Error Delayed is Not Error Handled: Understanding and Fixing Propagated Error-Handling Bugs

Haoran Liu, Shanshan Li, Zhouyang Jia,

Yuanliang Zhang, Linxiao Bai, Si Zheng,

Xiaoguang Mao, Xiangke Liao

National University of Defense Technology

2025.06.23

Error Handling Bug (EH Bugs)

- **Missing proper handling of errors in software systems.**

```
FILE fp = fopen(…);
if (fp == Null)
    …
else
    …
```

Error Handling Code

```
FILE fp = fopen(…);
?
fgets(str, N, fp);
```

Error Handling Bug

(EH Bug)

Impact

- **Threating software reliability**
  - Halting industrial control software [1]
  - Global service downtime [2]
  - $370 million loss [3]
- **Hard to fix**
  - Heavy Manual Effort
  - Partial Solution

[1] Israel National Cyber Directorate. CVE-2024-38435 Jul. 2024.
[2] Lars Rabbe. *skype reveals a bug in its windows client was what crashed its- system*. Feb. 2010.
[3] J L LIONS. Ariane 501 - Presentation of Inquiry Board report. Jul. 1996.

Existing Approaches

➢ General APR approaches

- Mutation based

  ✓Mutation strategies[4-6]

  ✓Templates[7,8]

[4]. Wong, C. P., et al, VarFix: balancing edit expressiveness and search effectiveness in automated program repair. FSE 2021
[5]. Ghanbari A., et al, Practical program repair via bytecode mutation. ISSTA 2019
[6]. Wen M., et al, Context-aware patch generation for better automated program repair. ICSE 2018
[7]. Liu K., et al, Tbar: Revisiting template-based automated program repair. ISSTA. 2019
[8]. Liu K., et al, Avatar: Fixing semantic bugs with fix patterns of static analysis violations. SANER. 2019

Existing Approaches

➢ General APR approaches

- Mutation based
- Generation based
  - ✓Seq2Seq[9, 10]
  - ✓LLM-Enhanced[12,13]

[9]. Ye H, et al. Selfapr: Self-supervised program repair with test execution diagnostics. ICSE. 2022
[10]. Jiang N, et al. Cure: Code-aware neural machine translation for automatic program repair. ICSE. 2021
[11]. Xia C S, et al. Keep the Conversation Going: Fixing 162 out of 337 bugs for $0.42 each using ChatGPT. ISSTA. 2024
[12]. Chen Y, et al. When Large Language Models Confront Repository-Level Automatic Program Repair: How Well They Done. ICSE. 2024
[13]. Jiang N, et al. Impact of code language models on automated program repair. ICSE. 2023

Existing Approaches

- **General APR approaches**
  - Mutation based
  - Generation based

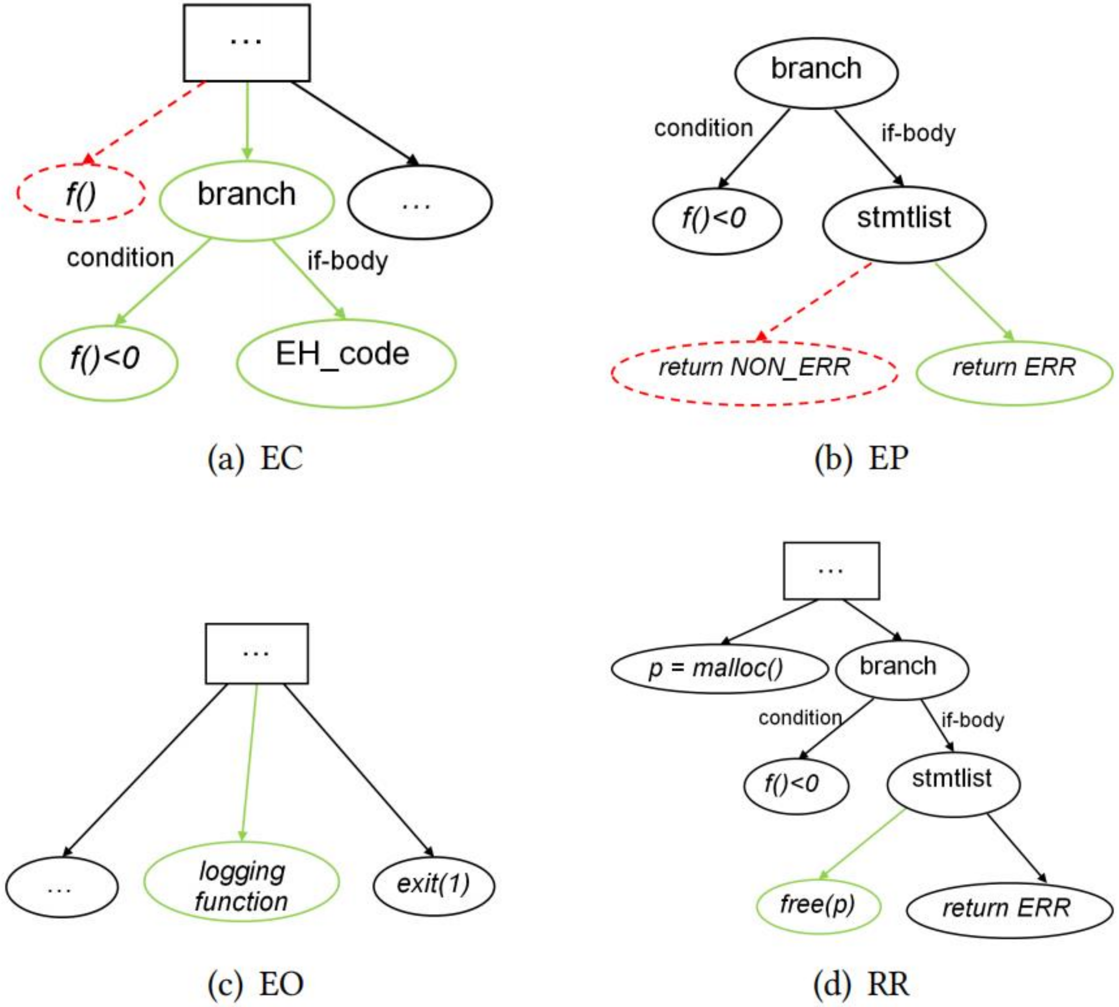
General APR approches highly rely on tests to verfiy the generated patches

What's new for EH Bugs?

➢ **Fixing EH bugs**

- Template based[14]

  ✓Replicate **near-by** EH code snippets



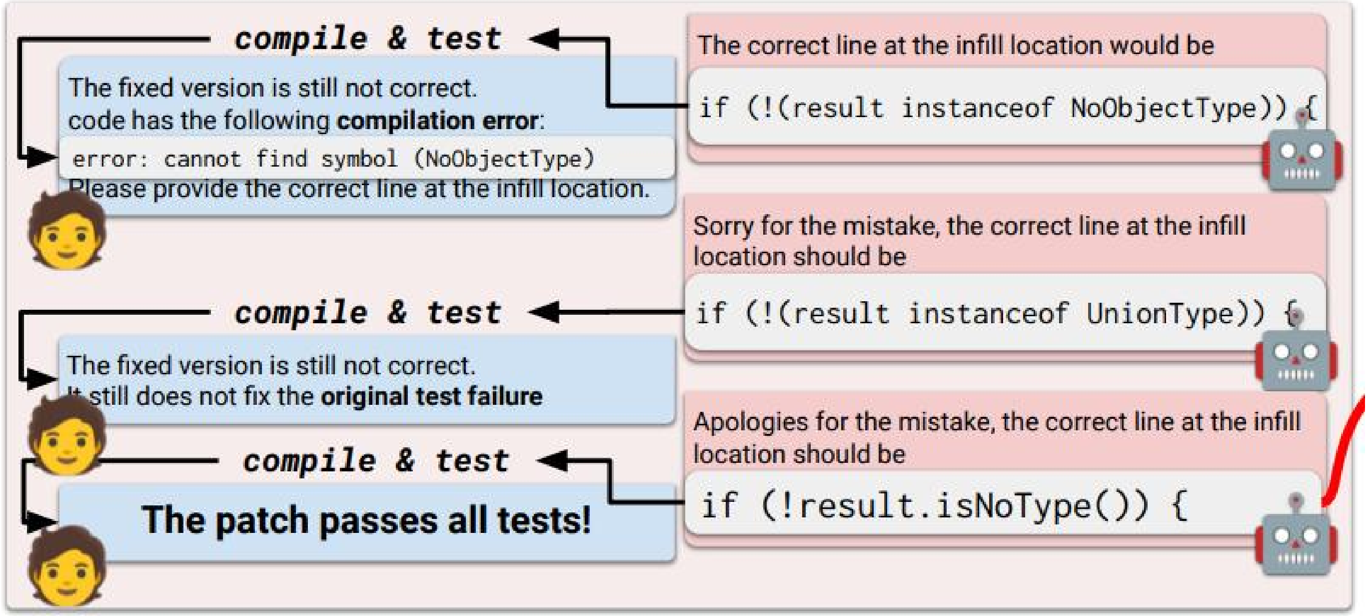
[14]. Tian Y, Ray B. Automatically diagnosing and repairing error handling bugs in c. FSE. 2017

What's new for EH Bugs?

➢ Fixing EH bugs

- LLM based[11,12]
  - ✓Iterative **compile&test** feedback

Are we good?



[11]. Xia C S, et al. Keep the Conversation Going: Fixing 162 out of 337 bugs for $0.42 each using ChatGPT. ISSTA. 2024
[12]. Chen Y, et al. When Large Language Models Confront Repository-Level Automatic Program Repair: How Well They Done. ICSE. 2024

Limitations

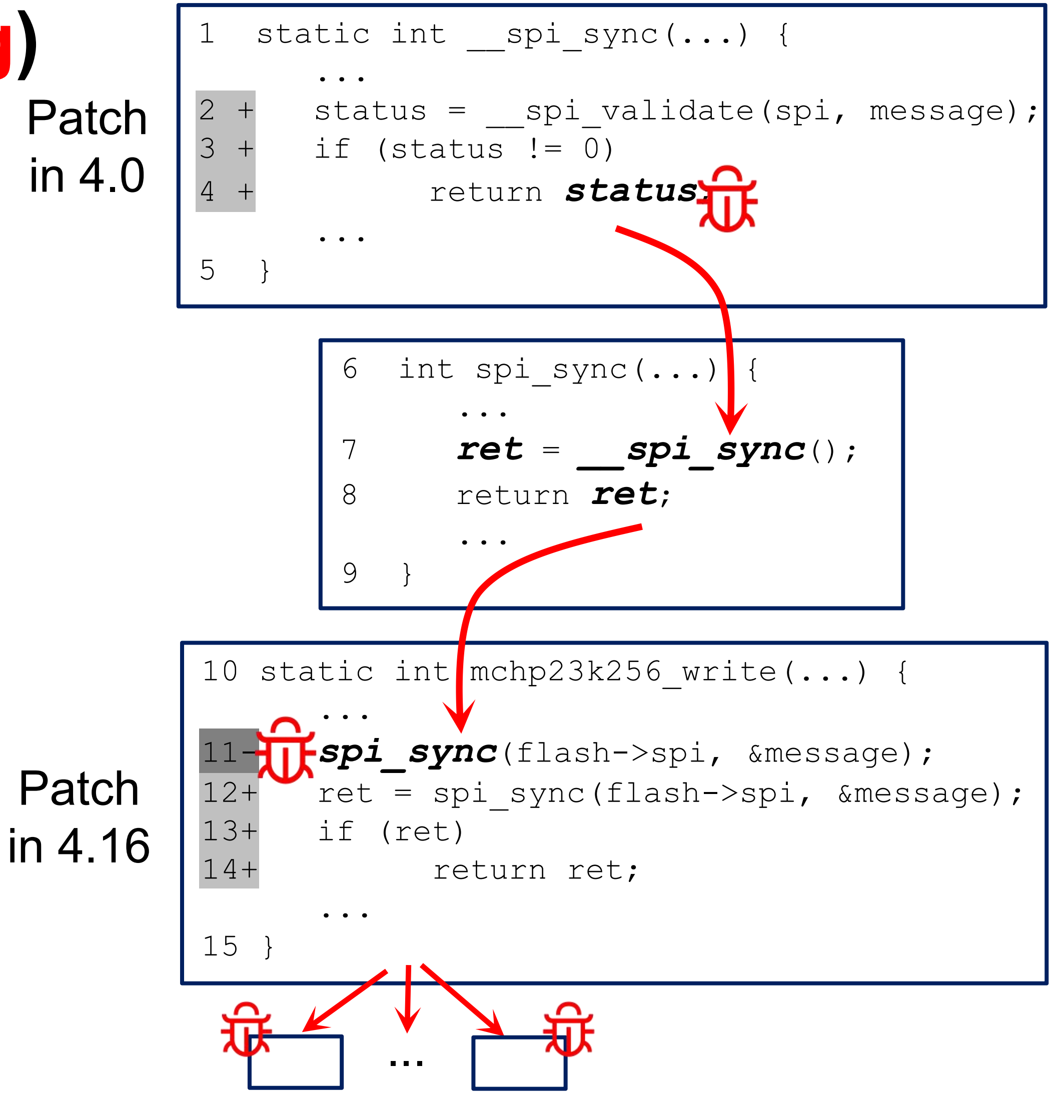
- **EH bugs may propagated (PEH Bug)**
  - Affecting multiple functions
  - Repairs may introduce new bugs

Studied 11 software systems from 9 domains

✓ **41.9% (367/876)** of errors are **propagated**

✓ Affecting an average of **16.7 functions**

✓ Requiring **44.1 days** to repair



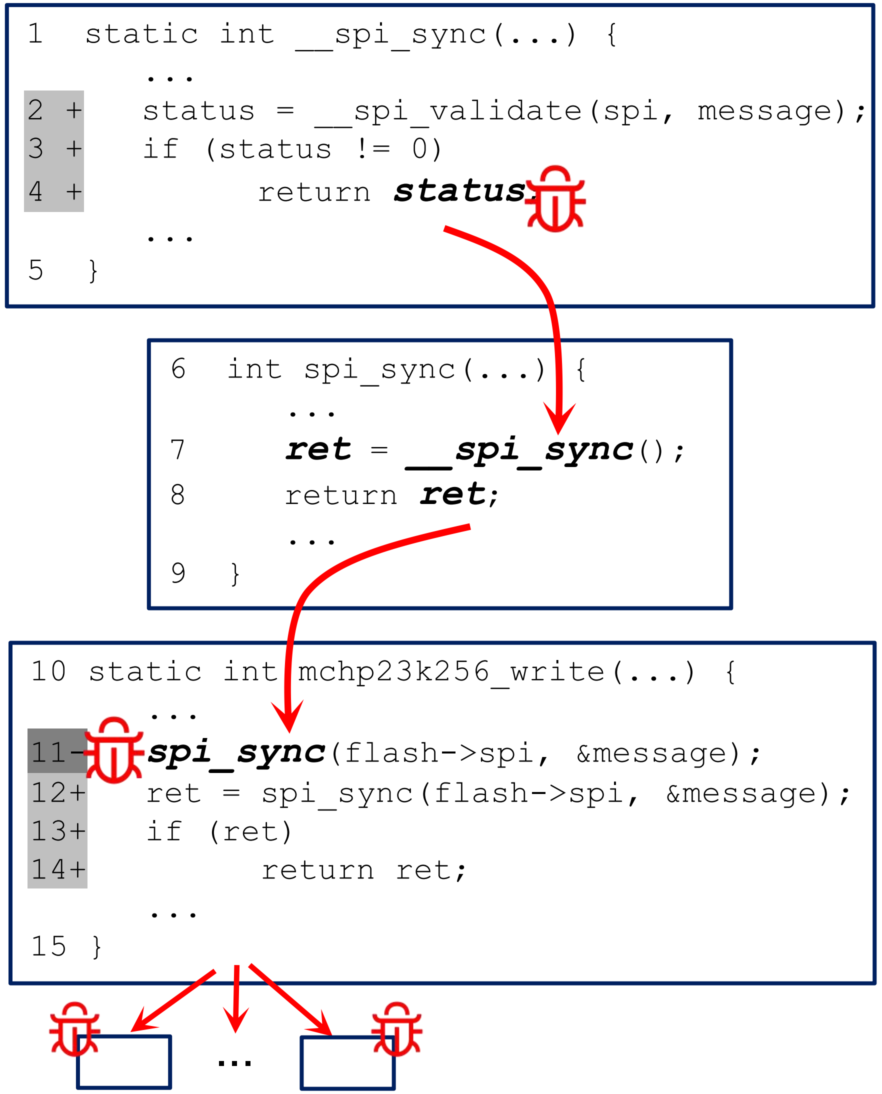
Limitations

- **EH bugs may propagated (PEH Bug)**
  - Affecting multiple functions
  - Repairs may introduce new bugs

Existing work mainly focus on individual functions (intra-procedure)

Patch in 4.0

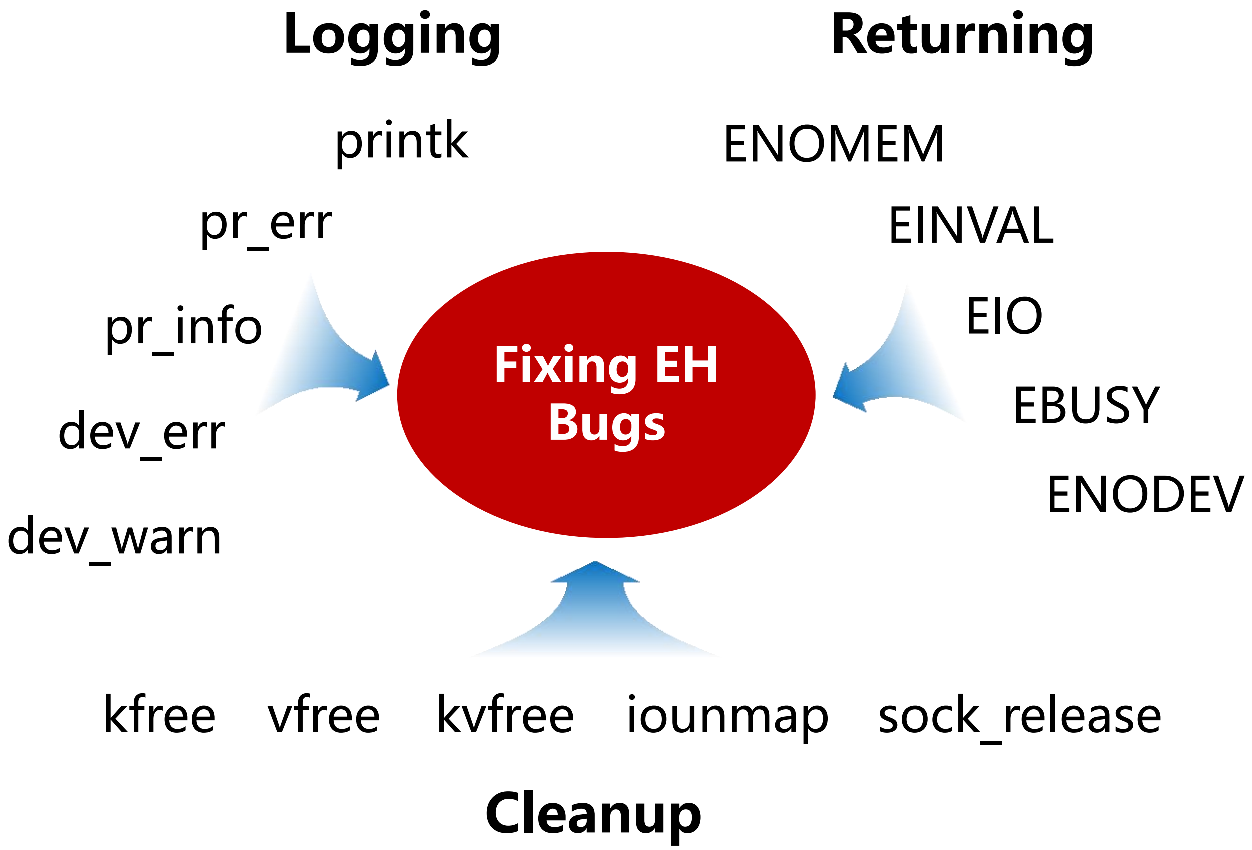
Patch in 4.16



Limitations

- **Diverse error-handling strategies**
  - Error-handling **strategies vary** across different functions
  - **Diverse optional** handling actions

Replicate near-by EH code can't handle the divergency



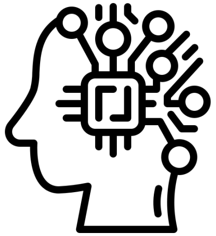
Insight

- Resolving error-handling bugs requires a **step-by-step fix along the propagation path**

Insight

➢ **Resolving error-handling bugs requires a step-by-step fix along the propagation path**

- Semantic understanding capability



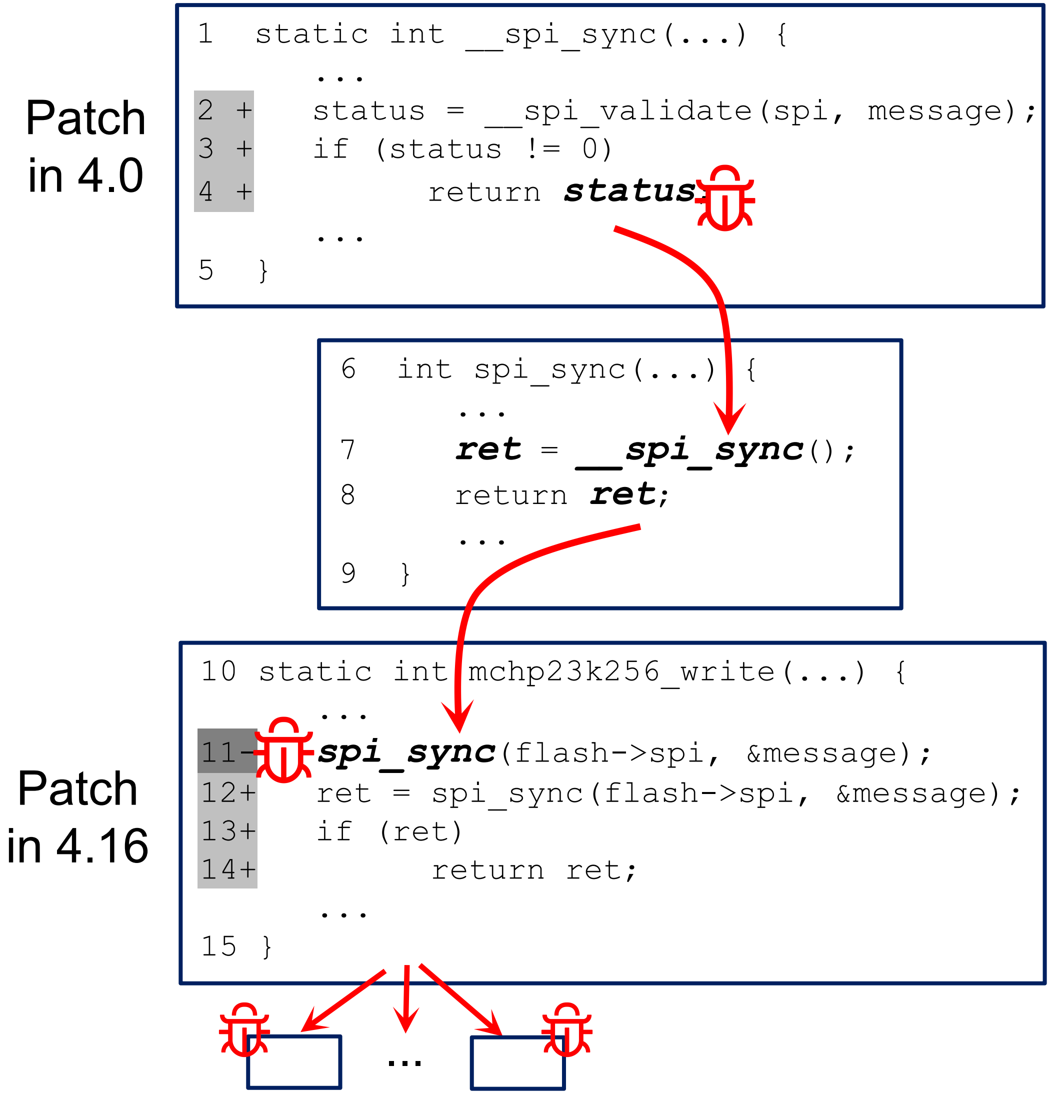
LLM

Repair ⇄ Propagation

Understanding and Fixing PEH Bugs

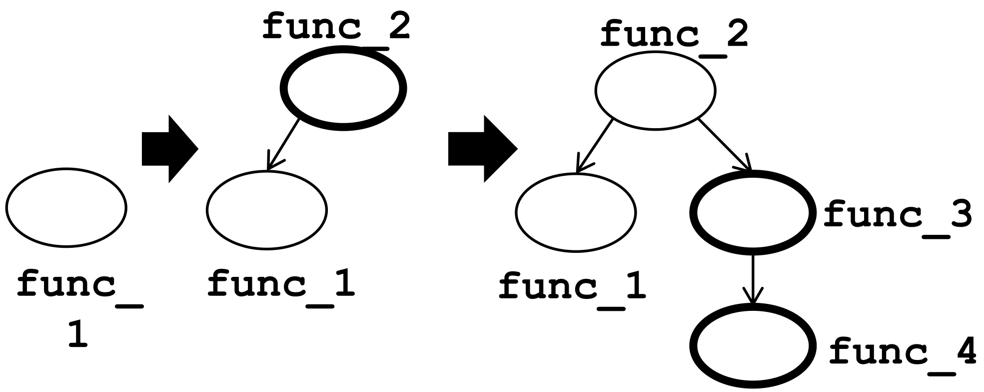
➢ Challenges (1/3)

- Tracing the propagation path

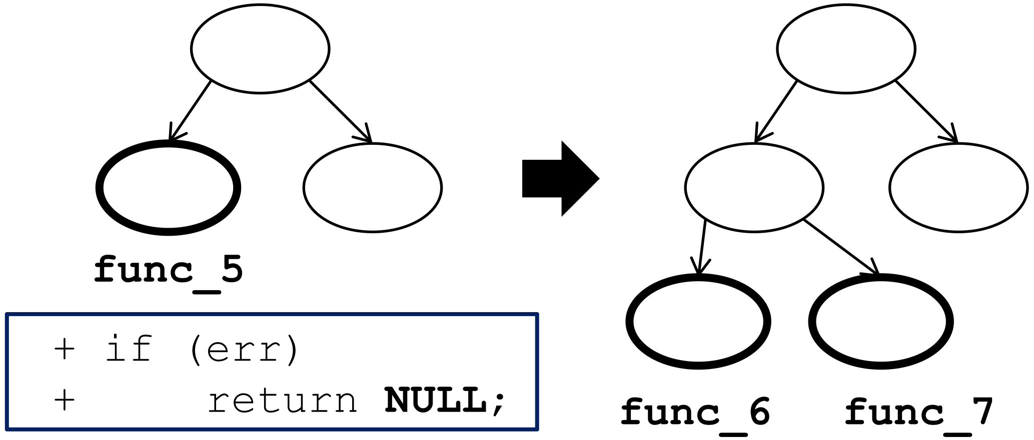


Understanding and Fixing PEH Bugs

- **Challenges (1/3)**
  - Tracing the propagation path
    - ✓Studied **11** software systems from **9** domains
      - **Return values (85.7%)**
      - **Parameters (12.3%)**



Path Construction

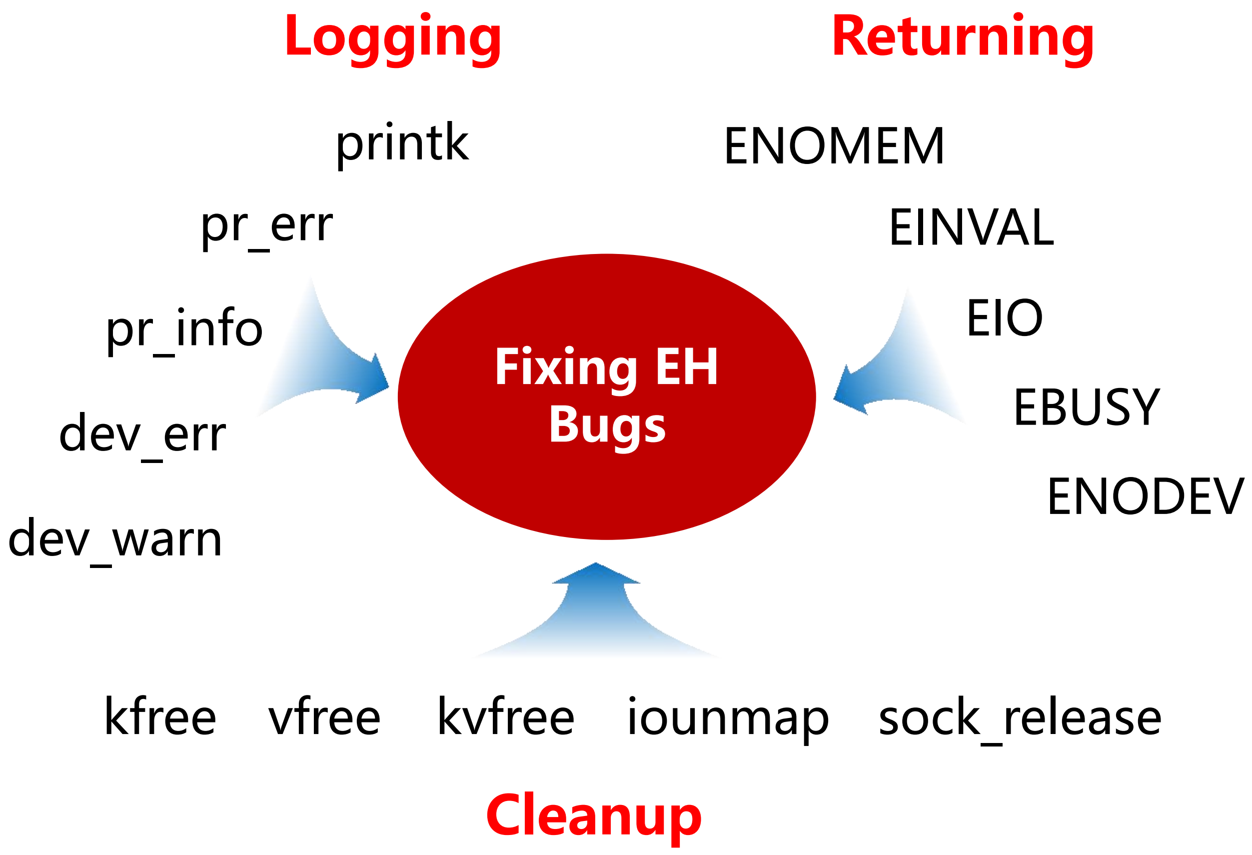


Path Extension

Understanding and Fixing PEH Bugs

➢ Challenges（2/3）

- Selecting proper handling **strategies** and **actions**



Understanding and Fixing PEH Bugs

- **Challenges（2/3)**
  - Selecting proper handling **strategies** and **actions**
    - ✓Studied 11 software systems from 9 domains
      - ☐**Exisiting EH code snippets** can guide the selection：
        - ■In the **same file**
        - ■In the **same propagation path**
        - ■Corresponding to **same specific resources**

Understanding and Fixing PEH Bugs

➢ Challenges (3/3)

- Validating the generated patch
  - ✓Lack of test cases

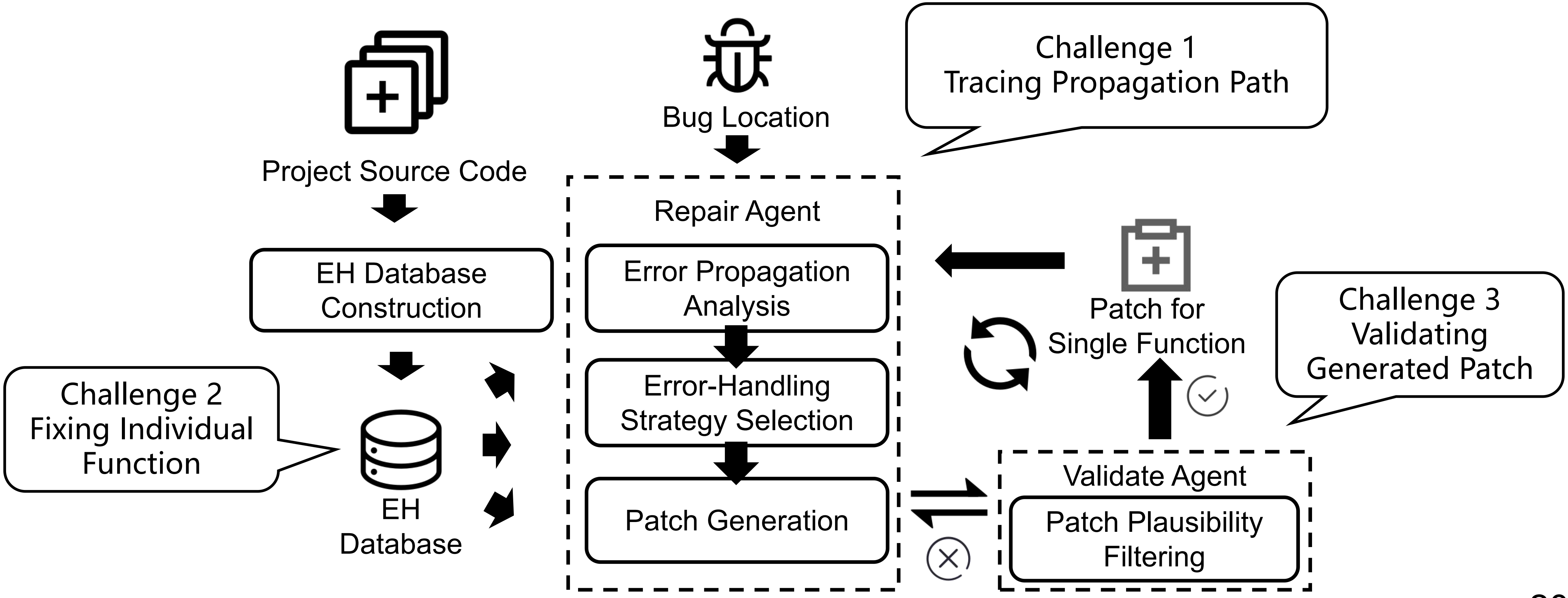
Understanding and Fixing PEH Bugs

- **Challenges (3/3)**
  - Validating the generated patch
    - ✓Collected 600 incorrect patches generated by LLMs
      - ☐Non-existent return values (52.8%)
      - ☐Non-existent cleanup functions (43.3%)
      - ☐Irrelevant modifications (43.1%)
    - ✓Validate generated patches
      - ☐Static analysis
      - ☐LLMs

Workflow

➢ APR approach for PEH bug：EH-Fixer

- Leveraging “Propagation–Repair”



Design

➢ Construction of EH Database

```
static int max1111_read(...) {
    ...
    err = spi_sync(data->spi, &data->msg);
    if (err < 0) {
       dev_err(...,"spi_sync failed",...);
       mutex_unlock(&data->drvdata_lock);
       return err;
    }
     ...
}
```

Source Code

Semantics and Dependencies

1)Summary: "This function reads data from the MAX1111 ADC device for a specified channel. It locks a mutex, configures the transmission buffer, sends a SPI command, and retrieves the result. If the response is valid, it processes and returns the data; otherwise, it returns an error code."

2)CallGraph, Data/Control Dependencies Graph

Available Action

3)Cleanup: (max1111_data->drvdata_lock, mutex_unlock)

4)Logging: ("/drivers/hwmon", dev_err)

5)Return: ("/drivers/hwmon", "-EINVAL")

Error Impact & Action set

6)<"SPI transmission failure", ("Logging", "Resource cleanup", "Early stop")>

7)<"Cannot pass input data for caller functions", ("Propagate error")>

EH Dataset

Design

➤ Prompt

- Instruction
- Example
- Constraints
- Contextual Information

```
Instruction
    ...
    1. Analysis error Impact:
    ...
    2. Predict Handling Actions:
    ...
    3. Generate Patch:
    ...
Example
    Examples for In-Context Learning
    ...
Constraints
    Output format
Contextual Information
    Source Code
    Relation Pairs
    Available Actions
    ...
```

Prompt Structure of the Repair Agent

Experiment

➢ Dataset

- 10 software in 8 domains
  - ✓**89 historical** PEH bugs
- Comparative approaches
  - ✓Template-based
    - ErrDoc (FSE 17)
  - ✓LLM-based
    - ChatRepair (ISSTA 24)
    - RLCE (ICSE 24)

Domain	Name	PEH Bugs
Operating System	Linux Kernel	44
Networking	OpenVPN	11
Developer Tools	ESP-IDF	7
	BPF Compiler Collection	4
Database	Redis	6
Window Manager	Sway	5
	Mutter	2
Emulator	iSH	4
Media	HandBrake	4
Data Transfer Tool	Curl	2

Experiment

➢ Performance on Real-World PEH Bugs

- Fixing new PEH bugs
  - ✓Fixed **9 new** PEH bugs in Linux Kernel
    - **2** confirmed by developers， others are pending
- Fixing historical PEH bugs
  - ✓Repair rate of **83.1%** (74/89)

Experiment

➢ Comparison with the State-of-the-art

- **48.6% (36/74)** PEH bugs fixed by EH-Fixer **cannot be fixed** by all comparative approaches

Table 3. Comparison in real-world bugs.

	Precision	Repair Rate
ChatRepair	13.7%(61/445)	14.6%(13/89)
ErrDoc	28.3%(126/445)	30.3%(27/89)
RLCE	38.0%(169/445)	42.7%(38/89)
EH-Fixer	**72.1%(321/445)**	**83.1%(74/89)**

Summary

➢ Contributions

- Studied 153 PEH bugs from 11 software systems, revealing the limitation of existing approaches
- Proposed EH-Fixer to step-by step repair PEH bugs automatically
- Built a dataset of 89 real-world PEH bugs for future study, and evaluate the effectiveness of EH-Fixer

https://github.com/EH-Fixer/EH-Fixer

Thanks for your attention