CODE HEALER

The Knowledge Hub Universities
Coventry University
School of Computing

Ramy Naiem

Project Advisors: Dr. Haitham Ghalwash & Eng. May Shalab

Ethical hacking and Cybersecurity Course

Code Healer
School of Computing

Problem Statement

Fuzzing, which systematically tests software with random data to identify security flaws, is labor-intensive. Our solution offers a more automated, efficient approach for contemporary software complexities.

Proposed Solutions

- Code Healer Introduction: AI software automates error handling and vulnerability detection with smart AI suggestions.

- Code Healer: Streamlines bug detection and resolution with an enhanced fuzzing process, reducing manual effort.

Specifications

- Automated Debugging: Leverages AI API's for efficient error resolution and security.

  Gemini

- Design Constraints: Seamlessly fits into current development tools and workflows.

Design

Seed Selector: Chooses seeds based on coverage and novelty		
Seed Selector: Chooses seeds based on coverage and novelty		
Seed Selector: Chooses seeds based on coverage and novelty		
Seed Selector: Chooses seeds based on coverage and novelty		
Seed Selector: Chooses seeds based on coverage and novelty		
Seed Selector: Chooses seeds based on coverage and novelty		

- Python-Based Architecture: Code Healer is developed in Python.

- Seed Generation: Code Healer autonomously generates seeds, streamlining the fuzzing process

- AFL++ Integration: Utilizes the advanced capabilities of AFL++ to enhance the fuzzing efficiency and coverage.

Results and Validation

Initial Results: Code Healer proves more efficient at fixing software vulnerabilities compared to traditional methods.

Conclusions and Future Work

our Product Marks a new era in software testing, enhances security and reliability, and foreshadows AI's role in cybersecurity advancements. Api based smart suggestion during coding. Supporting more Fuzzers