# School of Informatics


## Research Methods In Security, Privacy, and Trust
## Detecting Ethereum Smart Contract Security Loopholes

**2187344**
**November 2021**

**Abstract**

Date: Thursday 18th November, 2021

**Supervisor:** Lorenzo Martinico

# 1 Introduction

Ethereum is a general-purpose Blockchain, providing a platform to run decentralized applications executing code called Smart Contracts. Smart Contracts mainly manage valuable digital assets, and thus securing them is a top priority. Yet, it is typical for any piece of code to have bugs. However, Smart Contract bug fixes on the fly are not feasible since blockchain is an immutable append-only data structure. Hence, detecting code bugs and vulnerabilities before deploying Smart Contracts is vital.

In recent years there have been several attempts to create practical vulnerability detection tools for Smart Contracts. This is a niche topic, and there are several schools of thought when it comes to detecting security loopholes. This literature review aims to probe associated studies, focusing on different methods of detecting Smart Contract vulnerabilities, comparing different approaches, and taxonomizing existing frameworks.

...

In this literature review, I explore the most prominent research attempts towards creating an effective vulnerability detection tool for Ethereum Smart Contracts. The method used to filter relevant studies comprised multiple steps. Initially, I used trustworthy academic search engines such as Google Scholar and IEEE Explorer to retrieve a few papers and have them in a paper pool. The main selection criteria were the number of citations in combination with the paper release year. Highly cited articles with recent release dates usually include important research outcomes and can be the cornerstone for future work. After isolating a few reputable studies, I used a graph representation tool [1] that links relevant papers. This tool allowed me to identify remarkable research papers rapidly. Afterward, I manually inspected the search results and included the most reliable in my paper pool. Iterating the procedure mentioned above, I converged into a set of papers to form this literature review.

...

# 2 Literature Review

## 2.1 White/Grey-box Fuzzing

### 2.1.1 Static Smart Contract Analysis

Smart Check [2]
Slither [3]
MadMax [4]
Zeus [5]

### 2.1.2 Dynamic Smart Contract Analysis

Oyente [6]
Manticore [7]

### 2.2 Black-box Fuzzing

Contract Fuzzer [8]
ReGaurd [9]

### 2.3 Analysing Smart Contracts Using Formal Verification

Securify [10]

### 2.4 Analysing Smart Contracts Using Machine Learning

SoliAudit [11]

## 3 Summary & Conclusion

## References


[1] Find and explore academic papers. `https://www.connectedpapers.com/`. Accessed: 2021-11-04.

[2] Sergei Tikhomirov, Ekaterina Voskresenskaya, Ivan Ivanitskiy, Ramil Takhaviev, Evgeny Marchenko, and Yaroslav Alexandrov. Smartcheck: Static analysis of ethereum smart contracts. In *Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain*, pages 9–16, 2018.

[3] Josselin Feist, Gustavo Grieco, and Alex Groce. Slither: a static analysis framework for smart contracts. In *2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, pages 8–15. IEEE, 2019.

[4] Neville Grech, Michael Kong, Anton Jurisevic, Lexi Brent, Bernhard Scholz, and Yannis Smaragdakis. Madmax: Surviving out-of-gas conditions in ethereum smart contracts. *Proceedings of the ACM on Programming Languages*, 2(OOPSLA):1–27, 2018.

[5] Sukrit Kalra, Seep Goel, Mohan Dhawan, and Subodh Sharma. Zeus: Analyzing safety of smart contracts. In *Ndss*, pages 1–12, 2018.

[6] Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor. Making smart contracts smarter. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 254–269, 2016.

[7] Mark Mossberg, Felipe Manzano, Eric Hennenfent, Alex Groce, Gustavo Grieco, Josselin Feist, Trent Brunson, and Artem Dinaburg. Manticore: A user-friendly symbolic execution framework for binaries and smart contracts. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 1186–1189. IEEE, 2019.

[8] Bo Jiang, Ye Liu, and WK Chan. Contractfuzzer: Fuzzing smart contracts for vulnerability detection. In *2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 259–269. IEEE, 2018.

[9] Chao Liu, Han Liu, Zhao Cao, Zhong Chen, Bangdao Chen, and Bill Roscoe. Reguard: finding reentrancy bugs in smart contracts. In *2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion)*, pages 65–68. IEEE, 2018.

[10] Petar Tsankov, Andrei Dan, Dana Drachsler-Cohen, Arthur Gervais, Florian Buenzli, and Martin Vechev. Securify: Practical security analysis of smart contracts. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 67–82, 2018.

[11] Jian-Wei Liao, Tsung-Ta Tsai, Chia-Kang He, and Chin-Wei Tien. Soliaudit: Smart contract vulnerability assessment based on machine learning and fuzz testing. In *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, pages 458–465. IEEE, 2019.

[12] Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. A survey of attacks on ethereum smart contracts (sok). In *International conference on principles of security and trust*, pages 164–186. Springer, 2017.

[13] Imran Ashraf, Xiaoxue Ma, Bo Jiang, and Wing Kwong Chan. Gasfuzzer: Fuzzing ethereum smart contract binaries to expose gas-oriented exception security vulnerabilities. *IEEE Access*, 8:99552–99564, 2020.

[14] Qingzhao Zhang, Yizhuo Wang, Juanru Li, and Siqi Ma. Ethploit: From fuzzing to efficient exploit generation against smart contracts. In *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 116–126. IEEE, 2020.

[15] Karthikeyan Bhargavan, Antoine Delignat-Lavaud, Cédric Fournet, Anitha Gollamudi, Georges Gonthier, Nadim Kobeissi, Natalia Kulatova, Aseem Rastogi, Thomas Sibut-Pinote, Nikhil Swamy, et al. Formal verification of smart contracts: Short paper. In *Proceedings of the 2016 ACM workshop on programming languages and analysis for security*, pages 91–96, 2016.

[16] Ivica Nikolić, Aashish Kolluri, Ilya Sergey, Prateek Saxena, and Aquinas Hobor. Finding the greedy, prodigal, and suicidal contracts at scale. In *Proceedings of the 34th Annual Computer Security Applications Conference*, pages 653–663, 2018.

[17] Johannes Krupp and Christian Rossow. teether: Gnawing at ethereum to automatically exploit smart contracts. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 1317–1333, 2018.