An evaluation of the reentrancy vulnerability on GoQuorum-based smart contracts

1. Introduction

- In Ethereum, **smart contracts** were integrated
  - **Smart contracts** can thus be used to automate behaviour, both **productive** and **exploitative**
- **GoQuorum** is based on Ethereum, with a focus on **security** and **privacy**
- Motivation
  - little research into GoQuorum-based smart contract vulnerabilities so far
  - reentrancy is well-known but missing depth
  - complete evaluation of both vulnerability and countermeasures




2. Research Question

How is the **reentrancy** vulnerability exploited in **GoQuorum**-based **smart contracts**, and **what** can be done to **prevent or mitigate** the associated risks?

References

Image references introduction, left to right.
[1] The Noun Project, [Online]. Available at: https : / /thenounproject . com / term / smart - contract / 1688703/
(Accessed on 06/29/2021).
[2] Vecteezy, [Online]. Available at: https://www.vecteezy.com / vector - art / 548141 - cryptocurrency - and -blockchain-icons-or-logo
(Accessed on 06/29/2021).
[3] Ethereum Foundation, [Online]. Available at: https ://ethereum.org/en/assets/
(Accessed on 06/29/2021).
[4]ConsenSys, [Online]. Available at: https : / / cdn .consensys.net/uploads/QuorumAvatarBlue02.png
(Accessed on 06/29/2021).

3. Methodology

1. Explore literature
2. Ethereum vs GoQuorum
3. Collect vulnerabilities
   a. Implement vulnerability
   b. Implement countermeasure
   c. Analyse side-effects
4. Collect findings

4. Findings

- Reentrancy
  - Reentrancy exploits external contract calls combined with incorrect contract state updates to extract ether from the victim.
  - Both public and private contracts may be vulnerable
- Attack features
  - Contract's control flow on a state variable
  - Gas limit
- Countermeasures per category

Table 1: Categorized prevention and mitigation techniques. *Each presented technique is categorized into addressing one of three vulnerability aspects: attack features, function access, and vulnerability awareness. Moreover, M marks a mitigation technique and P a prevention technique.*

Countermeasure	feature	access	awarenes
Correct state variable update	P		
Gas limit	M		
Mutex/Guard		P	
Analysis tools			M
Naming convention			M
(Enhanced) permissioning		M	
Private contract		M	

5. Discussion

- Testing done on small network
- Only one vulnerability evaluated

6. Future work

- Reentrancy in big network.
- Countermeasure effectiveness testing
- More known Ethereum vulnerabilities should be evaluated in a GoQuorum context.
- Studies into novel GoQuorum-specific vulnerabilities.
- Other Ethereum soft-fork should get a similar treatment

7. Conclusion

- Reentrancy countermeasures
  - Checks-effects-interaction pattern
  - Combination of other techniques, especially those of the awareness category
- First scientific paper presenting methods to deploy and interact with GoQuorum network.

by Sara Op den Orth
S.M.OpdenOrth@student.tudelft.nl - 02-07-2021
Responsible professor: Kaitai Liang Supervisor: Huanhuan Chen (PhD)