Vulnerable by Design: Educational Smart Contracts for Teaching Blockchain Security through Capture the Flag (CTF) Challenges

Problem Statement and Goals

Problem Statement:

Considering the widespread use of blockchains and the persistence of critical smart contract vulnerabilities, there is a need for practical and hands-on educational resources that help researchers, practitioners, and students identify and mitigate security flaws in blockchain applications.

Goals:

- Create vulnerable smart contract examples for use as educational resources.
- Design a hands-on smart contract security curriculum that leverages CTFs.
- Evaluate learning outcomes through a small pilot study to then share the curriculum within the open-source information security community.

Approach

Vulnerable Smart Contract CTF Challenges:

Intentionally vulnerable smart contracts are provided as CTF challenges, allowing students to explore simulated blockchain security flaws and immediately apply exploitation to understand malicious actors' common attack vectors.

CTF-Style Learning Modules:

CTF challenges built through Challenge Manager (cmgr) guide students through identifying vulnerabilities, exploiting them, and then implementing secure fixes in a low-risk environment.

Guided Walkthroughs, Demos, & Assesments:

The curriculum begins and ends with a self-assessment quiz, allowing students to evaluate their progress and growth, while each challenge is guided and demonstrated (via an included lecture video) to build confidence and promote problem-solving.

Results

- Developed a modular curriculum using intentionally vulnerable smart contracts to teach common blockchain vulnerabilities.
- Created a public repository of vulnerable Solidity contracts with learning objectives, exploit walkthroughs, pre- and post-quizzes, and mitigation guidance.
- Designed directly applicable CTF challenges to reinforce secure smart contract development through hands-on practice as a simulated attacker.

Future Work:

- Conduct a pilot study (3-5 participants) and observe learning outcomes using the pre- and post-quizzes, aggregated CTF challenge data, and participant feedback.
- Explore broader classroom deployment into existing secure coding and information security courses in primary and higher education.

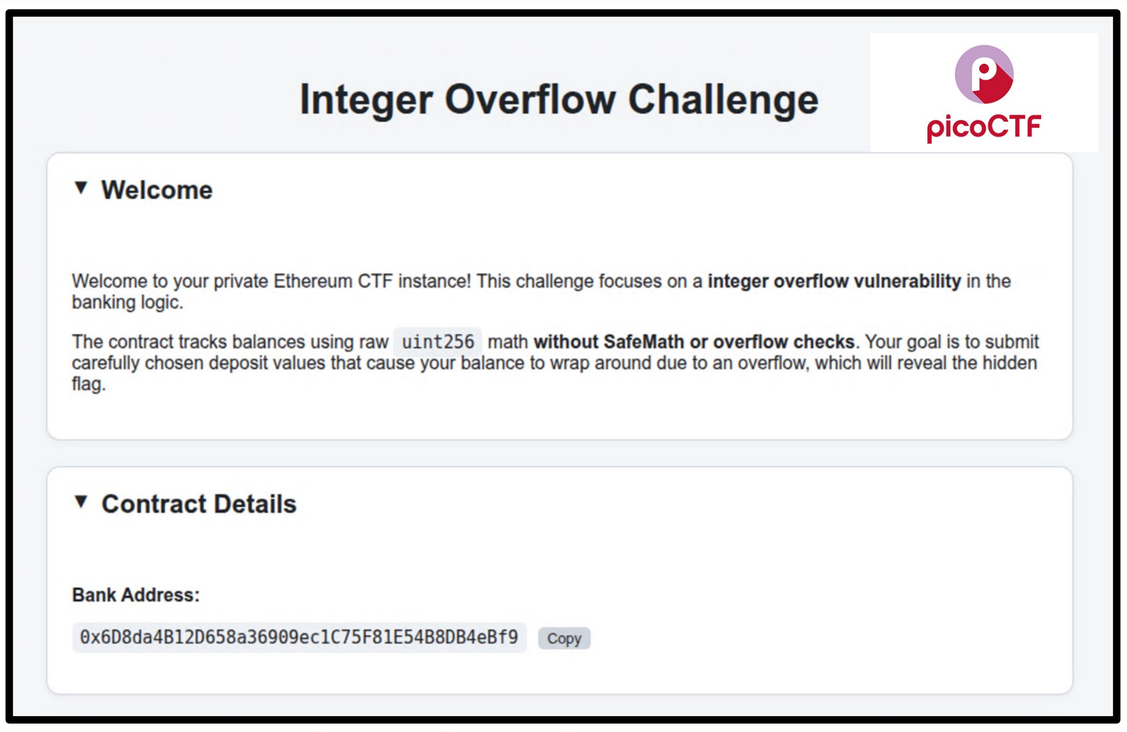

Figure 1: Example Problem (Integer Overflow)

```
4   contract VulnBank {
5       mapping(address => uint) public balances;
6       address public owner;
7       address public target;
8       string private flag;
9       bool public revealed;
10
11      event Deposit(address indexed who, uint amount);
12      event Withdraw(address indexed who, uint amount);
13      event FlagRevealed(string flag);
14
```

Figure 2: Example Vulnerable CTF Smart Contract (Solidity)

Logan O'Brien
Carnegie Mellon University

Carnegie Mellon University

Vulnerable by Design: Educational Smart Contracts for Teaching Blockchain Security through Capture the Flag (CTF) Challenges

Abstract

Blockchain technologies and smart contracts are increasingly implemented and used in real-world systems, but they still face many of the same vulnerabilities found in traditional software. Public exploits have shown that insecure smart contract development can lead to significant financial risks. Despite this, there is a lack of hands-on educational resources for learning smart contract security, resulting in practitioners and soon-to-be developers relying on automated analysis tools that are not designed for learning and are known to miss common vulnerabilities [1].

This research aims to teach secure smart contract development by developing a curriculum that teaches students to secure intentionally vulnerable smart contracts through completing CTF challenges. The curriculum is stored in a GitHub repository containing vulnerable Solidity contracts. Each of these contracts is paired with learning objectives, exploit walkthroughs, pre- and post-quizzes, and mitigation strategies. This allows learners to progress from simply using an automated tool to understanding potential exploitations from an attack and how to fix those vulnerabilities. Common vulnerabilities such as integer overflow, reentrancy, unchecked external calls, and confused deputy scenarios are emphasized to teach the foundations of common security flaws in smart contracts [2].

The curriculum integrates CTF-style challenges that encourage active problem-solving and experimentation in a controlled environment. In addition to developing smart contract technical skills, the curriculum encourages students to think critically when applying secure coding practices found in traditional software to their smart contract development [2].

Future work includes conducting a pilot study of 3-5 graduate level students to evaluate the effectiveness of the curriculum, the learners' comprehension of the material, and their confidence in developing secure smart contracts without the use of an automated tool. The results of this study will inform further refinement to then adopt the curriculum into a larger study in university and continued education courses.

References:

[1] A. Ghaleb and K. Pattabiraman, "How effective are smart contract analysis tools? evaluating smart contract static analysis tools using bug injection," in Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis, in ISSTA 2020. New York, NY, USA: Association for Computing Machinery, Jul. 2020, pp. 415–427. doi: 10.1145/3395363.3397385.

[2] G. Iuliano and D. D. Nucci, "Smart Contract Vulnerabilities, Tools, and Benchmarks: An Updated Systematic Literature Review," May 26, 2025, arXiv: arXiv:2412.01719. doi: 10.48550/arXiv.2412.01719.

Logan O'Brien
Carnegie Mellon University

Carnegie Mellon University