School of
Information Systems

SMU
SINGAPORE MANAGEMENT UNIVERSITY

HERMES: Using Commit-Issue Linking to Detect Vulnerability-fixing Commit

Truong-Giang Nguyen, Hong Jin Kang, David Lo
Singapore Management University

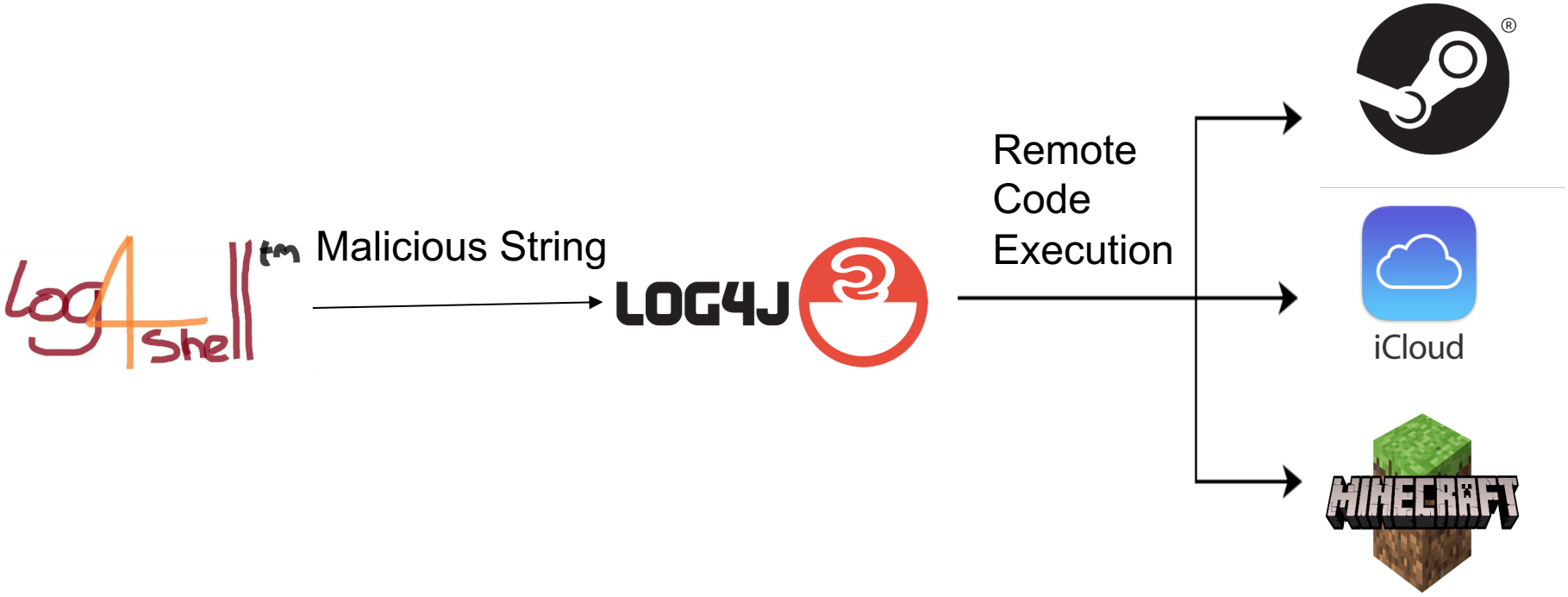
Abhishek Sharma, Andrew E. Santosa, Asankhaya Sharma, Ming Yi Ang
Veracode

Content

- Motivation and Challenge
- Approach
- Experimental result

Motivation

- Modern software relies on third-party libraries
- Open Source Software (OSS) users are exposed to vulnerabilities (e.g., Log4Shell)



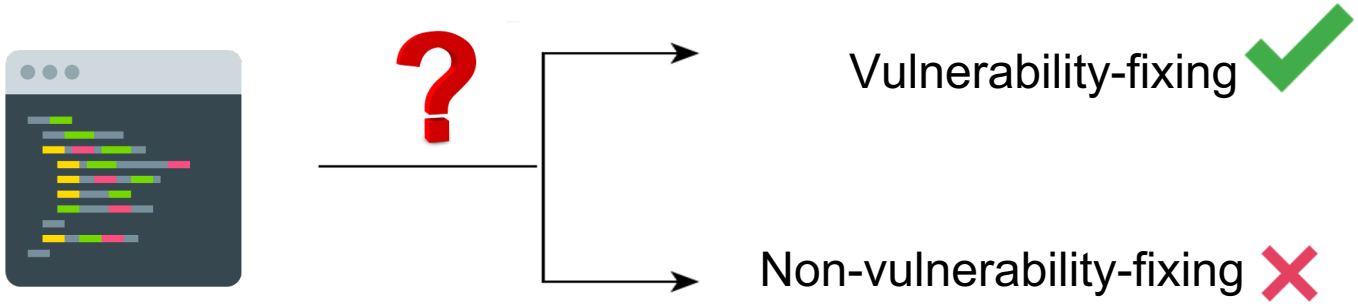
Motivation

- OSS users must keep up-to-date with vulnerabilities by monitoring public vulnerabilities advisories

Motivation

- Vulnerability disclosure can vary from days to years
  - CVE-2018-11766 was disclosed two months later after patching

- Solution for in-time vulnerability monitoring => Automatically identify vulnerability-fixing commits

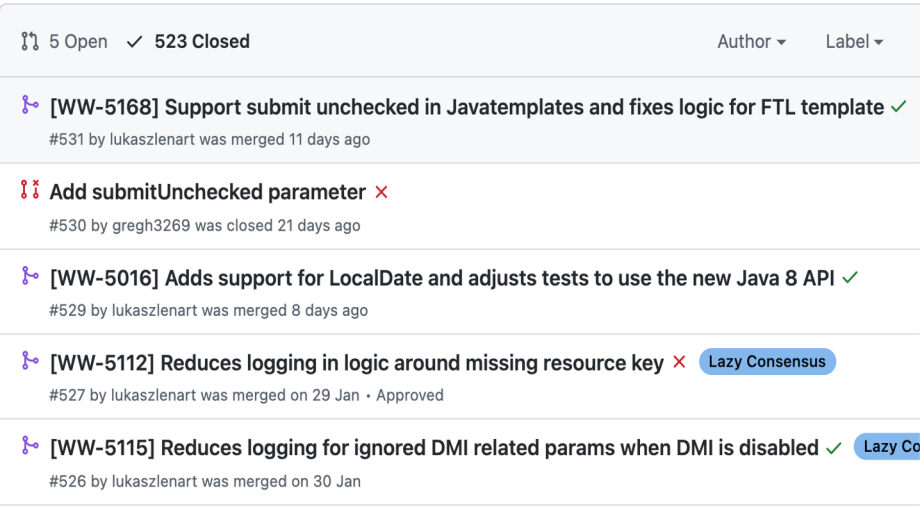


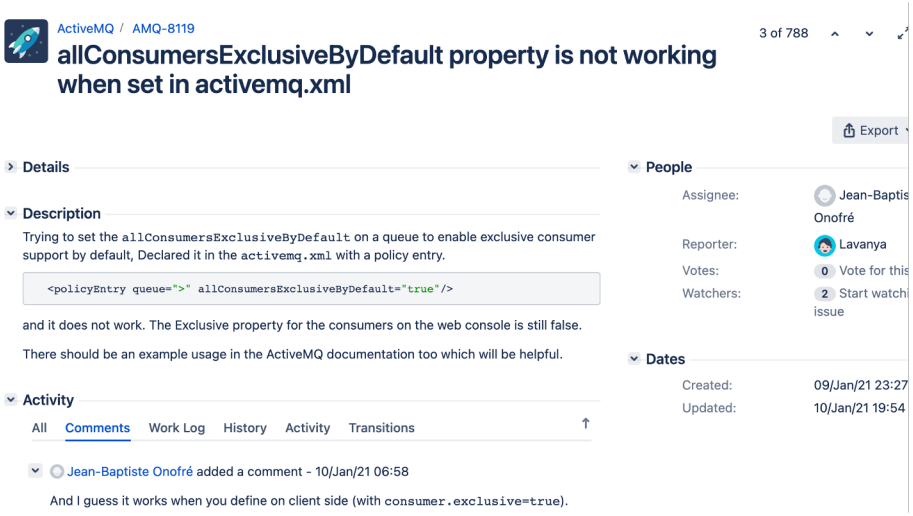
Commit

Motivation

- Propose techniques rely on commit messages and code changes

- Issue trackers contain rich source of information
  - GitHub issue
  - JIRA





Motivation - Example

× **Fix #486**

master

release-1.13.1 ··· release-1.12.0

decebals committed on 12 Dec 2018

```
@@ -25,6 +25,9 @@
25   import javax.xml.bind.JAXBException;
26   import javax.xml.bind.Marshaller;
27   import javax.xml.bind.Unmarshaller;
28 + import javax.xml.stream.XMLInputFactory;
29 + import javax.xml.stream.XMLStreamException;
30 + import javax.xml.stream.XMLStreamReader;
31   import java.io.StringReader;
32   import java.io.StringWriter;
33

@@ -69,10 +72,16 @@ public String toString(Object object) {
72       public <T> T fromString(String content, Class<T> classOfT) {
73           try (StringReader reader = new StringReader(content)) {
74               JAXBContext jaxbContext = JAXBContext.newInstance(classOfT);
   -            Unmarshaller jaxbUnmarshaller = jaxbContext.createUnmarshaller();
75
   -            return (T) jaxbUnmarshaller.unmarshal(reader);
   -        } catch (JAXBException e) {
76 +            XMLInputFactory xmlInputFactory = XMLInputFactory.newFactory();
77 +            xmlInputFactory.setProperty(XMLInputFactory.IS_SUPPORTING_EXTERNAL_ENTITI
78 +            xmlInputFactory.setProperty(XMLInputFactory.SUPPORT_DTD, true);
79 +            XMLStreamReader xmlStreamReader = xmlInputFactory.createXMLStreamReader(r
80 +
81 +            Unmarshaller unmarshaller = jaxbContext.createUnmarshaller();
82 +
83 +            return (T) unmarshaller.unmarshal(xmlStreamReader);
84 +        } catch (JAXBException | XMLStreamException e) {
85               throw new PippoRuntimeException(e, "Failed to deserialize content to '{}'
86           }
```

Closed **xxe vulnerabilities** #486

QiAnXinCodeSafe opened this issue on 11 Dec 2018 · 3 com...

Hello, I am a member of the 360 Code Guard team. In our open source project code audit, we found that Pippo has xxE vulnerabilities. Details are as follows.
pippo/pippo-content-type-parent/pippo-jaxb/src/main/java/ro/pippo/jaxb/JaxbEngine.java

Because the XML parser does not disable dtd, xxE attacks can occur when content parameters are controlled by malicious attackers

SMU SINGAPORE MANAGEMENT UNIVERSITY

Challenge

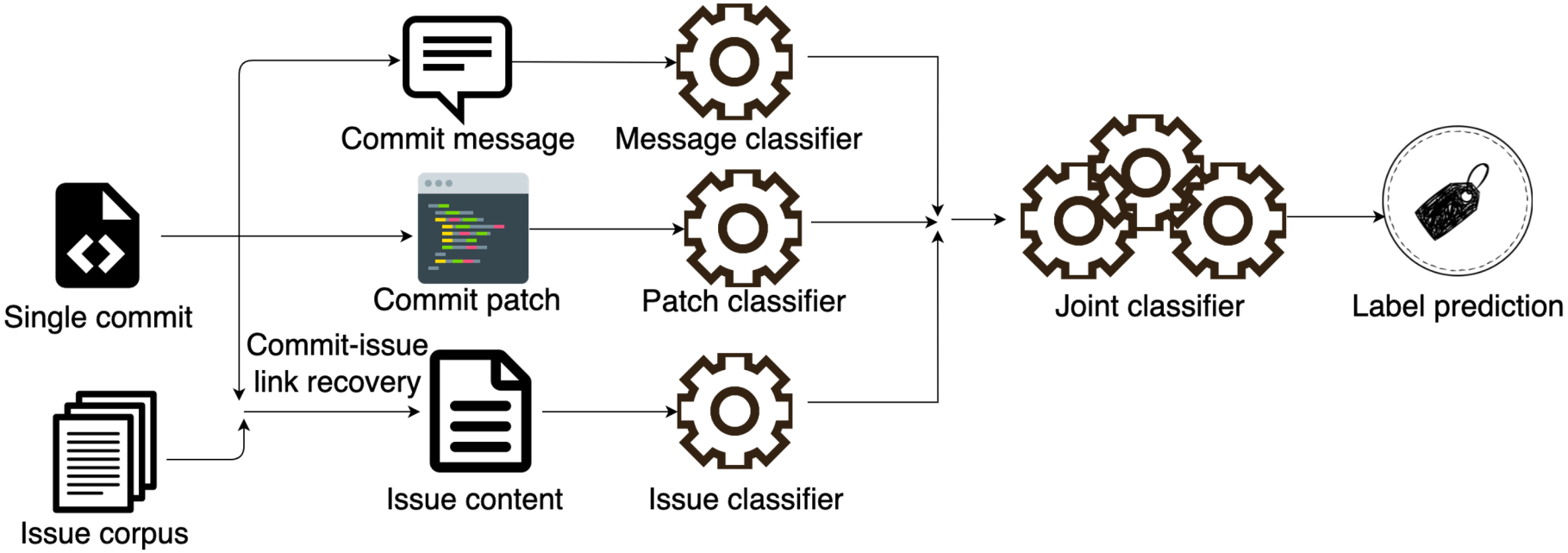
- However, many commits are not linked to issues
  - The proportion of unlinked commits have been reported from 35% to 40%
  - In our dataset, nearly 63% of commits are unlinked

- Solution: HERMES uses existing commit-issue link recovery technique to infer links between each unlinked commit and an issue that best matches it

Approach

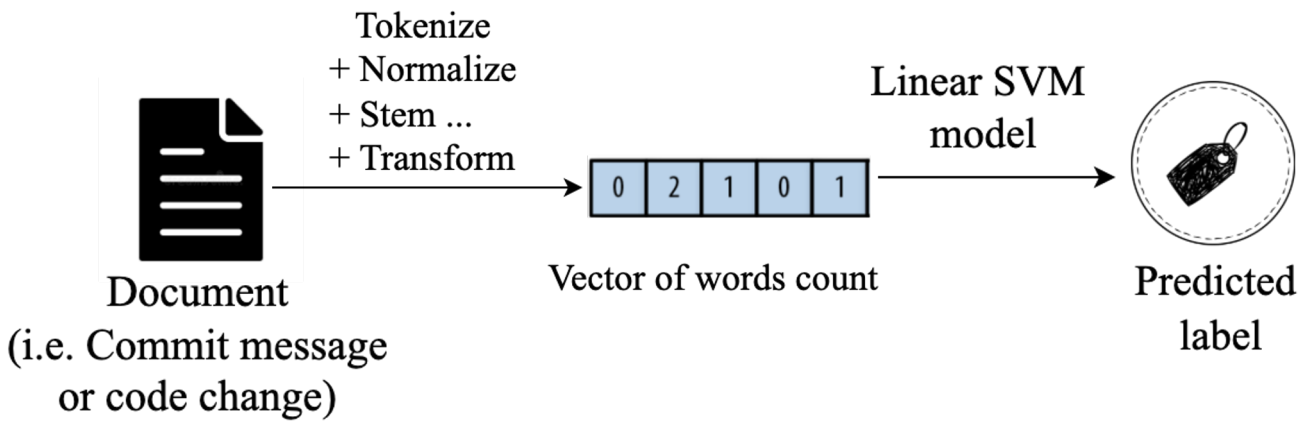
- High-level design of HERMES consists of:
  - An issue linker
  - Three base classifiers (message classifier, code change classifier, issue classifier)
  - A joint classifier

Approach


Commit message
Message classifier
Single commit
Commit patch
Patch classifier
Joint classifier
Label prediction
Commit-issue
link recovery
Issue corpus
Issue content
Issue classifier

Approach

- Message Classifier + Patch Classifier
  - Proposed by Sabetta et al. [1]



[1] Sabetta, A., & Bezzi, "A practical approach to the automatic classification of security-relevant commits", ICSME 2018

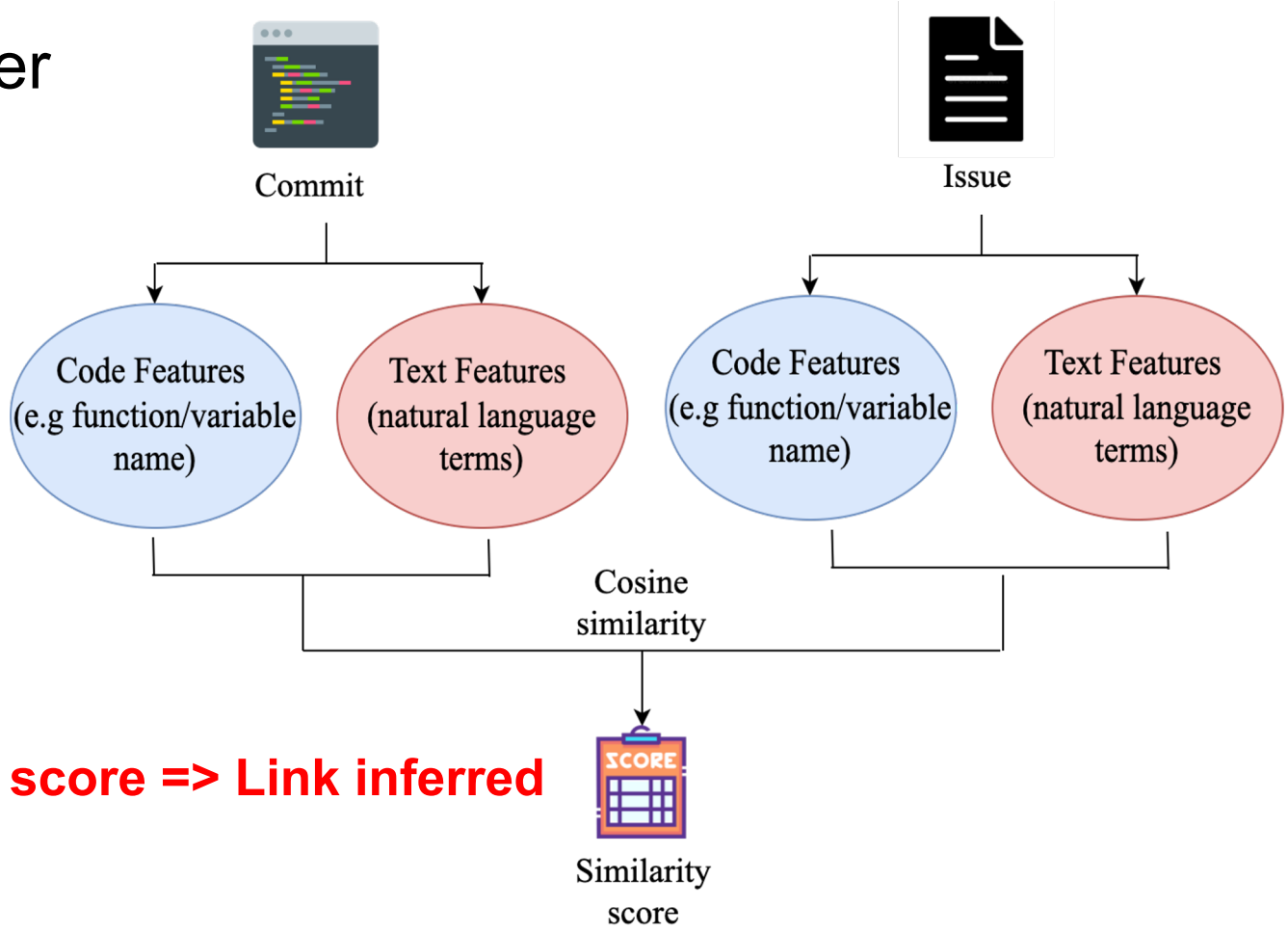
Approach

- Issue Classifier
  - Commit with explicit links to issues
    - Manually identify relevant issue trackers
    - Use regular expression for matching issue ID in commit message (e.g. CAMEL-16527”, ”WW-4348”, ”STS-262”)
  - Commit without explicitly link to any issue
    - Build a corpus of over 290k issues from multiple projects in the dataset
    - Implement FRLink[1] as an issue linker

[1] Sun, Yan and Wang, Qing and Yang, Ye, “Frlink: Improving the recovery of missing issue-commit links by revisiting file relevance”, Information and Software Technology 2017

Approach

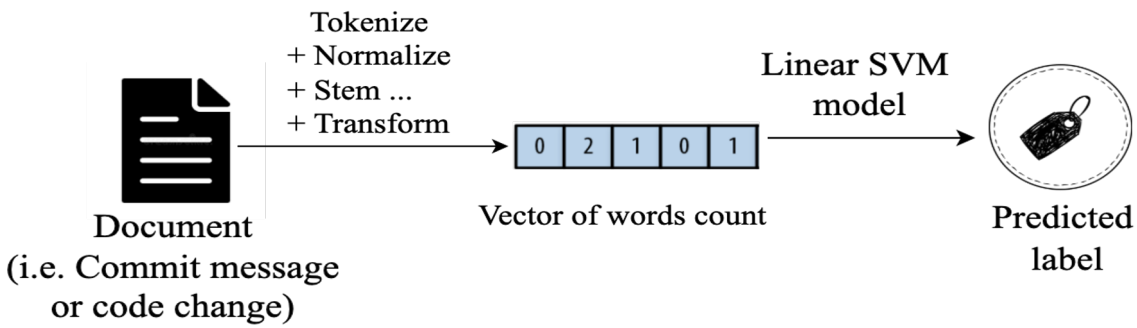
- Issue linker



Highest similarity score => Link inferred

Approach

- Issue Classifier (follow up)
  - Issue content extraction
    - GitHub issue: Title, Body, Comment(s)
    - JIRA issue: Summary, Description, Comment(s)



- Joint classifier
  - Employs a Logistic Regression Classifier to combine outputs of three base classifier

Experimental Result

- How effective is HERMES for commits with explicit links?
  - Evaluate HERMES on subset of dataset where commit and issue are linked by commit authors

Model	Precision	Recall	F1
Sabetta el al. [1]	0.54	0.82	0.64
HERMES	0.8	0.67	**0.72**

[1] Sabetta, A., & Bezzi, “A practical approach to the automatic classification of security-relevant commits”i, ICSME 2018

Experimental Result

- How effective is HERMES when leveraging commit-issue link recovery technique?
  - Evaluate HERMES on the full dataset after performing commit-issue linking

Model	Precision	Recall	F1
Sabetta el al. [1]	0.52	0.81	0.63
HERMES	0.74	0.66	**0.70**

[1] Sabetta, A., & Bezz, “A practical approach to the automatic classification of security-relevant commits”i, ICSME 2018

Thank for watching