DOI:10.1145/3416259

To view the accompanying paper, visit doi.acm.org/10.1145/3416262

Technical Perspective
Analyzing Smart Contracts With MadMax

By Benjamin Livshits

SMART CONTRACTS PROVIDE a way to bring computational integrity to executing more or less general-purpose programs. While proposed a long time ago, they have only become popular with the advent of newer blockchain-based systems such as Ethereum with its associated Ethereum Virtual Machine (EVM), and several other similar systems. Smart contracts give the hope of being able to capture complex financial interactions and relationships with the help of executing code. As a result, we have seen a multitude of projects in areas as diverse as law and what is frequently referred to as decentralized finance (DeFi) based on smart contracts.

Somewhat notoriously, smart contracts, because they often directly manage financial transactions, wallets, and transfers, have been subject to vulnerability discovery, with many high-profile vulnerabilities, such as the DAO hack, a highly impactful exploit from mid-2016, where a hacker found a loophole in a smart contract that has led to the theft of about $70 million. This attack and some of the others have generated a great deal of interest in using static analysis and verification techniques to find bugs and vulnerabilities in contracts before they are allowed to be deployed onto a blockchain (since, after all, contracts are generally immutable as well, making bugs fairly difficult to fix after the fact).

MadMax focuses on a fairly specific aspect of smart contracts, that of *metering*. Metering is an approach to charge for contract execution, which plays the dual role of compensating blockchain participants and of preventing denial-of-service attacks. How to do metering properly is actually quite a hard problem. The EVM proposes a specific way to charge for contact execution, as specified in the Ethereum yellow paper. Gas is provided for the purpose of contract execution but if not enough gas is provisioned, contract state can be rolled back.

MadMax tackles gas-related vulnerabilities, which permit an attacker to force key contract functionality to run out of gas—effectively performing a permanent denial-of-service attack on the contract. As such, the following paper first effectively discovers a new vulnerability. Second, it proposes a detection approach based on a static analysis (defined with the help of Datalog). MadMax analyses the entirety of smart contracts in the Ethereum blockchain at the time of this writing in just 10 hours and flags vulnerabilities in contracts that hold billions of dollars. The analysis MadMax proposes is fairly precise: manual inspection of a sample of flagged contracts shows that 81% of the sampled warnings do indeed lead to vulnerabilities.

The impact of this work is long-ranging and has some implications for the blockchain industry as a whole. Specifically, the metering approach that is based on gas measurements is a highly imperfect design. Fundamentally, assigning fixed weights to individual instructions is bound to create a mismatch with the specifics of individual hardware architectures.

However, given that blockchain is experiencing rapid adoption, the focus on metering and out-of-gas attacks of the following paper is well-warranted and more research is needed in this space to both propose new ways to do metering and to fix existing attacks.

Benjamin Livshits is Chief Scientist of Brave Software and an associate professor at Imperial College London, U.K.