Preliminary Findings on FOSS Dependencies and Security

A Qualitative Study on Developers' Attitudes and Experience

Ivan Pashchenko, Duc-Ly Vu, Fabio Massacci
University of Trento, Italy

Vulnerable Deps - Cause of Disaster

Disaster Rank	OWASP Top 10	# of Breaches Root Cause	% of Breaches Root Cause
1	**Components with known vulns**	**12**	**24%**
2	Security misconfiguration	10	18%
3	SQL-injection	4	8%
4	Weak Authentication	3	6%
4	Sensitive Data Exposure	3	6%
5	Function level Access control	2	4%
…			

https://snyk.io/blog/owasp-top-10-breaches/

Developers keep using vuln deps…

Derr et al. [1]: Many dependencies are vulnerable, but could be easily updated	but	Huang et al. [2]: `Easy' update would have broken around 50% of dependent projects
Kula et al. [3]: Many Java libraries do not react on security updates	but	Pashchenko et al. [4]: Some vulnerabilities are in test/dev scopes, hence, not exploitable

1. E. Derr, S. Bugiel, S. Fahl, Y. Acar, and M. Backes. 2017. Keep me updated: An empirical study of third-party library updatability on Android. In *Proc. of CCS'17*.
2. J. Huang, N. Borges, S. Bugiel, and M. Backes. 2019. Up-To-Crash: Evaluating Third-Party Library Updatability on Android. In *Proc. of EuroS&P'19*.
3. R.G. Kula, D.M. German, A. O. TakashiIshio, and K.Inoue. 2017. Do developers update their library dependencies? *Emp. Soft. Eng. Journ*.
4. I. Pashchenko, H. Plate, S.E. Ponta, A. Sabetta, and F. Massacci. 2018. Vulnerable Open Source Dependencies: Counting Those That Matter. In *Proc. of ESEM'18*.

Developers may not be entirely irrational in not always updating dependencies

Interviewees in our sample

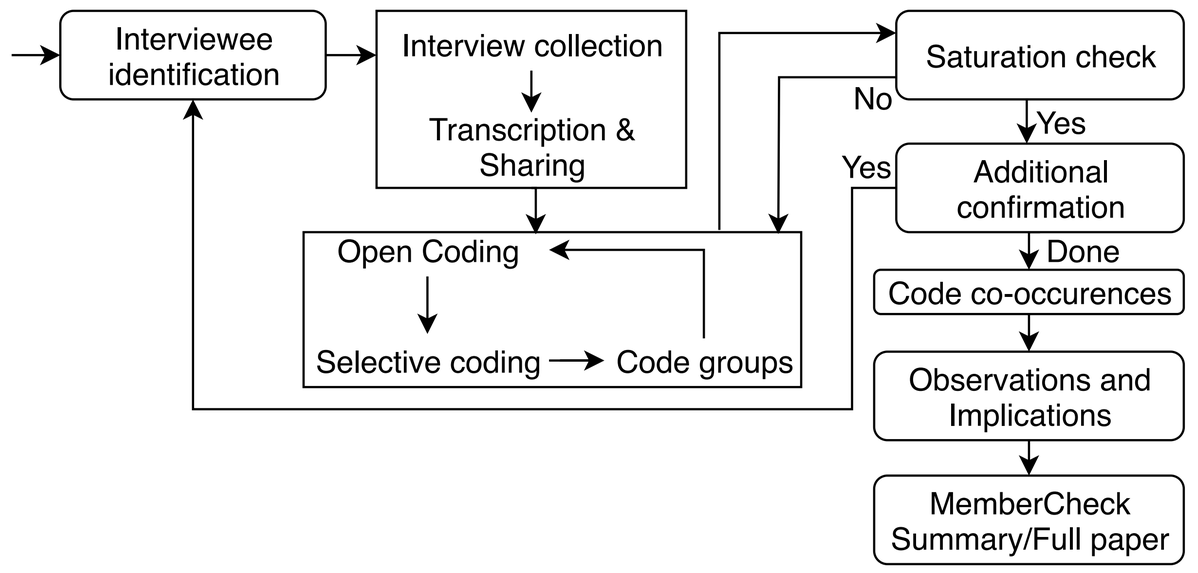
#	Position	Comp. type	Country	Exper. (years)	Languages
1	CTO	SME	DE	3+	Python,JS
2	Moderator	UG	IT	10+	Java
3	Developer	LE	IT	10+	Java,JS
4	CEO	SME	SI	7+	Python,JS
5	Developer	SME	NL	3+	Python
6	Freelancer	SME	RU	3+	Python,JS
7	Developer	SME	DE	5+	Python,JS
8	Developer	LE	RU	4+	Python,JS
9	CTO	SME	IT	4+	JS
10	Developer	LE	DE	10+	C/C++
11	Developer	LE	VN	5+	C/C++
12	Developer	SME	DE	4+	Java,Python
13	Team Leader	LE	RU	10+	JS
14	Developer	SME	RU	4+	Java
15	Project Leader	FOSS	UK	10+	Python,C/C++
16	Developer	SME	IT	8+	Java
17	Developer	LE	VN	3+	Java
18	Sr Software Engineer	LE	IT	10+	Python,C/C++
19	Developer	SME	RU	3+	Java
20	Security Engineer	LE	DE	3+	JS
21	Developer	SME	HR	3+	JS
22	Developer	SME	IT	8+	JS
23	Developer	LE	IT	9+	Java
24	Full Stack Developer	SME	IT	3+	JS,Python
25	Developer	SME	ES	3+	C/C++

Interview topics

We interviewed developers of 25 companies from 9 countries:

- Selecting new dependencies
- Updating currently used dependencies
- Using automatic dependency management tools
- Mitigating bugs and vulnerabilities, for which there is no fixed dependency version

Analysis approach



Preliminary findings*

Library selection:

- Developers pay attention to security only if it is required and enforced by the policy of their company.
- Rely on popularity and community support of libraries (e.g., number of stars, forks, project contributors).

Updating software dependencies:

- Avoid updating dependencies for any reason (afraid of breaking changes).
- Security motivate for updating only if vulnerabilities are severe, widely known, and adoption of the fixed dependency version does not require significant efforts.

Automation of dependency management:

- Sensitive tasks (e.g., updates) performed manually
- Current dependency analysis tools (if used) only facilitate the identification of vulnerabilities in the project dependencies
- Dependency tools produce many false-positive and low-priority alerts

Unfixed vulnerabilities:

- assess whether this vulnerability impacts their projects;
- wait for the fix or a community workaround;
- adapt own project: disable affected functionality or rollback to a safe version;
- maintain own fork of a dependency project (possibly fixing and making a pull request to the dependency project).

*For complete findings, please, refer to: I. Pashchenko, D.L. Vu, and F. Massacci. 2020. A qualitative study of dependency management and its security implications. To appear in *Proc. of CCS'20*. (https://bit.ly/pashchenko2020qualitative)

Cyber Security for Europe — NECS

Future Work

- Broaden our study to more countries
- Find actionable implications of the analysis results
- Correlate results with different type of companies

Contact information

E-mail: ivan.pashchenko@unitn.it
Skype: ivanpashchenko
Web-site: http://disi.unitn.it/~pashchenko