DeHacker

Code Security Assessment

BOUNDLESS NETWORK TOKEN

July 31th, 2024

Contents

Summary

DeHacker's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best prac tices. Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow/underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service/logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Issue Categories

Every issue in this report was assigned a severity level from the following:

Critical severity issues

A vulnerability that can disrupt the contract functioning in a number of scenarios or creates a risk that the contract may be broken.

Major severity issues

A vulnerability that affects the desired outcome when using a contract or provides the opportunity to use a contract in an unintended way.

Medium severity issues

A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.

Minor severity issues

A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.

Informational

A vulnerability that has informational character but is not affecting any of the code.

Overview

Project Summary

Project Name	BOUNDLESS NETWORK TOKEN
Platform	Ethereum
Website	burritowallet.com/
Type	DeFi
Language	Solidity
Codebase	https://github.com/rotonda1/bun-token/tree/bc129f42c0be8181a8d0ad76e3642744a79af629

Vulnerability Summary

Vulnerability Level	Total	Mitigated	Declined	Acknowledged	Partially Resolved	Resolved
Critical	0	0	0	0	0	0
Major	4	4	0	0	0	0
Medium	0	0	0	0	0	0
Minor	0	0	0	0	0	0
Informational	0	0	0	0	0	0
Discussion	0	0	0	0	0	0

Audit scope

ID	File	SHA256 Checksum
BTB	rotonda1/bun-token	d46cd5c47bc78f61d57aa30e2b54fc43020bd421d7df8400259c6b170d9d63ef
BTU	Rotonda1/bun-token	b8d0db5c27e15f87afb4a52d85b7c46e61b2e95736f1c9adbaa4091609127b93

Findings

ID	Issue	Severity	Status
BTB-02	Initial Token Distribution	Major	Mitigated
BTB-03	Centralization Risks	Major	Mitigated
BTB-04	Pausing Centralization Risks	Major	Mitigated
BTB-07	Withdrawal Centralization Risk	Major	Mitigated

MAJOR

BTB-02|Initial Token Distribution

Issue	Severity	Location	Status
Centralization	Major	contracts/BunToken.sol (bc129f42c0be8181a8d0ad76e3642744a79af629): 86	Mitigated

Description

All of the BUN tokens are sent to the contract deployer. This is a centralization risk because the deployer can distributetokens without obtaining the consensus of the community. Any compromise to these addresses may allow a hacker to stealand sell tokens on the market, resulting in severe damage to the project.

Recommendation

It is recommended that the team be transparent regarding the initial token distribution process. The token distribution planshould be published in a public location that the community can access. The team should make efforts to restrict access tothe private keys of the deployer account or EOAs. A multi-signature (⅔, ⅗) wallet can be used to prevent a single point offailure due to a private key compromise. Additionally, the team can lock up a portion of tokens, release them with a vesting schedule for long-term success, and deanonymize the project team with a third-party KYC provider to create greater accountability.

MAJOR

BTB-03 | Centralization Risks

Issue	Severity	Location	Status
Centralization	Major	contracts/BunToken.sol (bc129f42c0be8181a8d0ad76e3642744a79af629): 51, 89, 94, 98, 102, 148, 160, 166, 175, 179	Mitigated

Description

In the contract BunToken the role DEFAULT_ADMIN_ROLE has authority over the functions. Anycompromise to the DEFAULT_ADMIN_ROLE account may allow the hacker to take advantage of this authority to grant the DEFAULT_ADMIN_ROLE , PAUSER_ROLE , SYSTEM_ROLE role.

In the contract BunToken the role PAUSER_ROLE has authority over the functions. Any compromise to the PAUSER_ROLE account may allow the hacker to take advantage of this authority to pause the contract.

In the contract BunToken the role SYSTEM_ROLE has authority over the functions. Any compromise to the SYSTEM_ROLE account may allow the hacker to take advantage this authority lock tokens and revoke the lock.

In the contract BunToken the role _owner has authority over the functions. Any compromise tothe _owner account may allow the hacker to take advantage of this authority to transfer ownership.

In the contract LockedToken the role donor has authority over the functions. Any compromise to the donor account may allow the hacker to take advantage of this authority to unlock the tokens and withdraw tokens.

Description

In the contract LockedToken the role system has authority over the functions shown in the diagram below. Anycompromise to the system account may allow the hacker to take advantage of this authority to revoke the lock and transfertokens to the donor.

Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level ofdecentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefullymanage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommendcentralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accountswith enhanced security practices, e.g., multisignature wallets. Indicatively, here are some feasible suggestions that wouldalso mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign (⅔, ⅗) combination mitigate by delaying the sensitive operation and avoiding a single point of keymanagement failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Recommendation

Long Term:

Timelock and DAO, the combination, mitigate by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered fully resolved.

- Renounce the ownership and never claim back the privileged roles.
  OR
- Remove the risky functionality.

MAJOR

BTB-04|Pausing Centralization Risks

Issue	Severity	Location	Status
Centralization	Major	contracts/BunToken.sol (bc129f42c0be8181a8d0ad76e3642744a79af629): 98, 102	Mitigated

Description

In the contract BunToken , the PAUSER_ROLE has the authority to update the status of the _paused and further pause/resume the functionality of the token transfers.

```
98 function pause() public onlyRole(PAUSER_ROLE) {
99     _pause();
100 }
101
102 function unpause() public onlyRole(PAUSER_ROLE) {
103     _unpause();
104 }
```

Any compromise to the private key of the PAUSER_ROLE may allow hackers to take advantage of this authority and allow/prevent user access to token transfer functionalities

Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level ofdecentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefullymanage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommendcentralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accountswith enhanced security practices, e.g., multi-signature wallets.

Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign (⅔, ⅗) combination mitigate by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, mitigate by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement;
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Recommendation

Permanent:

Renouncing the ownership or removing the function can be considered fully resolved.

- Renounce the ownership and never claim back the privileged roles;
  OR
- Remove the risky functionality.

MAJOR

BTB-07|Withdrawal Centralization Risk

Issue	Severity	Location	Status
Logical Issue, Centralization	Major	contracts/BunToken.sol (bc129f42c0be8181a8d0ad76e3642744a79af629): 51, 160	Mitigated

Description

In the contract LockedToken , the donor or the system role has the authority to withdraw the beneficiary's locked tokensfrom the contract.

```
51 function revoke() public {
52       require(revocable, "L: not revocable");
53       require((msg.sender == donor) || (msg.sender == system),
"L: no permission");
54
55 uint256 amount = _token.balanceOf(address(this));
56 require(amount > 0, "L: no tokens");
57
58 _token.safeTransfer(donor, amount);
59 emit Revoke(donor, amount);
60 }
```

```
160 function revokeLockedToken(LockedToken _lockToken) public onlyRole(
SYSTEM_ROLE) {
161       _lockToken.revoke();
162 }
```

Any compromise to the account may allow a hacker to take advantage of this authority and withdraw the beneficiary's locked tokens.

Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level ofdecentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefullymanage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommendcentralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accountswith enhanced security practices, e.g., multisignature wallets. Indicatively, here are some feasible suggestions that wouldalso mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign (⅔, ⅗) combination mitigate by delaying the sensitive operation and avoiding a single point of keymanagement failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, mitigate by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Recommendation

Permanent:

Renouncing the ownership or removing the function can be considered fully resolved.

- Renounce the ownership and never claim back the privileged roles;
  OR
- Remove the risky functionality.

Disclaimer

This report is based on the scope of materials and documentation provided for a limited review at the time provided. Results may not be complete nor inclusive of all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Blockchain technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. A report does not indicate the endorsement of any particular project or team, nor guarantee its security. No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate.

FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

Appendix

Finding Categories

Centralization / Privilege
Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

Coding Style
Coding Style findings usually do not affect the generated bytecode but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

Volatile Code
Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

Logical Issue
Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block. timestamp works.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux “sha256sum" command against the target file.

About

DeHacker is a team of auditors and white hat hackers who perform security audits and assessments. With decades of experience in security and distributed systems, our experts focus on the ins and outs of system security. Our services follow clear and prudent industry standards. Whether it's reviewing the smallest modifications or a new platform, we'll provide an in-depth security survey at every stage of your company's project. We provide comprehensive vulnerability reports and identify structural inefficiencies in smart contract code, combining high-end security research with a real-world attacker mindset to reduce risk and harden code.

BLOCKCHAIINS

- Ethereum
- Cosmos
- Eos
- Substrate

TECH STACK

- Python
- Solidity
- Rust
- c++

CONTACTS

- https://dehacker.io
- https://twitter.com/dehackerio
- https://github.com/dehacker/audits_public
- https://t.me/dehackerio
- https://blog.dehacker.io/

DeHacker

July 31th 2024