$ ▮

Competitive Security Assessment

QnA3_CheckinSolana

Apr 22nd, 2024

Secure3

secure3.io

Summary

This report is prepared for the project to identify vulnerabilities and issues in the smart contract source code. A group of NDA covered experienced security experts have participated in the Secure3's Audit Contest to find vulnerabilities and optimizations. Secure3 team has participated in the contest process as well to provide extra auditing coverage and scrutiny of the finding submissions.

The comprehensive examination and auditing scope includes:

- Cross checking contract implementation against functionalities described in the documents and white paper disclosed by the project owner.

- Contract Privilege Role Review to provide more clarity on smart contract roles and privilege.

- Using static analysis tools to analyze smart contracts against common known vulnerabilities patterns.

- Verify the code base is compliant with the most up-to-date industry standards and security best practices.

- Comprehensive line-by-line manual code review of the entire codebase by industry experts.

The security assessment resulted in findings that are categorized in four severity levels: Critical, Medium, Low, Informational. For each of the findings, the report has included recommendations of fix or mitigation for security and best practices.

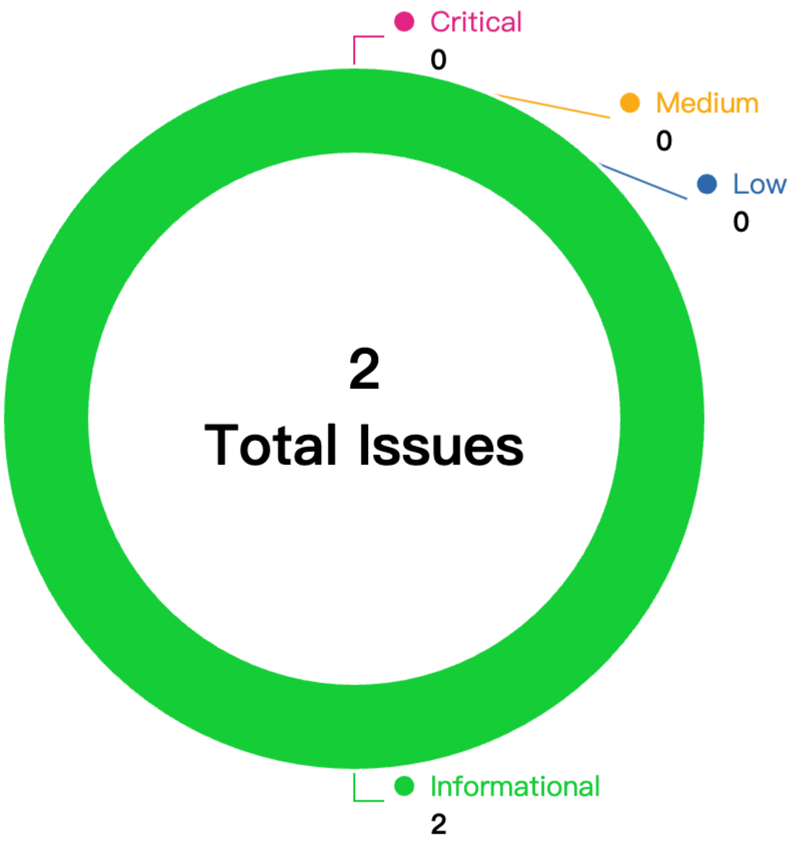
Overview

Project Name	QnA3_CheckinSolana
Language	Rust
Codebase	• shared by file • audit version - 91b9b221f37e691b9f70ced128dfdfbd0e050140 • final version - 91b9b221f37e691b9f70ced128dfdfbd0e050140
Audit Methodology	• Audit Contest • Business Logic and Code Review • Privileged Roles Review • Static Analysis

Audit Scope

File	SHA256 Hash
QnACheckinSolana.rs	91b9b221f37e691b9f70ced128dfdfbd0e050140

Code Assessment Findings



ID	Name	Category	Severity	Client Response	Contributor
QAS-1	Unused Variable	Logical	Informational	Acknowledged	0xCO2
QAS-2	The number of check-ins recorded is always one day less than the actual number	Logical	Informational	Acknowledged	xyzqwe123

QAS-1:Unused Variable

Category	Severity	Client Response	Contributor
Logical	Informational	Acknowledged	0xCO2

Code Reference

- code/QnACheckinSolana.rs#L9

```
9: pub fn checkin(ctx: Context<DailyCheckIn>, _user: Pubkey) -> Result<()> {
```

Description

0xCO2: In function `` `checkin` ``, the variable `` `_user` `` never be used, which is a `` `Pubkey` `` type. It may cause unexpected results.

Recommendation

0xCO2: It is recommended to remove the variable `` `_user` ``.

Client Response

client response for 0xCO2: Acknowledged - It may be used in the future.

QAS-2:The number of check-ins recorded is always one day less than the actual number

Category	Severity	Client Response	Contributor
Logical	Informational	Acknowledged	xyzqwe123

Code Reference

- code/QnACheckinSolana.rs#L12

```
12: let today = clock.unix_timestamp / 86400; // 将当前时间戳转换为天数
```

Description

xyzqwe123: The check-in timestamp calculation is flawed due to the nature of integer division in Rust, which adheres to the truncation rules. When performing integer division, the operation inherently disregards the decimal portion of the result, potentially leading to a loss of precision. To illustrate, if the current time corresponds to the fifth day, the expression 400000 / 86400 would yield 4 for the variable today, rather than the correct 5.

Recommendation

xyzqwe123: The code is changed as follows

```
#[account]
pub struct CheckInAccount {
    pub last_check_in: i64, // Store the last check-in day, where the first day is represented as 0
    pub is_first_check_in: bool, // Flag whether the user is checking in for the first time
}

pub fn checkin(ctx: Context<DailyCheckIn>, _user: Pubkey) -> Result<()> {
    let check_in_account = &mut ctx.accounts.check_in_account;
    let clock = Clock::get()?;
    if check_in_account.is_first_check_in {
        check_in_account.last_check_in = 0; // Set the timestamp of the first day to 0
        check_in_account.is_first_check_in = false; // Mark the user as having checked in before
    } else {
        let today = clock.unix_timestamp / 86400; // Convert the current timestamp to days
        check_in_account.last_check_in = today;
    }
    Ok(())
}
```

Client Response

client response for xyzqwe123: Acknowledged - The calculation of the check-in days will be based on the number of times a user interacts with the contract on the chain.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Invoices, or the scope of services, and terms and conditions provided to you (“Customer” or the “Company”) in connection with the Invoice. This report provided in connection with the services set forth in the Invoices shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Invoice. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without Secure3’s prior written consent in each instance.

This report is not an “endorsement” or “disapproval” of any particular project or team. This report is not an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Secure3 to perform a security assessment. This report does not provide any warranty or guarantee of free of bug of codes analyzed, nor do they provide any indication of the technologies, business model or legal compliancy.

This report should not be used in any way to make decisions around investment or involvement with any particular project. Instead, it represents an extensive assessing process intending to help our customers increase the quality of their code and high-level consistency of implementation and business model, while reducing the risk presented by cryptographic tokens and blockchain technology.

Secure3’s position on the final decisions over blockchain technologies and corresponding associated transactions is that each company and individual are responsible for their own due diligence and continuous security.

The assessment services provided by Secure3 is subject to dependencies and under continuing development. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.