KING'S College LONDON

Smart Contracts Security

-Report-

Group 46

Method II – narrated presentation

Contributors:

Diana-Sandra Popa (K19014896)

Sebastian-Mihai Mesca (K1921743)

Zihao You (K1923149)

Iris-Catalina Simionas (K19009091)

Andrei-Bogdan Balcau (K19009092)

Livia-Oana Neagu (K19027251)

KING'S College LONDON

Documentary about the work presented

The cryptocurrency resources are limited. Therefore, internally, the “cryptocurrency” is just a large memory that can be sliced in very small pieces, with the end purpose of someday filling all the gaps and with the impossibility of further creation of ‘coins’. From each address (e.g. wallet) involved, there are some ‘pointers’ to the memory cells which contain stocks of cryptocurrency. A “transaction” is just a swap of pointers. A smart contract must execute this job, and, therefore, the security of this piece of software is a big concern.

From our analysis we found out that the main problems with the smart contract are in general related to the programming language, but also to the blockchain’s properties used. The language chosen in Ethereum is Solidity, which is Turing Complete. A Turing Complete language has more features available for the programmer, but it is prone to more attacks. On the other hand, a language like Script used in Bitcoin uses a simple stack and does not allow iterations, which means creating usable code is harder for both the developer and the attacker.

The blockchain technology comes with an additional security layer over the smart contracts, but this is also prone to more different and specific attacks. The blockchain increases the security of the smart contracts by using a consensus between all the nodes (PoW, PoS, SPoS). As Proof-of-Work (PoW) produces a very high carbon footprint, companies have decided to start implementing the Proof-of-Stake (PoS) consensus. This mechanism allows the validation of transactions to be done by staking a part of the cryptocurrency owned. This incentivises nodes to be fair when mining.

A peer-to-peer network using PoS is exposed to Sybil Attacks and Nothing at Stake Attacks. The consensus plays an important role in mitigating those two attacks. A Sybil Attack is a type of attack in which a node in the network impersonates multiple identities at the same time. This risk is mitigated by making it impossible for a user to join the network of validators if they don’t stake at least as much as the minimum staking value in that network. Another problem would be the Nothing at Stake Attack. This happens when node tries to validate a fraudulent transaction. The PoS consensus in designed in such way to reward user for being honest and punish the fraudulent nodes by locking their stake.

The security lifecycle of smart contracts is exactly like in software engineering. We have security design, security implementation, testing before deployment, plus monitoring and analysis.

The first step is focused on how to design the smart contracts to avoid security issues, by following some methodologies and principles such as ‘Prepare for failure’ – the contract code must respond gracefully to bugs (e.g. in the eventuality of an DOS attack, pause the contract to avoid further financial losses). The actual security implementation comes right after the design step - developers of the contracts make use of several security libraries (e.g. OpenZeppelin, SafeMath) to reach the endpoint of having a secure product.

KING'S College LONDON

Reflection over the group work experience

Teamwork has been very productive, both during meetings and in our research schedule. We've had weekly meetings, sometimes even 2 meetings per week on Microsoft Teams. We've also created a Whatsapp group chat mostly used for choosing an appropriate time and date for each meeting, for urgent matters and for gathering research materials. When it came to research, we've talked in every meeting about what each one of use has studied for the theme we've chosen (Smart Contracts Security). We discussed different opinions and ideas every individual had for the project, usually deciding the next steps we had and splitting the tasks. In the beginning, we've decided on what should be researched by every member in order to make sure everybody is up to date with the proposed themes of the coursework and eventually created an online poll (pollev.com) with pros and cons of every theme and answers from every member in order to discuss them in the upcoming meeting. After thorough talks, to make sure we were making the right decision, we've chosen Smart Contracts Security.

Creating a Whatsapp group chat was very effective especially because we gathered lots of useful research material and kept everything in a single place. Whenever one of the team members found something useful for the project, they shared their resource there and everyone had access to it.

We have thought of way to implement our presentation more like a virtual conference, where we play the role of specialists providing a brief description of smart contracts for the audience.

The times and dates of the meetings have been flexible. All the members had to be on board with the schedule of the meetings and in case something came up, we would move the meeting either at a later or sooner time. This "strategy" has turned out to be effective because it allowed us to host every meeting with every member of the team present. Even though everything was done online, none of the members have been in very different time zones which was also helpful because it only had a small impact on our scheduling. When it came to splitting the two major tasks (the report and the presentation) we've decided to split the members in half: 3 of us worked on the presentation and 3 of us worked on the report and set a deadline for both tasks. We've talked and approved beforehand whenever something had to be adapted to make sure everybody is on board with what had to be changed. After the presentation was finished by the 3 members in charge of it, the other 3 started working on the report. We held a rehearsal meeting to make sure everything was adequate and made small changes to the design and wording wherever it was needed.

If we had been given more time to submit the work, the biggest improvement would have been delivering more interactive content for the audience, by an actual demonstration of different types of attacks in an Ethereum Testnet.

KING'S College LONDON

References

Antonopoulos, A., 2017. *Mastering Bitcoin.* 2nd ed. s.l.:O'Reilly.

Antonopoulos, A. & Wood, G., 2018. *Mastering Ethereum.* 1st ed. s.l.:O'Rielly.

Asem, G., 2020. *Smart Contracts Security,* s.l.: s.n.

Elrond, 2019. *Elrond Technical Whitepaper,* Cluj, Romania: Elrond.

Huang, Y. et al., 2019. *Smart Contracts Security: A Software Lifecycle Perspective,* s.l.: s.n.

Nakamoto, S., 2008. *Bitcoin: A Peer-to-Peer Electronic Cash System.* s.l.:s.n.