TUHH Hamburg University of Technology

SoftSEC Institute of Software Security

SEC 4 AI 4 SEC

Funded by the European Union

Technical Track @ MSR'26

A Match Made in Heaven? AI-driven Matching of Vulnerabilities and Security Unit Tests

Emanuele Iannone, Quang-Cuong Bui, Riccardo Scandariato

***Institute of Software Security**, Hamburg University of Technology, Germany*

emanuele.iannone@tuhh.de

https://emaiannone.github.io/

emaiannone

Paper Preprint

*This work was supported by the **Horizon Europe project "Sec4AI4Sec"** (grant no. 101120393).*

Security Unit Tests: Context and Problem

```java
@Test public void rejectsTooLongPath() {
  String source = "foo.bar";
  for (int i = 0; i < 9; i++) {
    source = source + "." + source;
  }
  assertThat(source.split("\\.").length,
    is(greaterThan(1000)));
  final String path = source;
  exception.expect(IllegalArgumentException.class);
  PropertyPath.from(path, Left.class);
}
```

Test for **CVE-2018-1274** (resource allocation error) in **Spring Data Commons**

❌ **Fails on vulnerable code**

✅ **Passes on patched code**

We call tests like this:

Vulnerability-witnessing tests

Not much is known about them 🥀

👉 **We do not find many in vulnerability-fixing commits.**

👉 **If they exist in test suites, they are not explicitly mapped to known vulnerabilities.**

🎯 **FIND vulnerability tests in project repositories and MATCH them to the correct vulnerability statically (running tests is inconvenient).**

Preprint

Emanuele Iannone, *Quang-Cuong Bui, Riccardo Scandariato*

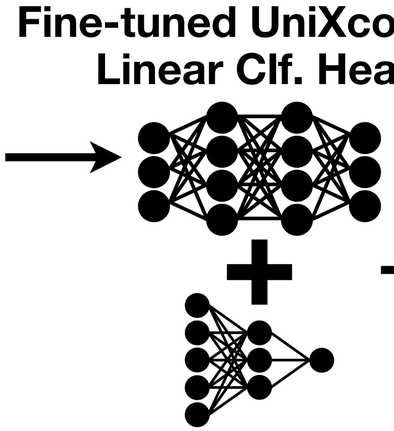
VuTeCo: Vulnerability Test Collector

VuTeCo makes two types of classifications…

Finding Task

```
@Test public void rejectsTooLongPath() {
  String source = "foo.bar";
  for (int i = 0; i < 9; i++) {
    source = source + "." + source;
  }
  assertThat(source.split("\\.").length,
    is(greaterThan(1000)));
  final String path = source;
  exception.expect(IllegalArgumentException.class);
  PropertyPath.from(path, Left.class);
}
```

Test Method



0.4

Probability of Positive Class

Security

Unclear

Matching Task

```
@Test public void rejectsTooLongPath() {
  String source = "foo.bar";
  for (int i = 0; i < 9; i++) {
    source = source + "." + source;
  }
  assertThat(source.split("\\.").length,
    is(greaterThan(1000)));
  final String path = source;
  exception.expect(IllegalArgumentException.class);
  PropertyPath.from(path, Left.class);
}
```

Test Method

```
Application X is
affected by...
```

Vulnerability Description

Prompt Template

```
You are an expert...
Vulnerability description: ...
JUnit test method: ...
```

Fine-tuned DeepSeek Coder 6.7B-Instruct

Extract Classification (Regex)

Matched

Not Matched

Preprint

VuTeCo Experimental Evaluation (RQ1)

RQ_1 **Best AI model to find security tests and match them to vulnerabilities.**

Vul4J*

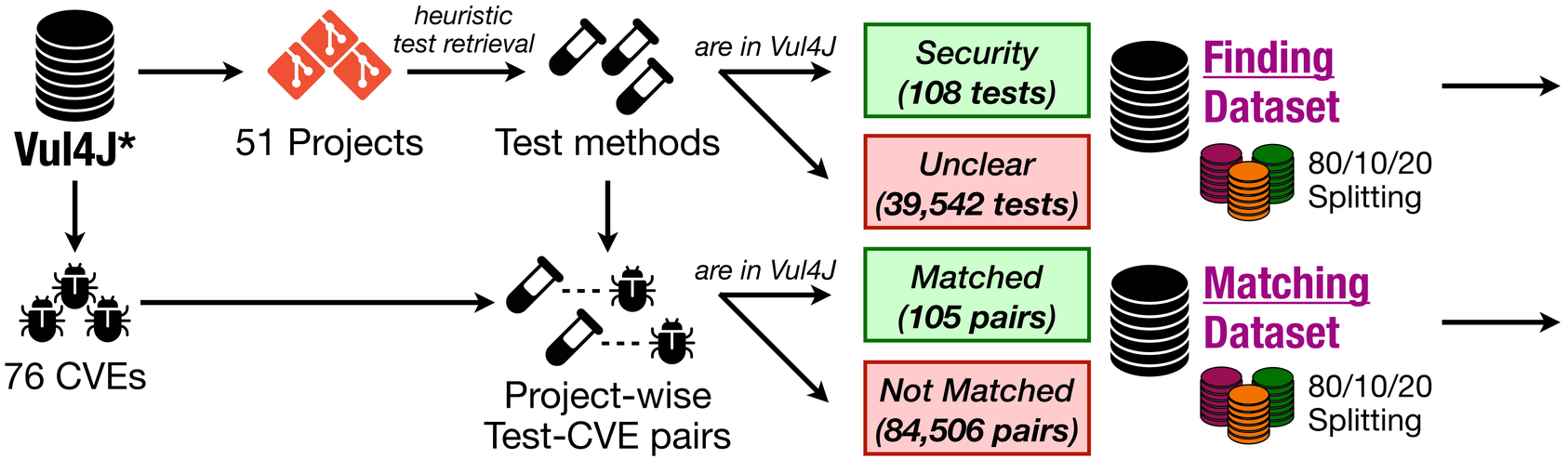
79 Java vulnerabilities with confirmed witnessing tests.

*Q. -C. Bui, R. Scandariato and N. E. D. Ferreyra, "Vul4J: A Dataset of Reproducible Java Vulnerabilities Geared Towards the Study of Program Repair Techniques," 2022 MSR doi: 10.1145/3524842.3528482.

Preprint

VuTeCo Experimental Evaluation (RQ1)

RQ1 **Best AI model to find security tests and match them to vulnerabilities.**



Train and Test

- Encoder + Clf. head: CodeBERT, CodeT5+, UniXcoder
- Instruction-tuned LLM: CodeLlama, DeepSeek Coder, Qwen2.5-Coder
- 7 Baselines: No direct competitors of VuTeCo (both tasks)

Focus on $F_{0.5}$ Score

- **Precision more important** than recall.
- **Build a good dataset** (quality over quantity).

*Q. -C. Bui, R. Scandariato and N. E. D. Ferreyra, "Vul4J: A Dataset of Reproducible Java Vulnerabilities Geared Towards the Study of Program Repair Techniques," 2022 MSR doi: 10.1145/3524842.3528482.

Preprint

VuTeCo Experimental Evaluation (RQ1)

Finding Task

Test Set

21 positive out of 7,931

Approach		Performance						
		Pr	Re	F_1	$F_{0.5}$	MCC	TP	FP
CRM	CodeBERT	0.78	0.34	0.47	0.61	0.51	7	2
	CodeT5+	0.70	0.33	0.45	0.57	0.48	7	3
	UniXcoder	0.83	**0.48**	**0.61**	**0.73**	**0.63**	**10**	2
LLM	CodeLlama	0.69	0.43	0.53	0.62	0.54	9	4
	DeepSeek Coder	0.69	0.43	0.53	0.62	0.54	9	4
	Qwen2.5-Coder	**0.88**	0.33	0.48	0.66	0.54	7	**1**
Baseline	*GrepFind*	0.01	0.24	0.03	0.02	0.05	5	375
	$VocabFind_{YAKE}$	0.08	0.05	0.06	0.07	0.06	1	11
	$VocabFind_{Iden}$	0.02	0.10	0.03	0.02	0.04	2	114

Matching Task

Test Set

21 positive out of 16,923

Approach		Performance						
		Pr	Re	F_1	$F_{0.5}$	MCC	TP	FP
CRM	CodeBERT	0.60	0.29	0.39	0.49	0.41	6	4
	CodeT5+	0.71	0.23	0.36	0.51	0.41	5	2
	UniXcoder	**1.00**	0.24	0.39	0.61	0.49	5	**0**
LLM	CodeLlama	0.64	0.43	0.51	0.52	**0.58**	9	5
	DeepSeek Coder	0.75	0.43	**0.55**	**0.65**	0.57	9	3
	Qwen2.5-Coder	0.86	0.29	0.43	0.61	0.50	6	1
Baseline	*GrepMatch*	0.50	0.14	0.22	0.33	0.27	3	3
	$SimMatch_{YAKE}$	0.01	0.10	0.02	0.01	0.03	2	220
	$SimMatch_{CRM}$	0.33	0.05	0.09	0.15	0.12	1	2
	FixCommits	0.30	**0.57**	0.39	0.41	0.32	**12**	29

The low precision of FixCommits suggests that tests in fixing commits are often not security-related!

Preprint

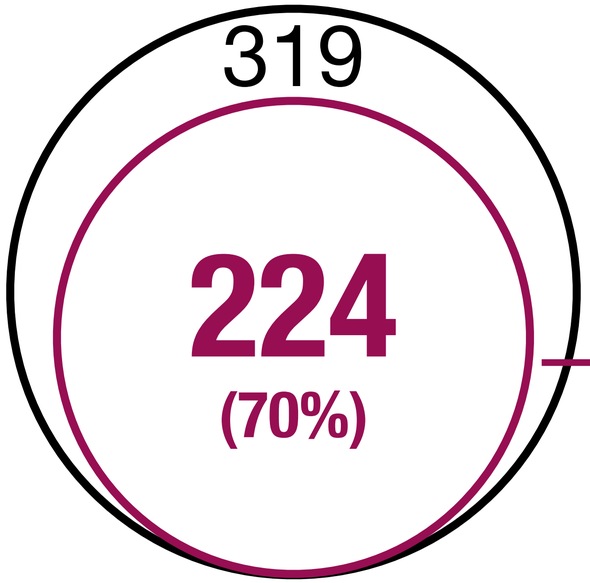
VuTeCo In-the-wild Evaluation (RQ2)

RQ2 **Effectiveness of VuTeCo finding and matching in the wild.**

427 Projects (HEAD revision) → **VuTeCo** → **Precision Assessment** (2 inspectors)

Finding Task

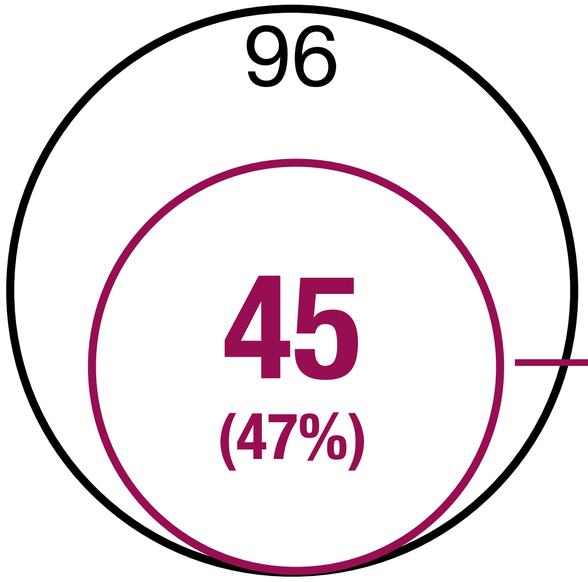
Pool: **1M+ test methods**



vs. 83% in RQ1

Matching Task

Pool: **5M+ test-CVE pairs**



vs. 75% in RQ1

False Positive caused by…

- **Misleading Terms** (“bad”, “inject”)
- **Literal Strings** (URLs, commands)

Need to…

- **Filter noisy tokens**
- **Add context to the input**

Preprint

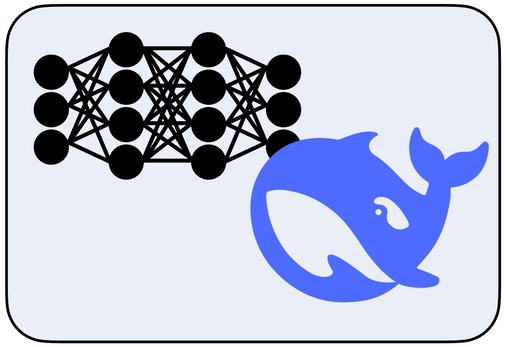
Emanuele Iannone, *Quang-Cuong Bui, Riccardo Scandariato*

… then what?

We collected the validated results in a new public dataset:

Test4Vul

VuTeCo



Now we can finally…

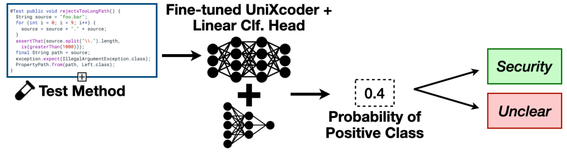
- **Perform analyses on code-level security tests**
- **Train AI models to generate code-level security tests**

Preprint

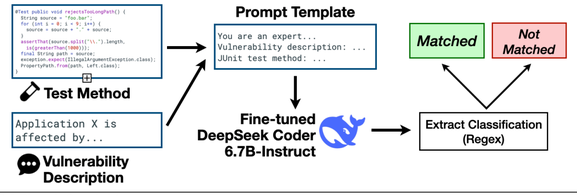
VuTeCo: Vulnerability Test Collector

VuTeCo makes two types of classifications…

Finding Task



Matching Task

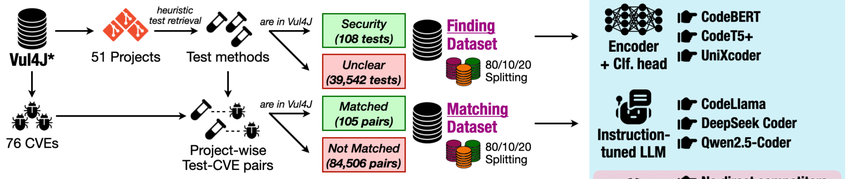


Preprint

Emanuele Iannone, Quang-Cuong Bui, Riccardo Scandariato

VuTeCo Experimental Evaluation (RQ1)

RQ1 Best AI model to find security tests and match them to vulnerabilities.



Focus on $F_{0.5}$ Score

- **Precision** more important than recall.
- **Build a good dataset** (quality over quantity).

*Q. -C. Bui, R. Scandariato and N. E. D. Ferreyra, "Vul4J: A Dataset of Reproducible Java Vulnerabilities Geared Towards the Study of Program Repair Techniques," 2022 MSR doi: 10.1145/3524842.3528482.

Preprint

Emanuele Iannone, Quang-Cuong Bui, Riccardo Scandariato

VuTeCo Experimental Evaluation (RQ1)

Finding Task

Test Set
21 positive out of 7,931

	Approach	Pr	Re	F_1	$F_{0.5}$	MCC	TP	FP
CRM	CodeBERT	0.78	0.34	0.47	0.61	0.51	7	2
CRM	CodeT5+	0.70	0.33	0.45	0.57	0.48	7	3
CRM	**UniXcoder**	0.83	**0.48**	**0.61**	**0.73**	**0.63**	**10**	2
LLM	CodeLlama	0.69	0.43	0.53	0.62	0.54	9	4
LLM	DeepSeek Coder	0.69	0.43	0.53	0.62	0.54	9	4
LLM	Qwen2.5-Coder	**0.88**	0.33	0.48	0.66	0.54	7	**1**
Baseline	*GrepFind*	0.01	0.24	0.03	0.02	0.05	5	375
Baseline	*VocabFind$_{AKE}$*	0.08	0.05	0.06	0.07	0.06	1	11
Baseline	*VocabFind$_{Iden}$*	0.02	0.10	0.03	0.02	0.04	2	114

Matching Task

Test Set
21 positive out of 16,923

	Approach	Pr	Re	F_1	$F_{0.5}$	MCC	TP	FP
CRM	CodeBERT	0.60	0.29	0.39	0.49	0.41	6	4
CRM	CodeT5+	0.71	0.23	0.36	0.51	0.41	5	2
CRM	UniXcoder	**1.00**	0.24	0.39	0.61	0.49	5	**0**
LLM	CodeLlama	0.64	0.43	0.51	0.52	**0.58**	9	5
LLM	**DeepSeek Coder**	0.75	0.43	**0.55**	**0.65**	0.57	9	3
LLM	Qwen2.5-Coder	0.86	0.29	0.43	0.61	0.50	6	1
Baseline	*GrepMatch*	0.50	0.14	0.22	0.33	0.27	3	3
Baseline	*SimMatchy$_{AKE}$*	0.01	0.10	0.02	0.01	0.03	2	220
Baseline	*SimMatch$_{CRM}$*	0.33	0.05	0.09	0.15	0.12	1	2
Baseline	*FixCommits*	0.30	**0.57**	0.39	0.41	0.32	**12**	29

The low precision of FixCommits suggests that tests in fixing commits are often not security-related!

Preprint

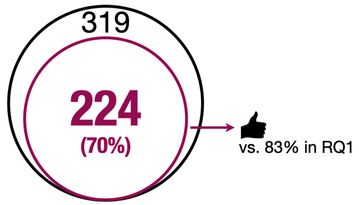
Emanuele Iannone, Quang-Cuong Bui, Riccardo Scandariato

VuTeCo In-the-wild Evaluation (RQ2)

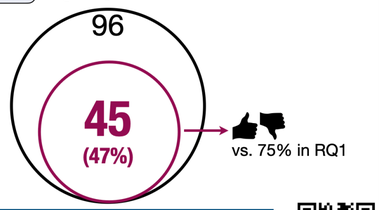
RQ2 Effectiveness of VuTeCo finding and matching in the wild.

427 Projects (HEAD revision) → VuTeCo → Precision Assessment (2 inspectors)

Finding Task — Pool: **1M+ test methods**



Matching Task — Pool: **5M+ test-CVE pairs**



False Positive caused by…
- **Misleading Terms** ("bad", "inject")
- **Literal Strings** (URLs, commands)

Need to…
- **Filter noisy tokens**
- **Add context to the input**

Preprint

Emanuele Iannone, Quang-Cuong Bui, Riccardo Scandariato

SEC 4 AI 4 SEC

Funded by the European Union

Emanuele Iannone

emanuele.iannone@tuhh.de

https://emaiannone.github.io/

emaiannone

tuhh-softsec/vuteco

A Match Made in Heaven? AI-driven Matching of Vulnerabilities and Security Unit Tests

Paper Preprint