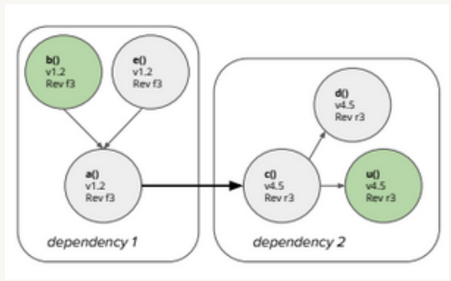
Method-Level Data in GitHub Pull Request Descriptions: Effects on Developers' Prioritization and Facilitation of Fixing Vulnerable Dependencies

Student - Tudor-Alexandru Popovici; Supervisors - Sebastian Proksch and Mehdi Keshani

t.popovici@student.tudelft.nl
02.07.2021

1 Background

- Dependency maintenance tools (i.e. *Dependabot*) help keep projects **vulnerability-free.**
- **Dependabot** opens **Pull Requests (PRs)** to update vulnerable dependencies in a project.
- These tools perform **package-level analyses** to **detect** vulnerabilities in projects.
- **Package-level analysis** is **prone** to giving **false positive (FP)** vulnerability results.
- **Method-level analysis** is a candidate option for **reducing FPs**, making use of **call graphs (CGs)**.
- **FASTEN Project** provides a library to generate CGs between a project and its dependencies.
- **FASTEN Database,** which contains both package and method-level vulnerability data on a vast amount of projects can be used.
- **Method-level analysis produces** a set of CG **path traces** from the project methods to the vulnerable dependency methods. This is known as **method-level data**



Generated call graph (CG) between two dependencies

2 Research Question

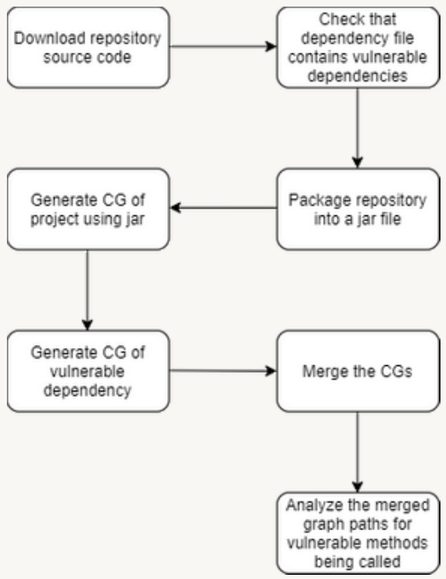
- Does the **fine-grained information** in the GitHub Pull Request descriptions help developers to **prioritize** the task?
- Does extra CG information make it **easier** for developers to **deal** with **vulnerable** dependencies?

3 Methodology

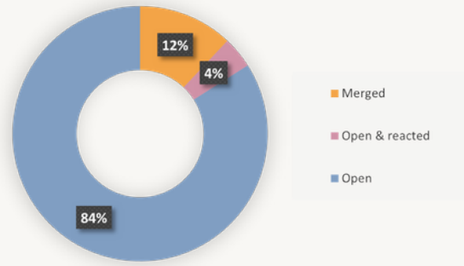
1. **Select** the **project set** on which to perform the study.
2. **Retrieve** a **vulnerable package set** to look for as **dependencies** in the selected **projects**.
3. **Implement** a **vulnerability analyzer** to perform package and method-level analyses on selected projects.
4. **Analyse** the selected **repositories** for vulnerabilities on the **package-level**.
5. **Analyse** the **positive package-level** repositories for vulnerabilities on the **method-level**.
6. **Open PRs** on GitHub for projects which are **vulnerable** on the **method-level**.
7. **Collect** and **process** data on **developers' reactions** through **survey**.

4 Method-level Data Collection

- 7.638 **projects** collected, **6.717 hosted** on **GitHub**.
- **211** packages linked to **393** vulnerabilities retrieved.
- **564** projects **package-level vulnerable.**
- **24** projects **method-level vulnerable.**
- **25** vulnerabilities found**.**

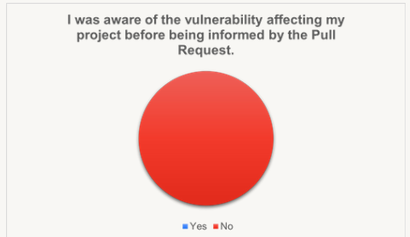


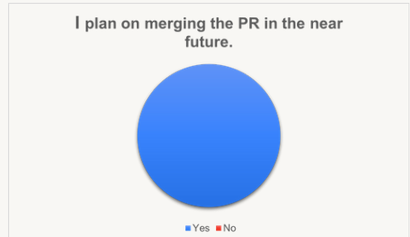
5 Results

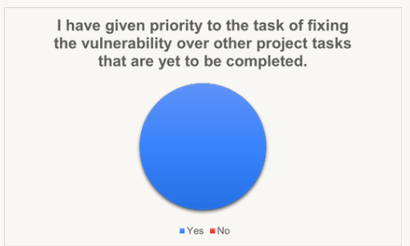


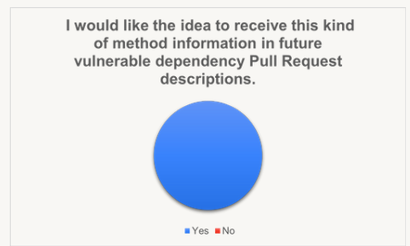
Project	Status	Stars	Forks	Contributors	Uses Dependabot	Affected by
#1	Merged	49	38	15	No	CVE-2019-14379
#2	Open & reacted	34	49	108	Yes	HTTPCLIENT-1803
#3	Merged	11	19	9	No	CVE-2019-14379
#4	Merged	144	57	24	Yes	HTTPCLIENT-1803

Project	No. dependency PRs	PR merge time			Ratio merged dependency PRs
		Recorded	Average	Median	
#1	7	1d	30.8d	6.5d	100%
#2	10	-	7.8d	2d	50%
#3	2	1d	1d	1d	50%
#4	11	22d	15.1d	1d	54.5%









6 Conclusions

- **Method-call data makes** developers **prioritize** the task of **fixing** vulnerabilities.
- **Developers indicate** that their security fix process **is to an extent facilitated** by the provided data.
- **Not enough data** collected to confidently **support** observations.
- **More data expected** as target projects part of **OSS**

TU Delft
Delft University of Technology