CONCLUSION

In this paper, we highlighted BLOCKEYE as an open platform to detect DeFi attacks on blockchain. Compared to existing analyzers for smart contracts, BLOCKEYE pro vides important capabilities to model dependency among DeFi projects and flag potential end-to-end attacks at real-time. The key insights behind BLOCKEYE are symbolic oracle analysis and pattern-based runtime transaction validation. We applied BLOCKEYE in several popular DeFi projects on Ethereum and managed to find potential attacks previously unreported.

In this SoK we have considered DeFi from two points of view, the DeFi Optimist and the DeFi Pessimist, and examined the workings of DeFi systematically and at length. First, we laid out the primitives for DeFi before categorizing DeFi protocols by the type of operation they provide. We examined the security challenges protocols are exposed to by making a distinction between technical and economic security risks. In so doing, we were able to systematize attacks that have been proposed in theory and/or occurred in practice into categories of attacks that either rely on an agent's ability to generate risk-free profits by exploiting the technical structure of a blockchain or to game the incentive structure of a protocol to obtain a profit at the expense of the protocol. Finally, we drew the attention to open research challenges that require a holistic understanding of both the technical and economic risks.

REFERENCE

[1] G. Wood, “Ethereum: A secure decentralised generalised transaction ledger,” Ethereum Project Yellow Paper, vol. 151, 2014.

[2] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, “Making smart contracts smarter,” in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016, pp. 254–269.

[3] P. Tsankov, A. Dan, D. D. Cohen, A. Gervais, F. Buenzli, and M. Vechev, “Securify: Practical security analysis of smart contracts,” arXiv preprint arXiv:1806.01143, 2018.
[4] C. Liu, H. Liu, Z. Cao, Z. Chen, B. Chen, and B. Roscoe, “Reguard: finding reentrancy bugs in smart contracts,” in ICSE (Companion). ACM, 2018, pp. 65–68.

[5] H. Liu, C. Liu, W. Zhao, Y. Jiang, and J. Sun, “S-gram: towards semantic-aware security auditing for ethereum smart contracts,” in ASE. ACM, 2018, pp. 814–819.

[6] Z. Yang, H. Liu, Y. Li, H. Zheng, L. Wang, and B. Chen, “Seraph: enabling cross-platform security analysis for evm and wasm smart con tracts,” in Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings, 2020, pp. 21–24.

[7] “Microsoft z3 smt solver,” https://z3.codeplex.com/, 2019.

[8] “Codefi inspect,” https://inspect.codefi.network/, 2020.

[9] T. Eiter, G. Gottlob, and H. Mannila, “Disjunctive datalog,” ACM Transactions on Database Systems (TODS), vol. 22, no. 3, pp. 364–418, 1997.

[10] A. Permenev, D. Dimitrov, P. Tsankov, D. Drachsler-Cohen, and M. Vechev, “Verx: Safety verification of smart contracts,” Security and Privacy, vol. 2020, 2019.

[11] K. Qin, L. Zhou, B. Livshits, and A. Gervais, “Attacking the defi ecosystem with flash loans for fun and profit,” arXiv preprint arXiv:2003.03810, 2020.

[12] J. Kamps and B. Kleinberg, “To the moon: defining and detecting cryptocurrency pump-and-dumps,” Crime Science, vol. 7, no. 1, p. 18, 2018.

[13] B. Liu and P. Szalachowski, “A first look into defi oracles,” arXiv preprint arXiv:2005.04377, 2020.

[14] L. Gudgeon, D. Perez, D. Harz, A. Gervais, and B. Livshits, “The decentralized financial crisis: Attacking defi,” arXiv preprint arXiv:2002.08099, 2020.