open-Source Establish vetting process for Open-Source Description of the security and the repos- tion of the security Testing the repose- tion of the security Testing the security th	Establish internal repository mirrors and ref. 36 30 Jan 37, 40 4 097 errece one private 6ed, our multiple [22] 36 30 Jan 37, 40 4 097 Application Security Texting 4.1 40 Jan 4.3 5.0 Jan 095 Establish witing process for Open-Source components hose in internal physics repose. 14 4.0 Jan 4.3 5.0 Jan 095	Application Security Testing 4.1 4.0 H 4.3 5.0 J 0.95 Establish vetting process for Open-Source 4.1 4.0 H 4.3 5.0 J 0.95 intervals	г	erence one private feed, not multiple [Application Security Testing Establish vetting process for Open-So components hosted in internal/public re-	urce pos-	4.1	4.0		
Reproducible builds 4.2 5.0 4.1 4.0 4.1 1.02 Isolation of build step [\$1] 3.1 3.0 3.1 3.0 3.1 3.0 1.02 Scoped package [\$2] 1.01 2.0 3.0 3.1 3.0 1.02 1.00 Establish internal repository mirrors and ref. 3.6 3.0 3.7 4.0 3.6 0.97 Application Security Testing 4.1 4.0 4.3 5.0 3.0 9.5 Establish vetting process for Open-Source components based in internal/public repose tories 4.1 4.0 4.3 5.0 3.0 9.5	Isolation of build steps [51] 3.1 3.0 3.1 3.0 3.1 3.0 Scoped packages [72], [72] 2.9 3.0 3.0 2.9 3.0 1.00 Estabilish internal process printers and reference one private feed, not multiple [72] 3.6 3.0 3.7 4.0 4.0 9.95 Application Security Testing 4.1 4.0 4.3 5.0 -4 0.95 Estabilish vetting process for Open-Source components hosted in internal/public repositories 4.1 4.0 4.3 5.0 -4 0.95	Scored packages [72], [74] 29 30 29 30 1.00 Exability interrupt regointy mirror and ref- erence one private feed, not multiple [72] 36 30 14 3.7 4.0 9.97 Application Security Testing 4.1 4.0 14 4.3 5.0 30 9.95 Establish vetting process for Open-Source components hosted in internal/public repos- tiones 4.1 4.0 4.3 5.0 30 9.95	Establish internal repository mirrors and ref- rence one private ford, our multiple [2] Application Security Testing Establish vetting process for Open-Source components hosted in internal/public repos- itories	Application Security Testing 4.1 4.0 4.4.3 5.0 4.0 0.95 Exclusion verting process for Open-Source 4.1 4.0 4.4.3 5.0 4.0 0.95 itories		erence one private feed, not multiple [Application Security Testing Establish vetting process for Open-So components hosted in internal/public re- itories	urce pos-	4.1	4.0		
Reproducible builds 4.2 5.0 1 4.1 4.0 1.02 Isolation of build steps [51] 3.3 3.0 3.3 3.0 3.3 3.0 1.02 Scored ackage [72] [74] 3.0 3.0 3.0 3.0 1.02 Establish instremal repository mimors and ref- erence one private feed, nor multiple [72] 3.6 3.0 Ja 3.7 4.0 4.0 9.97 Application Security Testing 4.1 4.0 Ja 5.0 Ja 0.95 Establish vetting process for Open-Source components hoteld in internalipable repos- itories 3.9 4.0 Ja 5.0 Ja 0.95 Code isolation and sandboxing 3.9 4.0 Ja 5.0 Ja 0.95	Industries of biald steps [51] 3.1 3.0 4.1 3.0 4.1 3.0 4.0 1.00 Scoped packages [72]. [72] 2.00 3.0 4.0 3.0 4.0 4.0 1.00 Establish internal repository mirrors and ref- remese one private feed, nor mariple [72] 3.0 4.0 4.1 3.0 4.0 0.07 Application Security Testing 4.1 4.0 4.3 5.0 -4 0.95 Establish verting process for Open-Source components hosted in internal/public repos- intorie 4.1 4.0 4.3 5.0 -4 0.95 Code isolation and santboxing 3.9 4.0 -4 4.2 4.0 -4 0.93	Scored packages [72], [74] 29 30 29 30 40 100 Exability intermal repository mirrors and repository mirrors and repository mirror and repository mir	Establish internal repository mirrors and ref- errence one private feed, not multiple [72] Application Security Testing Establish verting process for Open-Searce components hosted in internal/public repos- tioners Code isolation and sandboxing 39 4.0 4 4.3 5.0 4 0.95	Application Security Testing 4.1 4.0 4.3 5.0 -0.95 Exability wetting process for Open-Source tories 4.1 4.0 -4 4.3 5.0 -10 0.95 components hosted in internal/public reposition: 5.0 -10 0.95 -10 0.95 Code isolation and sandboxing 3.9 4.0 -10 4.2 4.0 -10 0.93	_	erence one private feed, not multiple (Application Security Testing Establish vetting process for Open-So components hosted in internal/public re itories Code isolation and sandboxing	urce pos-	4.1 3.9	4.0		1 4.2
Reproducible builds 4.2 5.0 1 4.1 4.0 10.0 Isolation of build steps [51] 3.1 3.0 3.1 3.0 10.0 Scoopd package [72] [74] 2.0 3.0 10.0 10.0 Establish internal repository mirror and ref- cence one private ford, not multiple [72] 3.6 3.0 1.0 3.7 4.0 4.0 4.3 5.0 -0.97 Application Security Testing 4.1 4.0 4.3 5.0 -1 0.97 Establish vetting process for Open-Source components hosted in internal/public reposi- intories 4.1 4.0 4.3 5.0 -1 0.97 Code isolation and sandboxing 3.9 4.0 -1 4.2 4.0 -1 0.8 Runtime Application Self-Protection 3.7 4.0 -1 4.2 4.0 -1 0.8	Indication of bald steps [51] 33 30 10 31 30 10 Scoped packages [72], [74] 29 30 10 29 30 10 10 Establish internal prository mirrors and ref- erence one private feed, not multiple [72] 63 30 10 33 50 10 09 Application Scurity Testing 4.1 4.0 10 4.3 5.0 10 0.9 Establish internal/public reposi- tiories 10 4.1 4.0 4.3 5.0 10 0.9 Code isolation and sandboxing 3.9 4.0 4.3 5.0 10 0.9 Runtime Application Scif-Protection 3.7 4.0 4.2 4.0 4.9	Scored pickages [72], [74] 20 30 10 29 30 10 Exabilish interrupt repository mirrors and ref- erence one private feed, not multiple [72] 36 30 14 3.7 4.0 99 Application Scority Testing 4.1 4.0 14 4.3 5.0 0.92 Exabilish interrupt process for Operators 4.1 4.0 14 4.3 5.0 0.92 Exabilish interrupt process for Operators 4.1 4.0 14 4.3 5.0 0.92 Exabilish interrupt process for Operators 4.1 4.0 14 4.3 5.0 0.92 Code isolation and sandboxing 3.9 4.0 4.2 4.0 0.93 Runtime Application Scif-Protection 3.7 4.0 4.2 4.0 0.93	Establish internal repository mirrors and ref. 36 30 Jan 3.7 4.0 40 097 erence one private feed, not multiple [72] Application Security Testing Establish verting process for Open-Source components hosted in internal/public repos- tiones Code isolation and sandboxing 3.9 4.0 442 4.0 40 0.97 Runtime Application Self-Protection 3.7 4.0 444 4.2 4.0 41 0.88	Application Security Testing 4.1 4.0 4.3 5.0 -0.92 Establish vetting process for Open-Source components hosted in internal/public repositories 4.1 4.0 4.3 5.0 -0.92 Code isolation and sandboxing 3.9 4.0 -4 4.2 4.0 -4 0.92 Runtime Application Self-Protection 3.7 4.0 -4 4.2 4.0 -4	-[erence one private feed, not multiple [Application Security Testing Establish vetting process for Open-So components hosted in internal/public re itories Code isolation and sandboxing Runtime Application Self-Protection	urce pos-	4.1 3.9 3.7	4.0 4.0	.etc	I 4.2 I 4.2
Reproducible builds 4.2 5.0 4.1 4.0 1 Isolation of build steps [51] 2.0 3.0 2.0 2.0 1 Scoped package [22], [72] 3.0 3.0 3.0 3.0 3.0 3.0 1 Exability internal repository mirrors and ref. 3.6 3.0 3.0 3.7 4.0 4.0 Application Security Testing 4.1 4.0 4.3 5.0 3.0 Exability internal reposes for Open-Source repose roots based in internal/public repose roots based in internal/public repose roots based in internal/public repose roots and sandboxing 3.9 4.0 4.2 4.0 3.0 Code isolation and sandboxing 3.7 4.0 3.0 4.2 4.0 3.0 Mamile source code review 4.1 4.0 3.8 5.0 3.0	Installation of build steps [5] 33 30 <td>Scored packages [22, [74] 22 30 J 22 30 J 24 37 4.0 J 26 50 J 27 50 J 26 50 J</td> <td>Establish internal repository mirrors and ref- rence one private feed, nor milliple [2] 3.6 3.0 Jan 3.7 4.0 Jan 0 Application Security Testing 4.1 4.0 Jan 4.3 5.0 Jan 0 Establish vetting process for Open-Source components hold in internal/public repos- tories block in and sandboxing 3.9 4.0 Jan 4.2 4.0 Jan 0 Runtime Application Self Protection 3.7 4.0 Jan 4.2 4.0 Jan 0 Manual source code review 4.1 4.0 Jan 4.8 5.0 Jan 0</td> <td>Application Security Testing 4.1 4.0 4.3 5.0 0 Establish vetting process for Open-Source components hosted in internal/public repositions 4.1 4.0 4.3 5.0 0 Code isolation and sandboxing 3.9 4.0 4.2 4.0 4.0 0 Rentime Application Self-Protection 3.7 4.0 4.2 4.0 0 0 Manual source code review 4.1 4.0 4.8 5.0 0 0</td> <td>-[</td> <td>erence one private feed, not multiple (Application Security Testing Establish vetting process for Open-So tories Code isolation and sandboxing Runtime Application Self-Protection Mamual source code review</td> <td>urce pos-</td> <td>4.1 3.9 3.7 4.1</td> <td>4.0 4.0 4.0</td> <td>10 10 10</td> <td>4.2 4.2 4.8</td>	Scored packages [22, [74] 22 30 J 22 30 J 24 37 4.0 J 26 50 J 27 50 J 26 50 J	Establish internal repository mirrors and ref- rence one private feed, nor milliple [2] 3.6 3.0 Jan 3.7 4.0 Jan 0 Application Security Testing 4.1 4.0 Jan 4.3 5.0 Jan 0 Establish vetting process for Open-Source components hold in internal/public repos- tories block in and sandboxing 3.9 4.0 Jan 4.2 4.0 Jan 0 Runtime Application Self Protection 3.7 4.0 Jan 4.2 4.0 Jan 0 Manual source code review 4.1 4.0 Jan 4.8 5.0 Jan 0	Application Security Testing 4.1 4.0 4.3 5.0 0 Establish vetting process for Open-Source components hosted in internal/public repositions 4.1 4.0 4.3 5.0 0 Code isolation and sandboxing 3.9 4.0 4.2 4.0 4.0 0 Rentime Application Self-Protection 3.7 4.0 4.2 4.0 0 0 Manual source code review 4.1 4.0 4.8 5.0 0 0	-[erence one private feed, not multiple (Application Security Testing Establish vetting process for Open-So tories Code isolation and sandboxing Runtime Application Self-Protection Mamual source code review	urce pos-	4.1 3.9 3.7 4.1	4.0 4.0 4.0	10 10 10	4.2 4.2 4.8

Experts

Develope

Summary

- Attackers are increasingly using vulnerabilities unintentionally injected into vulnerabilities or are maliciously injecting vulnerabilities into the supply chain
- We need to be smart about:
 - Detecting vulnerabilities
 - Updating components
 - Making good component choices
 - Implementing safeguards