SECOM: Towards a convention for security commit messages

Sofia Reis
sofia.o.reis@tecnico.ulisboa.pt
INESC-ID & IST/Técnico, U. of Lisbon
Lisbon, Portugal

Rui Abreu
rui@computer.org
INESC-ID & FEUP, U. Porto
Porto, Portugal

Hakan Erdogmus
Corina Păsăreanu
hakan.erdogmus@west.cmu.edu
pcorina@cmu.edu
Carnegie Mellon University
USA

ABSTRACT

One way to detect and assess software vulnerabilities is by extracting security-related information from commit messages. Automating the detection and assessment of vulnerabilities upon security commit messages is still challenging due to the lack of structured and clear messages. We created a convention, called SECOM, for security commit messages that structure and include bits of security-related information that are essential for detecting and assessing vulnerabilities for both humans and tools. The full convention and details are available here: https://tqrg.github.io/secom/.

CCS CONCEPTS

• **Software and its engineering** → Software evolution; • **Security and privacy** → **Software security engineering**.

KEYWORDS

security commit messages, convention, standard, best practices

ACM Reference Format:
Sofia Reis, Rui Abreu, Hakan Erdogmus, and Corina Păsăreanu. 2022. SECOM: Towards a convention for security commit messages. In *19th International Conference on Mining Software Repositories (MSR '22), May 23–24, 2022, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3524842.3528513

1 INTRODUCTION

Detecting, and especially assessing software vulnerabilities, continue to be a challenge in vulnerability prediction due to the scarcity and poor quality of curated data [1]. Several researchers have created datasets of security patches based on data collected from software repositories [2–6]. However, there are still very few known "gold standard" datasets useful for comparison and evaluation of the different approaches [7]. One way to detect and assess software vulnerabilities is by extracting security-related information from commit messages [2, 8]. Yet, automating the detection and assessment of vulnerabilities based on security commit messages is still challenging due to the lack of structured and clear messages.

Are security-relevant commit messages informative? We conducted an empirical analysis of ∼ 2k security commit messages collected from GitHub commits included in the references of CVE reports. We confirmed that 23% of these messages used to patch publicly-known vulnerabilities are 1) cryptic/poorly documented, or 2) not clearly related to security issues. These results suggest that guidance, best practices, and standardized templates could help security engineers creating better security commit messages.

How to write a good security commit messages? We searched for sources on writing security commits messages. But we only found guidelines for writing better generic commit messages [9–12], which do not consider crucial security-related information such as the CWE-ID, CVE-ID, impact/score of the vulnerability, and other details associated with vulnerabilities. These bits of security-related information are essential for the detection and assessment of vulnerabilities through commit messages for both humans and tools. Therefore, we created a convention for security commit messages that structure and integrate information about the vulnerabilities.

2 SECOM: A CONVENTION FOR SECURITY COMMIT MESSAGES.

The convention was created based on well-known sources [9–12] on writing good commit messages to facilitate the adoption. The structure and set of fields included in the convention were inferred from 1) the results of our empirical analysis of security-related commit messages; and, 2) feedback collected from presentations given to two Open Source Security Foundation (OpenSSF) working groups named "Best Practices" and "Vulnerability Disclosure".

The convention consists of the following different sections: **header**, includes the type `vuln-fix`, a simple description of the vulnerability and its identifier (when available); **body**, describes the vulnerability (what), its impact (why) and the patch to fix the vulnerability (how); **metadata**, such as type of weakness (CWE-ID), severity, CVSS, detection methods, report link and version of the software where the vulnerability was introduced; **authors** and **reviewers**; and, **references** to bug trackers.

3 FEEDBACK AND FUTURE IDEAS

Feedback received from the security community suggests that they see value in SECOM and would like to see it evolve into a standard practice. Writing more structured and informative commit messages for vulnerability disclosure and patching will improve the detection and assessment of security vulnerabilities through commit messages. In the future, new technologies can be developed on top of SECOM to help development teams assess compliance with the standard and automate the creation of structured security commit messages using features such as recommendations and auto-completion.

MSR '22, May 23–24, 2022, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-9303-4/22/05…$15.00
https://doi.org/10.1145/3524842.3528513

REFERENCES

[1] S. Chakraborty, R. Krishna, Y. Ding, and B. Ray. Deep learning based vulnerability detection: Are we there yet. *IEEE Transactions on Software Engineering*, (01):1–1, jun 5555.

[2] Sofia Reis and Rui Abreu. SECBENCH: A database of real security vulnerabilities. In *International Workshop on Secure Software Engineering in DevOps and Agile Development co-located with the (ESORICS 2017), Oslo, Norway, September 14, 2017*, pages 69–85, 2017.

[3] Serena E. Ponta, Henrik Plate, Antonino Sabetta, Michele Bezzi, and Cédric Dangremont. A manually-curated dataset of fixes to vulnerabilities of open-source software. In *Proceedings of the 16th International Conference on Mining Software Repositories*, MSR '19, page 383–387. IEEE Press, 2019.

[4] Jiahao Fan, Yi Li, Shaohua Wang, and Tien N. Nguyen. *A C/C++ Code Vulnerability Dataset with Code Changes and CVE Summaries*, page 508–512. Association for Computing Machinery, New York, NY, USA, 2020.

[5] Guru Bhandari, Amara Naseer, and Leon Moonen. Cvefixes: Automated collection of vulnerabilities and their fixes from open-source software. In *Proceedings of the 17th International Conference on Predictive Models and Data Analytics in Software Engineering*, PROMISE 2021, page 30–39, New York, NY, USA, 2021. Association for Computing Machinery.

[6] Arthur D. Sawadogo, Tegawendé F. Bissyandé, Naouel Moha, Kevin Allix, Jacques Klein, Li Li, and Yves Le Traon. Learning to catch security patches. *CoRR*, abs/2001.09148, 2020.

[7] Hazim Hanif, Mohd Hairul Nizam Md Nasir, Mohd Faizal Ab Razak, Ahmad Firdaus, and Nor Badrul Anuar. The rise of software vulnerability: Taxonomy of software vulnerabilities detection and machine learning approaches. *Journal of Network and Computer Applications*, 179:103009, 2021.

[8] Yunhui Zheng, Saurabh Pujar, Burn Lewis, Luca Buratti, Edward Epstein, Bo Yang, Jim Laredo, Alessandro Morari, and Zhong Su. *D2A: A Dataset Built for AI-Based Vulnerability Detection Methods Using Differential Analysis*, page 111–120. IEEE Press, 2021.

[9] Conventional commits. https://www.conventionalcommits.org/en/v1.0.0/. Accessed April 26, 2022.

[10] Sean Patterson. Developer tip: Keep your commits "atomic". https://www.freshconsulting.com/insights/blog/atomic-commits/. Accessed April 26, 2022.

[11] Linus Torvalds. Linus torvalds describes a good commit message. https://github.com/torvalds/subsurface-for-dirk/blob/a48494d2fbed58c751e9b7e8fbff88582f9b2d02/README#L88. Accessed April 26, 2022.

[12] Chris Beams. How to write a git commit message. https://cbea.ms/git-commit/. Accessed April 26, 2022.