CONFIDENTIAL

Digital Security

БЕЗОПАСНОСТЬ КАК ИСКУССТВО

EXPERT REPORT

on the results of the security assessment
PoA.network cross-chain bridge

Project timeframe:
11.04.2018-19.04.2018
Project manager:
Pertsev A.O.
Head of the Audit Department:
Cherbov G.S.

1. INTRODUCTION 3

1.1. General provisions 3
1.2. Generally accepted abbreviations 3
1.3. Abstract 3
1.4. Scope of work 4

2. SECURITY ASSESSMENT PRINCIPLES 5

2.1. IS threats 5
2.2. Attacker model 5
2.2.1. Used attacker models 6

3. SMART-CONTRACT CODE AUDIT 7

3.1. List of detected vulnerabilities of the System 7
3.1.1. The result of the onTokenTransfer call is not processed (ERC677) 7
3.2. List of detected security weaknesses of the System 8
3.2.1. Unused functionality 8
3.2.2. Redundant code 8
3.2.3. No address check 9
3.2.4. Inadequate checking of received values 9
3.2.5. Contract initialization method is not protected from an attacker calling it 10

4. OTHER RECOMMENDATIONS 11

4.1. Commissions 11
4.2. Selecting Validators for transfer confirmation 11
4.3. Bridge-UI security improvement 11

5. CONCLUSION 12

APPENDIX 1. SECURITY ASSESSMENT. BACKGROUND INFORMATION 13

Security level analysis 13
Vulnerability severity 13
Ease of vulnerability exploitation 13
Vulnerability accessibility 15
Likelihood of exploitation 15
Vulnerability impact 16

1. Introduction

1.1. General provisions

This expert report contains the results of security assessment of the cross-chain bridge (hereinafter referred to as “the System”) owned by PoA.network (hereinafter referred to as “the Company”) with detailed recommendations on improving its current security level.

1.2. Generally accepted abbreviations

Table 1.2–1. Generally accepted abbreviations

Abbreviation	Meaning
EVM	Ethereum virtual machine
Gaz	Smart contract account unit
Home	“PoA.network” network
Foreign	Ethereum foundation network (or any compatible network)
POA20	Token, issued on the Foreign side
POA	Native token of the Home network
Validator	Miner in the PoA.network network. Responsible person

1.3. Abstract

“Digital Security” experts conducted a security assessment of the System in the period of 11.04. 2018 – 19.04.2018.

In the scope of performed works, an external attacker model was used.

As a result of performed works, the experts found duplicate code, configuration weaknesses of the System, and provided the list of measure to decrease the rate.

The overall security level of the System was rated as “high.”

Below, you can find the detailed description of the found security imperfections and related security risks. The recommendation on fixing the vulnerabilities are provided as well.

1.4. Scope of work

The scope of security assessment included the Company's external resources listed in Table 1.4. and 1.5.:

Table 1.4. Analyzed github projects of the Company

№	Project	Link to a used commit
1	bridge-ui	#df5da44acc544cd93701c8214ba2ecd077b5a341
2	parity-bridge	#da91da9db70ac7c4a971eb86eab8e7adf6ce4884
3	poa-parity-bridge-contracts	#cf7f7d4f15bf51c24c93cb9dbde4d2b4560d2ddb

Table 1.5. Analyzed hosts of the Company

№	Hostname/IP address/Resource
1	https://poanetwork.github.io/bridge-ui/#/

2. Security assessment principles

2.1. IS threats

There are 3 types of IS threats that can affect the Company's information resources: violations of confidentiality, integrity or availability of information.

Confidentiality violation is usually aimed at information disclosure. If a violation is successful, the information becomes known to people, who should not have access to it: unauthorized personnel of the Company, clients, partners, competitors, and third parties.

Integrity violation is aimed at modification or corruption of the information, which can lead to modification of its structure or content, and to complete or partial destruction of the data.

Availability violation (denial-of-service threat) involves data system users' inability to access the information.

The core principle of this IS audit is an estimation of exploitation likelihood of the aforementioned threats affecting the Company's information resources, within the framework of a pre-defined attacker model.

2.2. Attacker model

A potential attacker is an individual or a group of individuals, acting either in a collision or independently, whose intended or unintended actions can carry the aforementioned threats to the IS, infringe information resources of the System or negatively affect the Company's interests.

IS threats are basic threats to confidentiality and integrity of information and the threat of the System denial-of-service.

Attackers can pursue the following goals (and their possible combinations):

- cause Denial of Service;
- escalate their privileges in the System;
- get unauthorized access to business-critical data.

In the scope of work, "Digital Security" experts used an external attacker model.

2.2.1. Used attacker models

The following submodels were used to conduct the security assessment:

- external attacker from the Ethereum network;
- external Internet attacker.

A potential attacker is deemed to have knowledge of the System since the platform code is open.

3. Smart-contract code audit

3.1. List of detected vulnerabilities of the System

3.1.1. The result of the onTokenTransfer call is not processed (ERC677)

Severity: Low
Likelihood of exploitation: Low
Overall impact level: Low

Description:

According to the ERC677 standard, if a smart contract is a recipient of a token, it must have the onTokenTransfer callback function. However, the POA20 contract does not process the result of the result of the onTokenTransfer call.

Impact:

In case of a failure, a recipient contract will return “false” and consider it does not receive any tokens, while POA20 will change a recipient’s balance.

Vulnerable project:

- poa-parity-bridge-contracts. POA20.sol#L43

Technical details:

To transfer tokens, the UI utilizes the transferAndCall functionality that does not process callback functions of a recipient contract.

Recommendations:

- Process responses of onTokenTransfer.

Status:

- Fixed #d4e9d831dc7212fd97730c8403540fb2db447388

3.2. List of detected security weaknesses of the System

3.2.1. Unused functionality

Description:

A smart contract employs some functionality that is not used during its operation.

Impact:

Additional functionality expands attack surface.

Vulnerable project:

- poa-parity-bridge-contracts. POA20.sol#L3
- poa-parity-bridge-contracts. POA20_flat.sol#L364

Technical details:

The POA20 token inherits the Pausable contract but does not use it. In addition, POA20 also inherits the Mintable contract that implements the finishMinting() function. According to requirements, the stoppage of token issuing is not necessary.

Recommendation:

- Remove unused code.

Status:

- Fixed #3a4c663d8446b1c40f8f6962086a4fda5cf8670b
- Fixed #d4e9d831dc7212fd97730c8403540fb2db447388

3.2.2. Redundant code

Description:

A smart contract contains redundant code.

Impact:

Redundant code results in excessive spending of Gaz.

Vulnerable project:

- poa-parity-bridge-contracts. U_BridgeValidators.sol#L20
- poa-parity-bridge-contracts. U_ForeignBridge.sol#L36

Technical details:

- U_BridgeValidators.sol#L20. Both checks (for copies and zero addresses) have been already implemented in the addValidatore function.
- U_ForeignBridge.sol#L36. There is a similar check in the setForeignDailyLimit function.

Recommendations:

- Remove unused code.

Status:

- Fixed #bcd608479f8f1678cbbfe9ca1b0cae69a9d63fed
- Fixed #abccf2b63ae92dbe0c3b9f61a59ee64bf13f5d59

3.2.3. No address check

Description:

A smart contract uses an internal method of identifying an owner. This method has checks of a passed address (unlike the public method implemented for the very same purpose).

Impact:

A contract does not notify about errors in an address.

Vulnerable project:

- poa-parity-bridge-contracts. U_BridgeValidators.sol#L16

Technical details:

Inside the initialize method, there is the setOwner method used to identify an owner. This method does not perform any address check (unlike the Ownable.transferOwnership method).

Note: the transferOwnership method cannot be used since it is marked with the onlyOwner modifier, while an owner is not identified at the moment of initialization (a consequence of updateable contracts being used).

Recommendations:

- Generally, if the public method cannot be used then it worth performing mandatory implemented checks before calling internal.

Status:

- Fixed #bcd608479f8f1678cbbfe9ca1b0cae69a9d63fed, and #a22b8713a126ccb85f9b7fec9a5c14fc81177466

3.2.4. Inadequate checking of received values

Description:

During its initialization, a smart contract receives the values of maximum and minimum transfer amount for a transaction. However, the performed check is inadequate.

Impact:

If the minimum value is bigger than the maximum one, a contract will not notify about the error.

Vulnerable project:

- poa-parity-bridge-contracts. U_ForeignBridge.sol#L37
- poa-parity-bridge-contracts. U_HomeBridge.sol#L25

Technical details:

There is the following check in the initialize method:

```
require(_maxPerTx > 0 && _minPerTx > 0);
```

A more efficient option is as follows:

```
require(_minPerTx > 0 && _maxPerTx >= _minPerTx);
```

Recommendations:

- See **Technical details**.

Status:

- Fixed #abccf2b63ae92dbe0c3b9f61a59ee64bf13f5d59

3.2.5. Contract initialization method is not protected from an attacker calling it

Description:

Because there is a special contract-updating mechanism used, contracts cannot employ the constructor. In conjunction with that, a separate method for Storage initialization is used. It is not possible to protect this method from calling from a random address.

Impact:

An attacker can track the updating process and perform a front-running attack to initialize a contract before its owner does.

Vulnerable project:

- poa-parity-bridge-contracts. U_HomeBridge.sol#L16
- poa-parity-bridge-contracts. U_ForeignBridge.sol#L27
- poa-parity-bridge-contracts. U_BridgeValidators.sol#L14

Technical details:

The initialize method that is responsible for the identification of an owner has no access modifiers. An attacker can track all contract update and use a front-running attack to become an owner of a new contract. After that, an attacker can sabotage the System's operation.

Recommendations:

- During the updating process and contract deployment, the upgradeToAndCall function should be used to execute updating during a single transaction. However, for this purpose, it necessary to remove "require" when a contract is called, because "initialize" returns nothing.

Status:

- Fixed #1f1281ed8c04b029a30bd830928762d0e08c5d2b

4. Other recommendations

4.1. Commissions

Parity-bridge does not charge transfer commissions. Meanwhile, the Validators send their signatures by calling ForeignBridge.submitSignature() during the POA20 to POA transfer, and ForeignBridge.deposit() during the POA to POA20 transfer, and are charged with a transfer commission.

Given this, we recommend:

- Turn over the bridge, i.e., transfer the submitSignature and deposit functions to the Home side (it will become free for Validators to call the functions).
- Set commission on POA20 issuing. For example, a user is accredited with N-x POA20 tokens, where:
  - N – the number of tokens sent to HomeBridge;
  - x – commission that should be sent as tokens to a special address. From this address, Validators will send those commission tokens to the exchange and sell them to new users.

4.2. Selecting Validators for transfer confirmation

At the time of the security assessment, Validators had no policy for selecting a person responsible for a particular transaction. Thus, Validators that are online would react on token transfers, although to confirm the transfer only a certain number of them that is less than the number of Validators would suffice.

Therefore, it is recommended to implement a mechanism that will provide an optimal set of Validators responsible for a particular transfer in the Home network; and a user managing signature sending to the Foreign side.

4.3. Bridge-UI security improvement

To improve the security of the UI, it is recommended to implement the following mechanisms (headers):

- X-Frame-Options: DENY
- Strict-Transport-Security: max-age=31536000; includeSubDomains;

5. Conclusion

During the security assessment, the experts found no critical vulnerabilities that may enable an attacker to obtain Validators' or users' funds.

Nonetheless, they managed to find security weaknesses and flows and provided the list of designated recommendations. The overall security level of the System was rated "High."

Appendix 1. Security assessment. Background information

Security level analysis

To analyze the security level of the System, it is necessary to measure severity and likelihood of exploitation of the detected vulnerabilities. The likelihood of exploitation is measured, according to the ease of vulnerability exploitation and the accessibility of a vulnerability.

Vulnerability severity

The "Severity" property of a vulnerability describes possible results of this vulnerability exploitation, regarding confidentiality, integrity, and availability of information processed on a vulnerable resource. Severity levels are described in the Table A-1.

Table A–1. Vulnerability severity levels

Security level	Confidentiality violation	Integrity violation	Availability violation
None	Does not happen.	Does not happen.	Does not happen.
Low	Obtaining access to noncritical information by an attacker through privilege escalation	Integrity violation of noncritical information by an attacker with basic user rights in the System	Short-time denial-of-service of a mission-critical application
Average	Confidentiality violation of sensitive data by an attacker with basic user rights in the System	Integrity violation of sensitive data by an attacker with basic user rights in the System	Denial of service of a mission-critical application or a short-time denial of service of the System
High	Confidentiality violation of critical information by an attacker with administrator rights in the System	Integrity violation of critical information by an attacker with administrator rights in the System	Denial of Service of the System

Ease of vulnerability exploitation

The "Ease of exploitation" property of a vulnerability defines what hardware and software, time and computing resources, and professional skills are required to exploit a vulnerability (Table A-2).

Table A–2. Ease of vulnerability exploitation levels

Level	Description
Low	Vulnerability exploitation requires high computing powers, significant time resources, developing new software, configuration analysis of the System, determination and testing possible ways and conditions of successful exploitation of this vulnerability.
Average	Vulnerability exploitation requires high-performance computing, extensive time resources, special hardware and software, and analysis of a violated system configuration. An attacker does not have to have deep knowledge of the system or professional skills to perform an attack.
High	Vulnerability exploitation does not require the use of any special hardware or software, high-performance computing, time resources or any professional skills to perform an attack.

Vulnerability accessibility

The “Accessibility” property of a vulnerability defines what user classes have access to a vulnerable resource (Table A-3).

Table A–3. Accessibility levels

Level	Description
Low	Privileged users
Average	Registered users
High	All users

Likelihood of exploitation

The likelihood of exploitation is calculated according to “ease of exploitation” and “accessibility” levels (Table A–4).

Table A–4. Likelihood of exploitation levels

Likelihood of exploitation		Ease of exploitation		
		Low	Average	High
Accessibility	Low	Low	Low	Average
	Average	Low	Average	High
	High	Average	High	High

Vulnerability impact

Vulnerability impact (for one of the existing threats) is measured, according to vulnerability severity (for one of the existing threats) and the likelihood of exploitation of a vulnerability (Table A-5).

Table A–5. Vulnerability impact levels

Vulnerability impact		Likelihood of exploitation		
		Low	Medium	High
Vulnerability severity	Low	Low	Low	Medium
	Medium	Low	Medium	High
	High	Medium	High	High