IMT SCHOOL FOR ADVANCED STUDIES LUCCA

Research Project Fellowship Grant
Financed with the funds of the project PAI 2018 *VeriOSS: a security-by-smart contract verification framework for Open Source Software* - P0137

VERIOSS SMART CONTRACT DEVELOPMENT - Activity report

FRANCESCO MUCCI
francesco.mucci@imtlucca.it, francesco.mucci@stud.unifi.it

Research Area: *CSSE - Computer Science and Systems Engineering*
Research Unit: *SysMA - System Modeling and Analysis*
Research Fields: *Smart contract, blockchain, ethereum*
Scientific Coordinator: Dr. *Gabriele Costa*

november 15, 2019 - may 14, 2020
may 06, 2020

INDICE

1 INTRODUCTION

1.1 VERIOSS PROJECT

The *VeriOSS* project fits into the context of bug bounty programs, which are used by large companies, *Bounty Issuer (BI)*, to encourage penetration testers, *Bounty Hunter (BH)*, to search for software bugs: BI offers a reward that, usually, depends on the typology and criticality of the bug; when a BH founds an eligible bug, the bug is disclosed against the payment of a reward. Nowadays there are web portals (e.g. hackerone.com) that allow companies to publicly make their own bounty offers; however, these bug bounty programs have some issues that reduce their effectiveness: for example, the lack of reliability of the BI, which often can refuses to pay the reward, force many BHs to seek alternative markets to sell the identified vulnerabilities.

The purpose of the VeriOSS project is the creation of a platform that allows to publish a bug bounty request and that assists the BI and BH during negotiation, verification and sale of the bug. To ensure the interaction between two parties which do not trust each other, the VeriOSS platform will use Blockchain technology for the publication of the bounty and for the management of the payment, while it will use proof of knowledge protocols to guide the automatic verification of bugs.

1.2 RESEARCH PROJECT'S GOAL

The goal of this research work was to create a prototype of the bounty-claim protocol for the VeriOSS blockchain-based platform. VeriOSS will be a bug bounty platform: it will allow a BI to issue a bounty (in the form of a `Bounty` smart contract) for certain types of bugs. The bounty-claim protocol will allow the claim of the bounty, guiding the bug-disclosure and the payment of the BH. Basically, the protocol will implement a remote reverse debugging process: the bug execution trace will be progressively disclosed by the BH, starting from the last program state of the trace, against a sequence of partial rewards. To make up for the lack of a trusted third party, each transaction will be carried out through the use of a separate `ChallengeReward` smart contract.

1.3 RESEARCH ACTIVITY PLANNING

On November 15, 2019, during a meeting with Dr. Gabriele Costa, scientific coordinator of the project, and Dr. Letterio Galletta, the following intermediate goals were identified:

1. outline the architecture of the bounty-claim protocol for the end of February;
2. have a working version of the prototype for the end of April;
3. design a second version of the prototype for the end of May.

2 REPORT OF THE WORK DONE

2.1 FIRST QUARTER: NOVEMBER 15 2019 - FEBRUARY 14 2020

2.1.1 *Study and acquisition of skills*

The first quarter was devoted entirely to the study and acquisition of the skills needed to design and develop the prototype of the bounty-claim protocol.

Following a preliminary study of the article *"VeriOSS: using the Blockchain to Foster Bounty Programs"* [1], it was established the need to acquire skills related to:

- design and development of Solidity smart contracts and their deployment on the Ethereum network;
- use of angr, a tool for automatic verification of programs based on symbolic execution.

Fundamental concepts and technologies have been studied in depth, such as:

- knowledge related to Blockchain technologies (Ethereum, Solidity, Dapp); in particular, the following have been studied and attended:
  - *"Mastering Ethereum"* by Wood and Antonopoulos [2];
  - open courses *"Blockchain Basics"* [3], *"Smart Contracts"* [4] and *"Decentralized Applications (Dapps)"* [5] taught by Prof. Bina Ramamurthy as part of the *"Blockchain Specialization"* offered by the University at Buffalo on Coursera;
- basic principles of symbolic execution with angr;
  in particular, the following have been studied and attended:
  - *"Introduction"*, *"Core Concepts"* and *"Examples"* sections present in the official angr documentation [6];
  - lessons on the analysis of programs [7] and on angr [8] held by Prof. Giovanni Vigna as part of the course *"CS279 Advanced Topics in Security"* offered by the University of California in Santa Barbara.

In addition to the traditional study activity, with the goal of acquiring more basic knowledge, the following were attended:

- seminar*"Blockchain, smart contract e cryptovalute per le imprese"*, held by Dr. Giovanni Brancalion Spadon on 26 November 2019, at the "Polo delle Scienze Sociali" of the University of Florence (`https://datascience.unifi.it/index.php/events/2019-11-26/`);
- two meetings of the *"IMT Reading group on Blockchain and Criptocurrencies"* coordinato del Prof. Nicola Dimitri, in particular:
  - the eighth meeting, on 18 December 2019, in which Dr. Galletta presented chapters 6, 7 and 8 of the *"Mastering Ethreum"* textbook [2];
  - the ninth meeting, on January 14, 2002, in which Dr. Galletta continued the presentation of the text *"Mastering Ethereum"* [2], addressing chapters 9, 10, 11 and 12.

2.2 SECOND QUARTER: FEBRUARY 15 2020 - MAY 14 2020

The second quarter was devoted to the design and development of the prototype of the bounty-claim protocol.

2.2.1 *Challenge-response loop v.2*

During the first weeks the article *"VeriOSS: using the Blockchain to Foster Bounty Programs"* [1] was studied in depth and a possible architecture for the prototype was envisioned. On March 9, 2020 a remote meeting with Dr. Costa and Dr. Galletta, to clarify some doubts about the architecture, was held and it was decided to go to move off-chain part of the data and the computations previously entrusted to the ChallengeReward contract (the smart contract responsible for the partial rewarding process). To do this, it was decided to rely on distributed data storage and an oracle service.

So, first of all, the documentation of the following services was studied:

- InterPlanetary File System, a decentralized data storage [9] [10];
- Provable, an oracle service [11].

Consequently, an updated version of the challenge-response interaction (the heart of the bounty-claim protocol) was defined and a document that described it in detail was written; this document was delivered on April 7, 2020. In the meantime, a first draft of the `ChallengeReward` contract has been defined.

2.2.2 *Challenge-reward smart contract*

On April 9, 2020, a further remote meeting with Dr. Costa and Dr. Galletta took place to discuss the next steps of the work. We decided to focus primarily on the completion and testing of the `ChallengeReward` smart contract; furthermore, the possibility of modelling the smart contract as a state machine in order to be able to perform a formal verification procedure using the VeriSolid framework emerged during the meeting. Therefore, the various articles relating to this framework were studied ([12] [13] [14] [15]) and an attempt to define the state machine corresponding to the contract was made; however, various issues have been encountered: only the third version of the framework (released in February 2020) supports Solidity v0.5. and that version doesn't seem to work properly (consistent problems have been identified with the code generation from the state machine and with the verification of the properties defined in CTL). We haven't yet had the opportunity to discuss these problems and, at the moment, the possibility of using this framework has been shelved. Despite this, the study done was not in vain: VeriSolid forces the use of some design patterns aimed at preventing known vulnerabilities; the use of some of these patterns will be maintained in the `ChallengeReward` smart contract code.

Currently, several alternative versions of the `ChallengeReward` contract were developed [1] and they are going to be unit tested; once the testing is complete, they will be checked with a real example.

2.3 NEXT STEPS

We are behind the original planning, however, the work started will be continued and concluded as part of my master's thesis. Therefore, the next steps could be:

- find a way to perform a formal verification of the `ChallengeReward` contracts (trying again to use the VeriSolid framework);
- evaluate and compare the costs (in terms of gas points and Provable fees) of different versions of the `ChallengeReward` contract;
- integrate the `Bounty` smart contract (defines the bug's bounty) into the project;
- design and develop Python codes for the challenge generation and for the debug-info verification.

1 `https://github.com/FrancescoMucci/VeriOSS-project`

BIBLIOGRAFIA

[1] Gabriele Costa, Letterio Galletta, "VeriOSS: using the Blockchain to Foster Bounty Programs", *Proceedings of AMC SAC Conference (position paper)*, 2020 (Cited on pages 3 and 4.)

[2] Andreas M. Antonopoulos, Gavin Wood, "Mastering Ethereum: Building Smart Contracts and DApps", *O'Reilly Media, Inc.*, `https://github.com/ethereumbook/ethereumbook`, 2018 (Cited on pages 3 and 4.)

[3] Bina Ramamurthy, "Blockchain Basics" course, *Blockchain Specialization offered by University at Buffalo on Coursera*, `https://www.coursera.org/learn/blockchain-basics` (Cited on page 3.)

[4] Bina Ramamurthy, "Smart Contracts" course, *Blockchain Specialization offered by University at Buffalo on Coursera*, `https://www.coursera.org/learn/smarter-contracts` (Cited on page 3.)

[5] Bina Ramamurthy, "Decentralized Applications (Dapps)" course, *Blockchain Specialization offered by University at Buffalo on Coursera*, `https://www.coursera.org/learn/decentralized-apps-on-blockchain` (Cited on page 3.)

[6] "angr Documentation", `https://docs.angr.io/` (Cited on page 3.)

[7] Giovanni Vigna, "Program Analysis" lecture, *class "CS279 Advanced Topics in Security" at the University of California in Santa Barbara*, fall 2017, `https://www.youtube.com/watch?v=v6O5ItbzoGI` (Cited on page 3.)

[8] Giovanni Vigna, "angr Tutorial" lecture, *class "CS279 Advanced Topics in Security" at the University of California in Santa Barbara*, fall 2017, `https://www.youtube.com/watch?v=XgHZ6QnZkgc` (Cited on page 3.)

[9] "IPFS Documentation beta", accessed during March 2020, `https://docs-beta.ipfs.io/` (Cited on page 4.)

[10] "IPFS Primer", accessed during March 2020, `https://dweb-primer.ipfs.io/` (Cited on page 4.)

[11] "Provable Documentation", accessed during March 2020, `https://docs.provable.xyz/` (Cited on page 4.)

[12] Anastasia Mavridou, Aron Laszka, "Designing Secure Ethereum Smart Contracts: A Finite State Machine Based Approach", *Proceedings of the 22nd*

International Conference on Financial Cryptography and Data Security, 2018 (Cited on page 5.)

[13] Anastasia Mavridou, Aron Laszka, "Tool Demonstration: FSolidM for Designing Secure Ethereum Smart Contracts", *Proceedings of the 7th International Conference on Principles of Security and Trust*, 2018 (Cited on page 5.)

[14] Anastasia Mavridou, Aron Laszka, Emmanouela Stachtiari, Abhishek Dubey, "VeriSolid: Correct-by-Design Smart Contracts for Ethereum", *Proceedings of the 23rd International Conference on Financial Cryptography and Data Security*, 2019 (Cited on page 5.)

[15] Keerthi Nelaturu, Anastasia Mavridou, Andreas Veneris, Aron Laszka, "Verified Development and Deployment of Multiple Interacting Smart Contracts with VeriSolid", *Proceedings of the 2nd IEEE International Conference on Blockchain and Cryptocurrency*, 2020 (Cited on page 5.)