An Overview of the Research in the Security Issues of Ethereum Ecosystem

Saurav Taneja, Ravi Shukla,
IIIT Bangalore

***Abstract*—Blockchain is a revolutionary technology that enables users to communicate in a trust-less manner. It enables users to store data globally on thousands of computers in an immutable format and empowers users to deploy small pieces of programs known as smart contracts. The blockchain-based smart contract enables auto enforcement of the agreed terms between two untrusted parties. There are several security vulnerabilities in Ethereum blockchain-based smart contracts, due to which sometimes it does not behave as intended. Because a smart contract can hold millions of dollars as cryptocurrency, so these security vulnerabilities can lead to losses. We present a review of the security issues in the Ethereum ecosystem.**

I. Survey Methodology

In this survey our main focus is on the security issues in this Ethereum ecosystem. We highlight the the research papers that deals with different kinds of economic attacks and market manipulation strategies, different kinds of vulnerabilities and their detections. For that we surveyed, the last five years of research papers oublished in top tier security venues.

II. Survey

Here we present a survey of the attacks and vulnerabilities in the Ethereum ecosystem,

Economic attacks: Blockchain has been the target of many economic attacks. In Ethereum, miners are incentivized through *gas* for their hard work. Uncommitted transactions and their gas bids are visible to other network participants. Therefore, an attacker can get their transactions mined earlier that the victim transaction by paying higher gas price. This is known as front-running [20]. They are the first to introduce a frontrunning taxonomy for blockchains. In [18], the authors show how arbitrage bots front-run transactions to generate revenues. Bonneau [11] is the first to study bribery attacks in the context of Bitcoin-style consensus. In Sandwich attacks, both front- and back-running happens. Flashloans allow a borrower immediate access to a large amount of funds without offering any collateral, under the condition that the loan needs to be repaid in the same transaction. Qin et.al [40] analyzed how flashloans have been used to execute arbitrage and oracle manipulation attacks, and they presented a constrained optimization framework to cleverly choose the attack parameters that maximize the profit. In [54], the authors propose how to generate profit through complex transactions. [31], [25], [52], [19] also investigated pump-and-dump schemes, a price manipulation schemes, security issues and market manipulation happening [4], [1] in the NFT ecosystem.

Vulnerabilities: Vulnerability detection is an old area research. Prior research in vulnerability detection spanned across different domains—vulnerability detection in iot devices, user-space applications, linux kernel etc [46], [13], [45], [55], [39], [9], [22], [36], [41], [14], [51], [15], [42], [17], [35], [34], [43]. In the recent years, research have been focusing on vulnerabilities in Ethereum ecosystem, specifically in smart contracts since they have been very popular and building block of many Decentralized protocols. However, a vulnerability in a smart contract can result in million of dollar in losses. One such attach happened in 2016 [8]. Since then there has been several such attacks [5], [7], [10].

Static analysis tools [48], [26], [30], [47], [21] have been developed to detect specific vulnerabilities in smart contracts. Madmax [26] uses a logic-based paradigm for gas-focused vulnerabilities. Securify [48] checks for compliance and violation signatures by checking control and data flows. Zeus [30] employs a static analysis to instrument the contract code with policy assertions, which are then lifted to `LLVM IR`. Slither's [21] analysis is scoped within a single function. Similarly, Smartcheck [47] uses `XML` as its intermediate representation, and issues `XPath` queries to find violation patterns.

Symbolic execution based tools [3], [33], [24], [23], [6] explore the state-space of the contract. Ethbmc [24], EVM transactions as state transitions. Teether [32] generates constraints along a critical path having attacker-controlled instructions. Maian [38] performs a symbolic analysis followed by a concrete validation phase to verify certain safety and liveness properties. All these tools encode a path as a set of constraints, and then ask the constraint solver to generate a counter-example that both violates (bug) a pre-defined security property, and act as a witness (exploit) for the same. Since enumerating all the paths in the contracts translates to an unbounded search space, These tools make unsound choices to enable scalable path exploration. Smartcopy [23] proposes a summary-based symbolic evaluation technique that attempts to reduce the number of paths without sacrificing the precision. Instead of solely relying on symbolic evaluation, Sailfish [12] uses the combination of static analysis and symbolic execution to detect reentrancy and tod bugs.

Ethertrust [27], based on formal verification, translates the semantics of `EVM` bytecode to a set of Horn clauses. Although providing strong security guarantees and sound results, such techniques require manual effort to encode the semantics of the execution environment.

Sereum [44] and Soda [16] perform run-time checks, and Txspector [53] performs a post-mortem analysis of transactions. Ecfchecker [28] checks whether a contract is callback-free. Dynamic analysis tools [29], [49], [50], [2], [37], [29], [49], [50], [2], [37] rely on test oracles to detect violations. Echidna [2] is a grammar-based fuzzer that generates inputs conforming to the contract ABI. Bran [50] combines the power of static analysis to augment greybox fuzzing.

III. Conclusion

pass

References

[1] Behaviors in the nft ecosystem that we hope will decrease in 2020. https://nonfungible.com/blog/bad-behaviors-nft-blockchain.

[2] Echidna. https://github.com/crytic/echidna. [accessed 07/27/2020].

[3] Mythril. https://github.com/ConsenSys/mythril.

[4] What is "wash trading" and why is it negative for non-fungible tokens? https://nonfungible.com/blog/wash-trading-and-why-its-negative-for-non-fungible-tokens.

[5] Governmental's 1100 eth payout is stuck because it uses too much gas. https://tinyurl.com/y83dn2yf/, 2016. [accessed 01/09/2019].

[6] Manticore. https://github.com/trailofbits/manticore/, 2016.

[7] On the parity wallet multisig hack. https://tinyurl.com/yca83zsg/, 2017.

[8] Understanding the dao attack. https://tinyurl.com/yc3o8ffk/, 2017.

[9] Bernhard K. Aichernig, Edi Mukardin, and Andrea Pferscher. Learning-based fuzzing of iot message brokers. *2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST)*, pages 47–58, 2021.

[10] Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. A survey of attacks on ethereum smart contracts (sok). In *Principles of Security and Trust - 6th International Conference, POST*, 2017.

[11] Joseph Bonneau. Why buy when you can rent? - bribery attacks on bitcoin-style consensus. In Jeremy Clark, Sarah Meiklejohn, Peter Y. A. Ryan, Dan S. Wallach, Michael Brenner, and Kurt Rohloff, editors, *Financial Cryptography and Data Security - FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers*, volume 9604 of *Lecture Notes in Computer Science*, pages 19–26. Springer, 2016.

[12] P. Bose, D. Das, Y. Chen, Y. Feng, C. Kruegel, and G. Vigna. Sailfish: Vetting smart contract state-inconsistency bugs in seconds. In *2022 2022 IEEE Symposium on Security and Privacy (SP) (SP)*, pages 1235–1252, Los Alamitos, CA, USA, may 2022. IEEE Computer Society.

[13] Alexander Bulekov, Bandan Das, Stefan Hajnoczi, and Manuel Egele. M orphuzz : Bending (input) space to fuzz virtual devices. 2021.

[14] Jiongyi Chen, Wenrui Diao, Qingchuan Zhao, Chaoshun Zuo, Zhiqiang Lin, Xiaofeng Wang, W. Lau, Menghan Sun, Ronghai Yang, and Kehuan Zhang. Iotfuzzer: Discovering memory corruptions in iot through app-based fuzzing. In *NDSS*, 2018.

[15] Jiongyi Chen, Chaoshun Zuo, Wenrui Diao, Shuaike Dong, Qingchuan Zhao, Menghan Sun, Zhiqiang Lin, Yinqian Zhang, and Kehuan Zhang. Your iots are (not) mine: On the remote binding between iot devices and users. *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 222–233, 2019.

[16] Ting Chen, Rong Cao, Ting Li, Xiapu Luo, Yufei Zhang, Zhou Liao, Hang Zhu, Gang Chen, Zheyuan He, Xiaodong Lin, and Xiaosong Zhang. Soda: A generic online detection framework for smart contracts. In *NDSS*, 2020.

[17] Jake Corina, Aravind Machiry, Christopher Salls, Yan Shoshitaishvili, Shuang Hao, Christopher Kruegel, and Giovanni Vigna. Difuze: Interface aware fuzzing for kernel drivers. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[18] Philip Daian, Steven Goldfeder, T. Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and A. Juels. Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges. In *SP*, 2020.

[19] Dipanjan Das, Priyanka Bose, Nicola Ruaro, Christopher Kruegel, and Giovanni Vigna. Understanding security issues in the nft ecosystem. 2021.

[20] Shayan Eskandari, Seyedehmahsa Moosavi, and Jeremy Clark. Sok: Transparent dishonesty: Front-running attacks on blockchain. In Andrea Bracciali, Jeremy Clark, Federico Pintore, Peter B. Rønne, and Massimiliano Sala, editors, *Proc. Financial Cryptography and Data Security*, 2020.

[21] J. Feist, G. Grieco, and A. Groce. Slither: A static analysis framework for smart contracts. In *IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, 2019.

[22] Xiaotao Feng, Ruoxi Sun, Xiaogang Zhu, Minghui Xue, Sheng Wen, Dongxi Liu, Surya Nepal, and Yang Xiang. Snipuzz: Black-box fuzzing of iot firmware via message snippet inference. *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021.

[23] Yu Feng, Emina Torlak, and Rastislav Bodik. Precise attack synthesis for smart contracts. *arXiv preprint arXiv:1902.06067*, 2019.

[24] Joel Frank, Cornelius Aschermann, and Thorsten Holz. ETHBMC: A bounded model checker for smart contracts. In *29th USENIX Security Symposium (USENIX Security)*, 2020.

[25] Neil Gandal, JT Hamrick, Tyler Moore, and Tali Oberman. Price manipulation in the bitcoin ecosystem. *Journal of Monetary Economics*, 95, 01 2018.

[26] Neville Grech, Michael Kong, Anton Jurisevic, Lexi Brent, Bernhard Scholz, and Yannis Smaragdakis. Madmax: surviving out-of-gas conditions in ethereum smart contracts. In *Proc. International Conference on Object-Oriented Programming, Systems, Languages, and Applications*, 2018.

[27] Ilya Grishchenko, Matteo Maffei, and Clara Schneidewind. Foundations and tools for the static analysis of ethereum smart contracts. 2020.

[28] Shelly Grossman, Ittai Abraham, Guy Golan-Gueta, Yan Michalevsky, Noam Rinetzky, Mooly Sagiv, and Yoni Zohar. Online detection of effectively callback free objects with applications to smart contracts. In *Proc. Symposium on Principles of Programming Languages*, 2018.

[29] Bo Jiang, Ye Liu, and W. K. Chan. Contractfuzzer: fuzzing smart contracts for vulnerability detection. In *Proc. International Conference on Automated Software Engineering*, 2018.

[30] Sukrit Kalra, Seep Goel, Mohan Dhawan, and Subodh Sharma. ZEUS: analyzing safety of smart contracts. In *Proc. The Network and Distributed System Security Symposium*, 2018.

[31] Josh Kamps and Bennett Kleinberg. To the moon: defining and detecting cryptocurrency pump-and-dumps. *Crime Science*, 7(1):18, Nov 2018.

[32] Johannes Krupp and Christian Rossow. teether: Gnawing at ethereum to automatically exploit smart contracts. 2018.

[33] Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor. Making smart contracts smarter. In *Proc. Conference on Computer and Communications Security*, 2016.

[34] Aravind Machiry, Eric Gustafson, Chad Spensky, Christopher Salls, Nick Stephens, Ruoyu Wang, Antonio Bianchi, Yung Ryn Choe, Christopher Krügel, and Giovanni Vigna. Boomerang: Exploiting the semantic gap in trusted execution environments. In *NDSS*, 2017.

[35] Aravind Machiry, Chad Spensky, Jake Corina, Nick Stephens, Christopher Kruegel, and Giovanni Vigna. Dr. checker: A soundy analysis for linux kernel drivers. In *USENIX Security Symposium*, 2017.

[36] Roberto Natella. Stateafl: Greybox fuzzing for stateful network servers. *ArXiv*, abs/2110.06253, 2021.

[37] Tai Nguyen, Long Pham, Jun Sun, Yun Lin, and Minh Quang Tran. sfuzz: An efficient adaptive fuzzer for solidity smart contracts. In *Proc. International Conference on Software Engineering*, 2020.

[38] 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*, 2018.

[39] Yan Pan, Wei Lin, Liang Jiao, and Yuefei Zhu. Model-based grey-box fuzzing of network protocols. *Security and Communication Networks*, 2022.

[40] Kaihua Qin, Liying Zhou, B. Livshits, and Arthur Gervais. Attacking the defi ecosystem with flash loans for fun and profit. In *Proc. Financial Cryptography and Data Security*, 2021.

[41] Nilo Redini, Andrea Continella, Dipanjan Das, Giulio De Pasquale, Noah Spahn, Aravind Machiry, Antonio Bianchi, Christopher Kruegel, and Giovanni Vigna. Diane: Identifying fuzzing triggers in apps to generate under-constrained inputs for iot devices. *2021 IEEE Symposium on Security and Privacy (SP)*, pages 484–500, 2021.

[42] Nilo Redini, Aravind Machiry, Dipanjan Das, Yanick Fratantonio, Antonio Bianchi, Eric Gustafson, Yan Shoshitaishvili, Christopher Krügel,

and Giovanni Vigna. Bootstomp: On the security of bootloaders in mobile devices. In *USENIX Security Symposium*, 2017.

[43] Nilo Redini, Aravind Machiry, Ruoyu Wang, Chad Spensky, Andrea Continella, Yan Shoshitaishvili, Christopher Kruegel, and Giovanni Vigna. Karonte: Detecting insecure multi-binary interactions in embedded firmware. *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1544–1561, 2020.

[44] Michael Rodler, Wenting Li, Ghassan O. Karame, and Lucas Davi. Sereum: Protecting existing smart contracts against re-entrancy attacks. In *NDSS*, 2019.

[45] Sergej Schumilo, Cornelius Aschermann, Ali Reza Abbasi, Simon Wörner, and Thorsten Holz. Nyx: Greybox hypervisor fuzzing using fast snapshots and affine types. In *USENIX Security Symposium*, 2021.

[46] Dokyung Song, Felicitas Hetzelt, Dipanjan Das, Chad Spensky, Yeoul Na, Stijn Volckaert, Giovanni Vigna, Christopher Krügel, Jean-Pierre Seifert, and Michael Franz. Periscope: An effective probing and fuzzing framework for the hardware-os boundary. *Proceedings 2019 Network and Distributed System Security Symposium*, 2019.

[47] 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*, 2018.

[48] Petar Tsankov, Andrei Marian Dan, Dana Drachsler-Cohen, Arthur Gervais, Florian Bünzli, and Martin T. Vechev. Securify: Practical security analysis of smart contracts. In *Proc. Conference on Computer and Communications Security*, 2018.

[49] Valentin Wüstholz and Maria Christakis. Harvey: A greybox fuzzer for smart contracts. *ArXiv*, abs/1905.06944, 2019.

[50] Valentin Wüstholz and Maria Christakis. Targeted greybox fuzzing with static lookahead analysis. 2020.

[51] Fenghao Xu, Wenrui Diao, Zhou Li, Jiongyi Chen, and Kehuan Zhang. Badbluetooth: Breaking android security mechanisms via malicious bluetooth peripherals. *Proceedings 2019 Network and Distributed System Security Symposium*, 2019.

[52] Jiahua Xu and Benjamin Livshits. The anatomy of a cryptocurrency pump-and-dump scheme. In *Proc. USENIX Security Symposium*, 2019.

[53] Mengya Zhang, Xiaokuan Zhang, Yinqian Zhang, and Zhiqiang Lin. TXSPECTOR: Uncovering attacks in ethereum from transactions. In *29th USENIX Security Symposium (USENIX Security)*, 2020.

[54] Liying Zhou, Kaihua Qin, Antoine Cully, Benjamin Livshits, and Arthur Gervais. On the just-in-time discovery of profit-generating transactions in defi protocols. In *IEEE SP*, 2021.

[55] Xiaogang Zhu, Sheng Wen, Seyit Ahmet Çamtepe, and Yang Xiang. Fuzzing: A survey for roadmap. *ACM Computing Surveys (CSUR)*, 2022.