The Quantstamp Protocol

Leonardo Passos, Sr. Researcher

(leo@quantstamp.com)

Quantstamp™

“A survey of attacks of Ethereum smart contracts”

Atezi & et al., 2017

Level	Cause of vulnerability
Solidity	Call to the unknown Gasless send Exception disorders Type casts Reentrancy Keeping secrets
EVM	Immutable bugs Ether lost in transfer Stack size limit
Blockchain	Unpredictable state Generating randomness Time constraints

DealBook / Business & Policy

A Hacking of More Than $50 Million Dashes Hopes in the World of Virtual Currency

By NATHANIEL POPPER JUNE 17, 2016

A hacker on Friday siphoned more than $50 million of digital money away from an experimental virtual currency project that had been billed as the most successful crowdfunding venture ever — taking with him not just a third of the venture's money but also the hopes and dreams of thousands of participants who wanted to prove the safety and security of digital currency.

The attack most likely puts an end to the project, known as the Decentralized Autonomous Organization, which had raised $160 million in the form of Ether, an alternative to the digital currency Bitcoin. While the computer scientists involved in the project are aiming to tweak the code that underpins Ether in a way that will recover the money, the theft is

RELATED COVERAGE

A Venture Fund With Plenty of Virtual Capital, but No Capitalist MAY 21, 2016

Paper Points Up Flaws in Venture Fund Based on Virtual Money MAY 27, 2016

ETHEREUM, TECHNOLOGY

BatchOverflow Exploit Creates Trillions of Ethereum Tokens, Major Exchanges Halt ERC20 Deposits

Sam Town | April 25, 2018 | 3 min read | 3139 Views

Smart Contracts need to be secured!

However, in practice...

(1) Manual auditing is laborious

Token sale contracts ~ days
Protocol ~ weeks

(2) Manual auditing is generally centralized

(3) Results are not generally verified by others

Not surprisingly, bugs continue to occur :(

Can we make things better?

Quantstamp: The protocol for securing smart contracts

Quantstamp is the first smart contract security-auditing protocol. We are extending Ethereum with technology that ensures the security of smart contracts. Our team is made of up of software testing experts who collectively have over 500 Google Scholar citations.

Founders

Richard Ma, Cornell ECE
Algorithmic Portfolio Manager

Steven Stewart, MCS, BA
PhD, U. Waterloo
Software verification, Database implementation

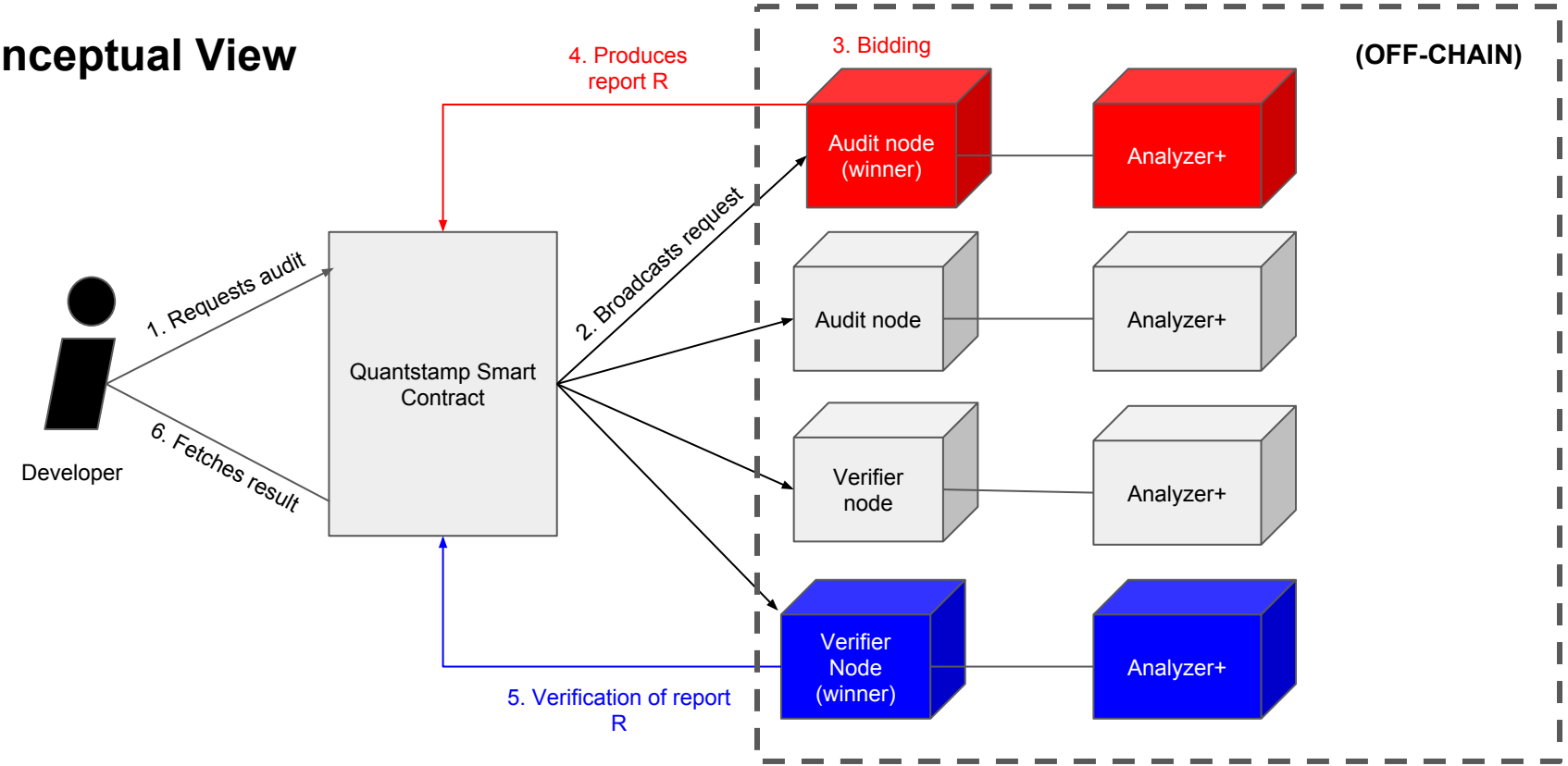
Founding Team Members

Dr. Vajih Montaghami, PhD
Formal methods

Ed Zulkoski, B.S.
PhD-candidate, U. Waterloo
SAT/SMT solvers

Leonardo Passos, PhD
Compilers and Programming Languages

Conceptual View



Beta-Net (Coming Soon)

1. A permissioned set of audit nodes (whitelisted)
2. Assumption: whitelisted audit nodes are trustworthy; audits are correct
3. Audit nodes run two third-party analyzers
4. Web interface to shield users from directly interfacing with the smart contract
5. Release: August, 2018

Challenges ahead...

How to efficiently verify audits?

Verifiers could check traces and challenge them

Requires all nodes to perform the exact same steps

Issue: no room for randomness; no parallelism

Full Automation is Hard!

To rule out false positives, human intervention is required

Can we provide a method to reward a decentralized set of auditors?

Possible solution: decentralized bug bounties

Decentralized Bug Bounties

Some properties, however, there are challenges in reinforcing them

(1) Witness Integrity

(2) Consensus of error

(3) Fair exchange problem

(4) Community reputation robustness

Summary

1. Smart contract security critical for widespread adoption
2. Manual audits are expensive and laborious
3. Automated audits raise the bar for contract security
4. Decentralized bug bounties complement automated audits
5. Blockchain still in its infancy, more research needed