verichains

SECURITY AUDIT OF

SING SING MOBILE API, SING SING WALLET

SingSing

Public Report

Apr 19, 2023

Verichains Lab

info@verichains.io

https://www.verichains.io

Driving Technology > Forward

ABBREVIATIONS

Name	Description
Ethereum	An open source platform based on blockchain technology to create and distribute smart contracts and decentralized applications.
Ether (ETH)	A cryptocurrency whose blockchain is generated by the Ethereum platform. Ether is used for payment of transactions and computing services in the Ethereum network.
Smart contract	A computer protocol intended to digitally facilitate, verify or enforce the negotiation or performance of a contract.
Solidity	A contract-oriented, high-level language for implementing smart contracts for the Ethereum platform.
Solc	A compiler for Solidity.
ERC20	ERC20 (BEP20 in Binance Smart Chain or *x*RP20 in other chains) tokens are blockchain-based assets that have value and can be sent and received. The primary difference with the primary coin is that instead of running on their own blockchain, ERC20 tokens are issued on a network that supports smart contracts such as Ethereum or Binance Smart Chain.

EXECUTIVE SUMMARY

This Security Audit Report was prepared by Verichains Lab on Apr 19, 2023. We would like to thank the SingSing for trusting Verichains Lab in auditing smart contracts. Delivering high-quality audits is always our top priority.

This audit focused on identifying security flaws in code and the design of the Sing Sing mobile API, Sing Sing wallet. The scope of the audit is limited to the source code files provided to Verichains. Verichains Lab completed the assessment using manual, static, and dynamic analysis techniques.

During the audit process, the audit team had identified some vulnerable issues in the smart contract code, along with some recommendations. SingSing team has resolved and updated most of the issues following our recommendations.

TABLE OF CONTENTS

1. MANAGEMENT SUMMARY

1.1. About Sing Sing mobile API, Sing Sing wallet

SingSing is a social singing platform, that connects SUPERFANS with their favorite SINGERS to build a new music economy together on blockchain.

Sing Sing Wallet and Sing Sing Mobile API are the two main products of the Sing Sing project. The Sing Sing wallet project utilizes the Django framework and is written in the Python programming language.

The project involves interaction with both the database and the blockchain, and it offers features commonly found in a web3 wallet. The Sing Sing mobile API project is written in the `Typescript` programming language and utilizes the `NestJS` framework.

The project is a RESTful API that provides a set of endpoints for the Sing Sing platform on mobile.

1.2. Audit scope

This audit focused on identifying security flaws in code and the design of Sing Sing mobile API, Sing Sing wallet.

It was conducted on commit `9ab9345620679d5f184d17443a8c363bbfcf4f06` from git repository *https://github.com/greentornado/singsing-verichains-mobileapi* and commit `1c2ecf995e363711e2b5b0aede76acd25e7866db` from git repository *https://github.com/greentornado/singsing-verichains-singwallet*.

1.3. Audit methodology

Our security audit process for smart contract includes two steps:

- Smart contract codes are scanned/tested for commonly known and more specific vulnerabilities using public and RK87, our in-house smart contract security analysis tool.
- Manual audit of the codes for security issues. The contracts are manually analyzed to look for any potential problems.

Following is the list of commonly known vulnerabilities that were considered during the audit of the smart contract:

- Integer Overflow and Underflow
- Timestamp Dependence
- Race Conditions
- Transaction-Ordering Dependence

- DoS with (Unexpected) revert
- DoS with Block Gas Limit
- Gas Usage, Gas Limit and Loops
- Redundant fallback function
- Unsafe type Inference
- Reentrancy
- Explicit visibility of functions state variables (external, internal, private and public)
- Logic Flaws

For vulnerabilities, we categorize the findings into categories as listed in table below, depending on their severity level:

SEVERITY LEVEL	DESCRIPTION
CRITICAL	A vulnerability that can disrupt the contract functioning; creates a critical risk to the contract; required to be fixed immediately.
HIGH	A vulnerability that could affect the desired outcome of executing the contract with high impact; needs to be fixed with high priority.
MEDIUM	A vulnerability that could affect the desired outcome of executing the contract with medium impact in a specific scenario; needs to be fixed.
LOW	An issue that does not have a significant impact, can be considered as less important.

Table 1. Severity levels

1.4. Disclaimer

Please note that security auditing cannot uncover all existing vulnerabilities, and even an audit in which no vulnerabilities are found is not a guarantee for a 100% secure smart contract. However, auditing allows discovering vulnerabilities that were unobserved, overlooked during development and areas where additional security measures are necessary.

2. AUDIT RESULT

2.1. Overview

The Sing Sing wallet project utilizes the Django framework and is written in the Python programming language. The project involves interaction with both the database and the blockchain, and it offers features commonly found in a web3 wallet. The Sing Sing mobile API project is written in the `Typescript` programming language and utilizes the `NestJS` framework. The project is a RESTful API that provides a set of endpoints for the Sing Sing backend.

2.2. Findings

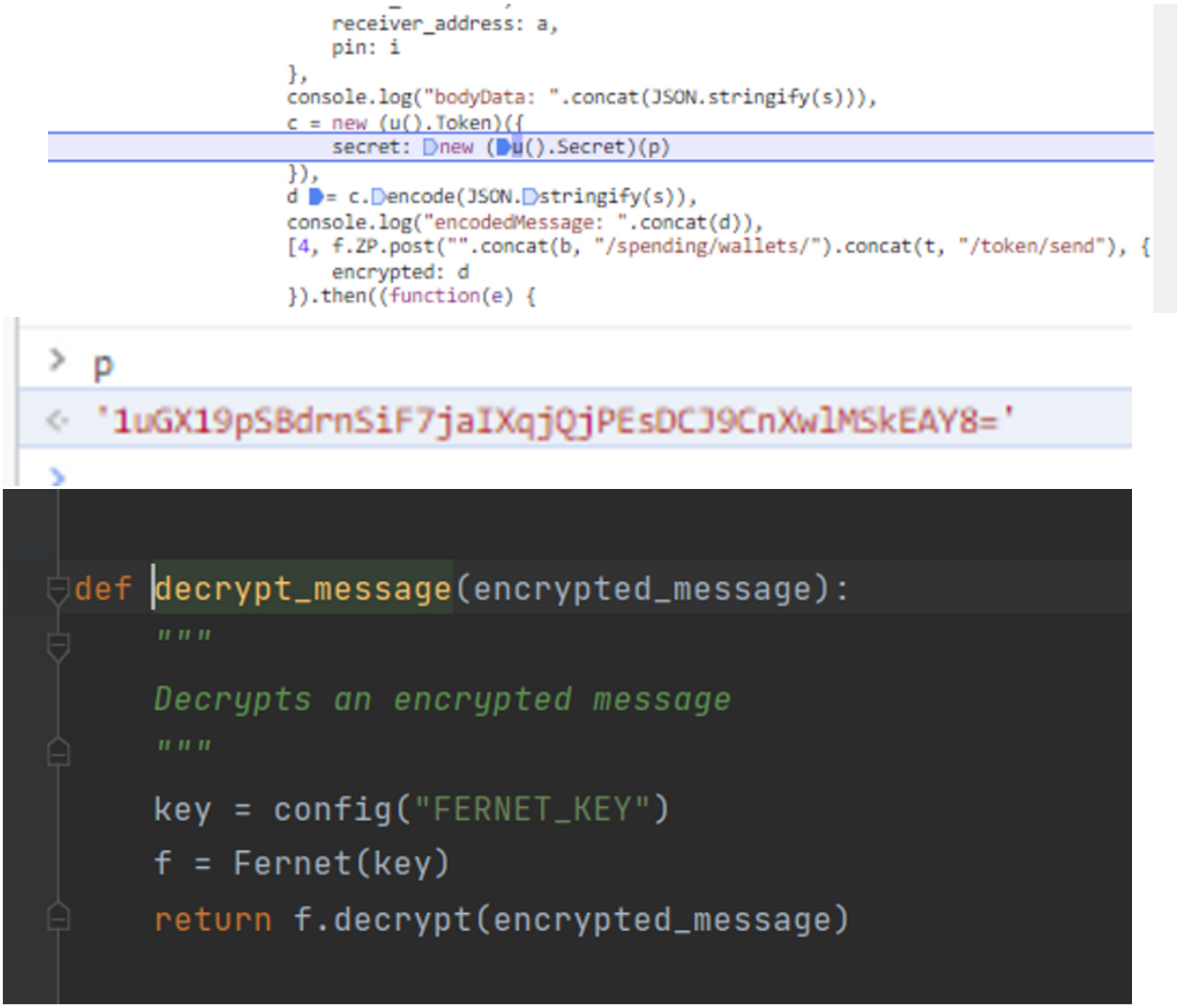
During the audit process, the audit team found some vulnerabilities in the given version of Sing Sing mobile API, Sing Sing wallet.

SingSing fixed the code, according to Verichains's draft report, in commit `9ab9345620679d5f184d17443a8c363bbfcf4f06` (mobileapi) and `1c2ecf995e363711e2b5b0aede76acd25e7866db` (singwallet).

2.2.1. Sing Wallet - Useless crypto mechanic and leak symmetric key MEDIUM

The API *`wallets/{wallet_id}/token/send`* employs the symmetric encryption algorithm to encrypt the payload. However, this method is rendered ineffective as users can easily intercept the transmission and modify the data before it is encrypted.

Furthermore, the symmetric key is also revealed during debugging, which can cause additional problems if it is used in other locations.

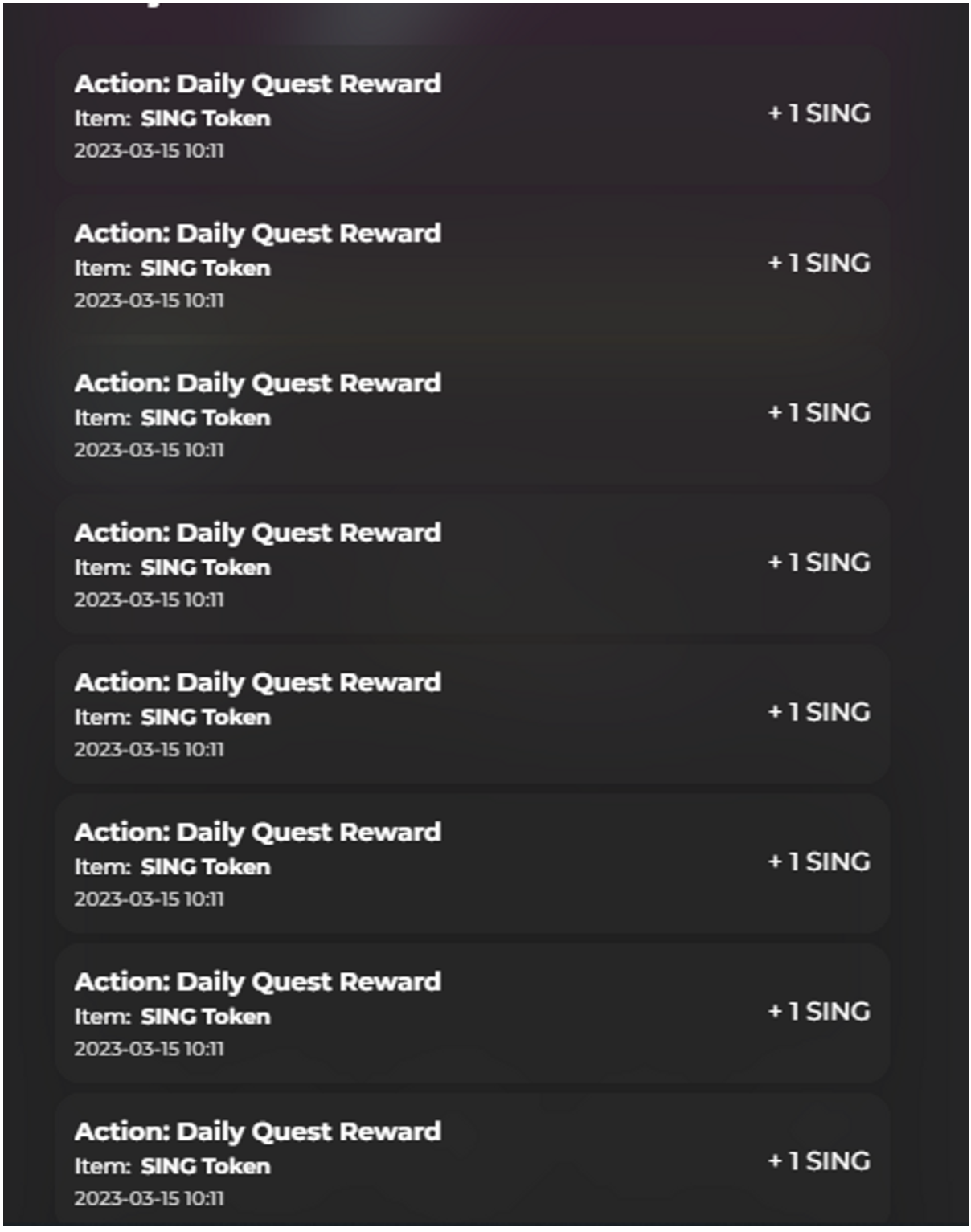


UPDATES

- *Apr 19, 2023*: This issue has been acknowledged by SingSing team.

2.2.2. Sing Wallet - Race Condition in Claim Daily Reward CRITICAL

The API `claim-reward` enables users to obtain a daily reward. However, if users make multiple requests simultaneously, they can deceive the system into giving them rewards more than once per day.

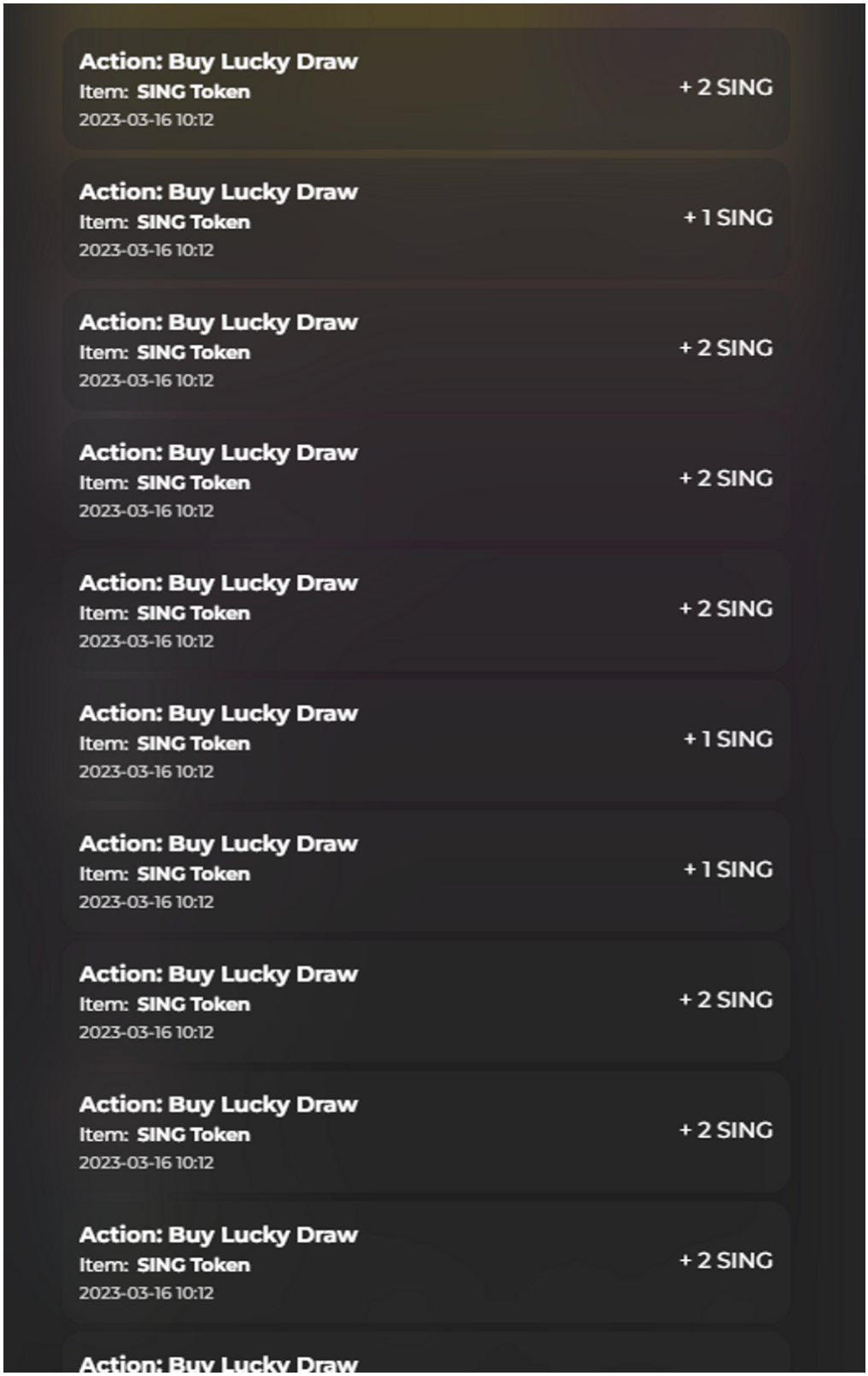

Action: Daily Quest Reward
Item: SING Token
2023-03-15 10:11
+1 SING
Action: Daily Quest Reward
Item: SING Token
2023-03-15 10:11
+1 SING
Action: Daily Quest Reward
Item: SING Token
2023-03-15 10:11
+1 SING
Action: Daily Quest Reward
Item: SING Token
2023-03-15 10:11
+1 SING
Action: Daily Quest Reward
Item: SING Token
2023-03-15 10:11
+1 SING
Action: Daily Quest Reward
Item: SING Token
2023-03-15 10:11
+1 SING
Action: Daily Quest Reward
Item: SING Token
2023-03-15 10:11
+1 SING
Action: Daily Quest Reward
Item: SING Token
2023-03-15 10:11
+1 SING

UPDATES

- *Apr 19, 2023*: This issue has been acknowledged and fixed by SingSing team.

2.2.3. Sing Wallet - Race condition in LuckyDraw, draw greater than 50 times per day CRITICAL

The LuckyDraw feature shares a similar issue with Claim Daily Reward in that users can send multiple draw requests simultaneously, potentially deceiving the system into allowing more draws than the set time restriction and without paying the draw fee.


Action: Buy Lucky Draw
Item: SING Token
2023-03-16 10:12
+ 2 SING
Action: Buy Lucky Draw
Item: SING Token
2023-03-16 10:12
+ 1 SING
Action: Buy Lucky Draw
Item: SING Token
2023-03-16 10:12
+ 2 SING
Action: Buy Lucky Draw
Item: SING Token
2023-03-16 10:12
+ 2 SING
Action: Buy Lucky Draw
Item: SING Token
2023-03-16 10:12
+ 2 SING
Action: Buy Lucky Draw
Item: SING Token
2023-03-16 10:12
+ 1 SING
Action: Buy Lucky Draw
Item: SING Token
2023-03-16 10:12
+ 1 SING
Action: Buy Lucky Draw
Item: SING Token
2023-03-16 10:12
+ 2 SING
Action: Buy Lucky Draw
Item: SING Token
2023-03-16 10:12
+ 2 SING
Action: Buy Lucky Draw
Item: SING Token
2023-03-16 10:12
+ 2 SING

UPDATES

- *Apr 19, 2023*: This issue has been acknowledged and fixed by SingSing team.

2.2.4. Sing Wallet - Mistakes in get_spending_wallet_asset_by_symbol function MEDIUM

When the session is not provided, the function misses to filter the `chainID` parameter.

```
if session:
    statement = select(SpendingWalletAssets) \
        .join(SpendingWalletAssets.contract) \
        .join(SpendingWalletAssets.wallet) \
        .options(joinedload(SpendingWalletAssets.contract, innerjoin=True),
                 joinedload(SpendingWalletAssets.wallet, innerjoin=True)) \
        .filter(SpendingWallets.user_id == user_id) \
        .filter(SpendingContractInfos.symbol == symbol.upper()) \
        .filter(SpendingContractInfos.chain_id == chain_id)

    print(f"get_spending_wallet_asset_by_symbol statement: {statement} {chain_id} {symbol} {user_id}")

    result = session.execute(statement).scalars().first()

    print(f"get_spending_wallet_asset_by_symbol result: {result}")

    return result

else:
    with Session(cockroach_engine) as session:
        session.begin()

        statement = select(SpendingWalletAssets)\
            .join(SpendingWalletAssets.contract)\
            .join(SpendingWalletAssets.wallet)\
            .options(joinedload(SpendingWalletAssets.contract, innerjoin=True),
                     joinedload(SpendingWalletAssets.wallet, innerjoin=True)) \
            .filter(SpendingWallets.user_id == user_id) \
            .filter(SpendingContractInfos.symbol == symbol.upper())
```

UPDATES

- *Apr 19, 2023*: This issue has been acknowledged and fixed by SingSing team.

2.2.5. Sing Wallet - RaceCondition in SwapP2P feature CRITICAL

The `SwapP2P` feature also encounters a problem similar to `Claim Daily Reward` when a user approves another user's offer. If the user sends multiple `approve` requests at the same time, the order may execute multiple times, causing a potential issue.

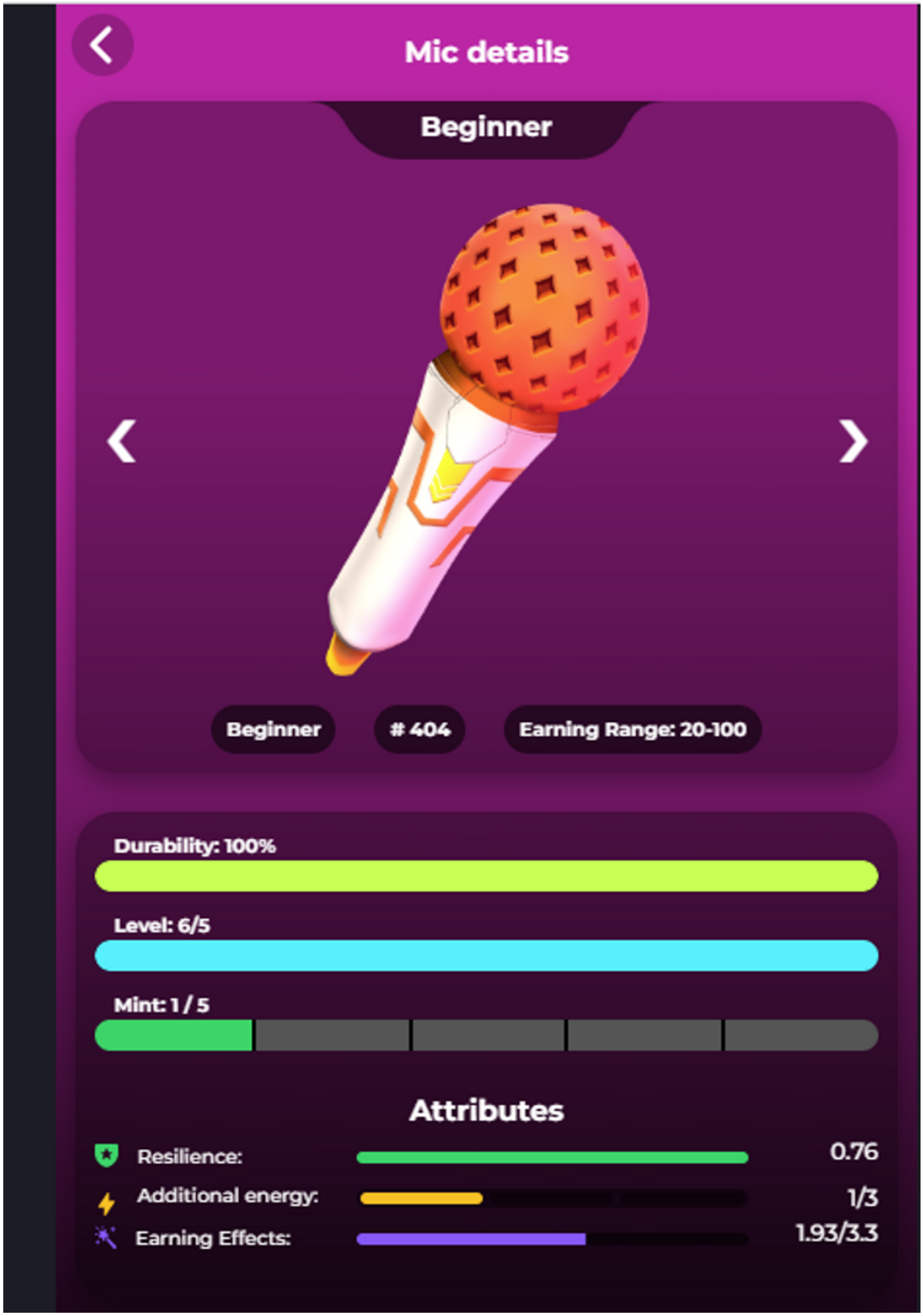

Action: Swap P2P
Item: SING Token
From: artmoney306@gmail.com
To: onekencs@gmail.com
2023-03-23 18:15
+ 0.1 SING
Action: Swap P2P
Item: BUSD Token
From: onekencs@gmail.com
To: artmoney306@gmail.com
2023-03-23 18:15
- 0.1 BUSD
Action: Swap P2P
Item: SING Token
From: artmoney306@gmail.com
To: onekencs@gmail.com
2023-03-23 18:15
+ 0.1 SING
Action: Swap P2P
Item: BUSD Token
From: onekencs@gmail.com
To: artmoney306@gmail.com
2023-03-23 18:15
- 0.1 BUSD
Action: Swap P2P
Item: SING Token
From: artmoney306@gmail.com
To: onekencs@gmail.com
2023-03-23 18:15
+ 0.1 SING
Action: Swap P2P
Item: BUSD Token
From: onekencs@gmail.com
- 0.1 BUSD

UPDATES

- *Apr 19, 2023*: This issue has been acknowledged and fixed by SingSing team.

2.2.6. Mobile API - Race condition in upgrade mic CRITICAL

Similar to the `Claim Daily Reward` feature, the `upgrade` item feature also faces a problem where if a user sends multiple `upgrade` requests at the same time, the item may level up beyond the level restriction.


Mic details
Beginner
Beginner
404
Earning Range: 20-100
Durability: 100%
Level: 6/5
Mint: 1 / 5
Attributes
Resilience:
0.76
Additional energy:
1/3
Earning Effects:
1.93/3.3

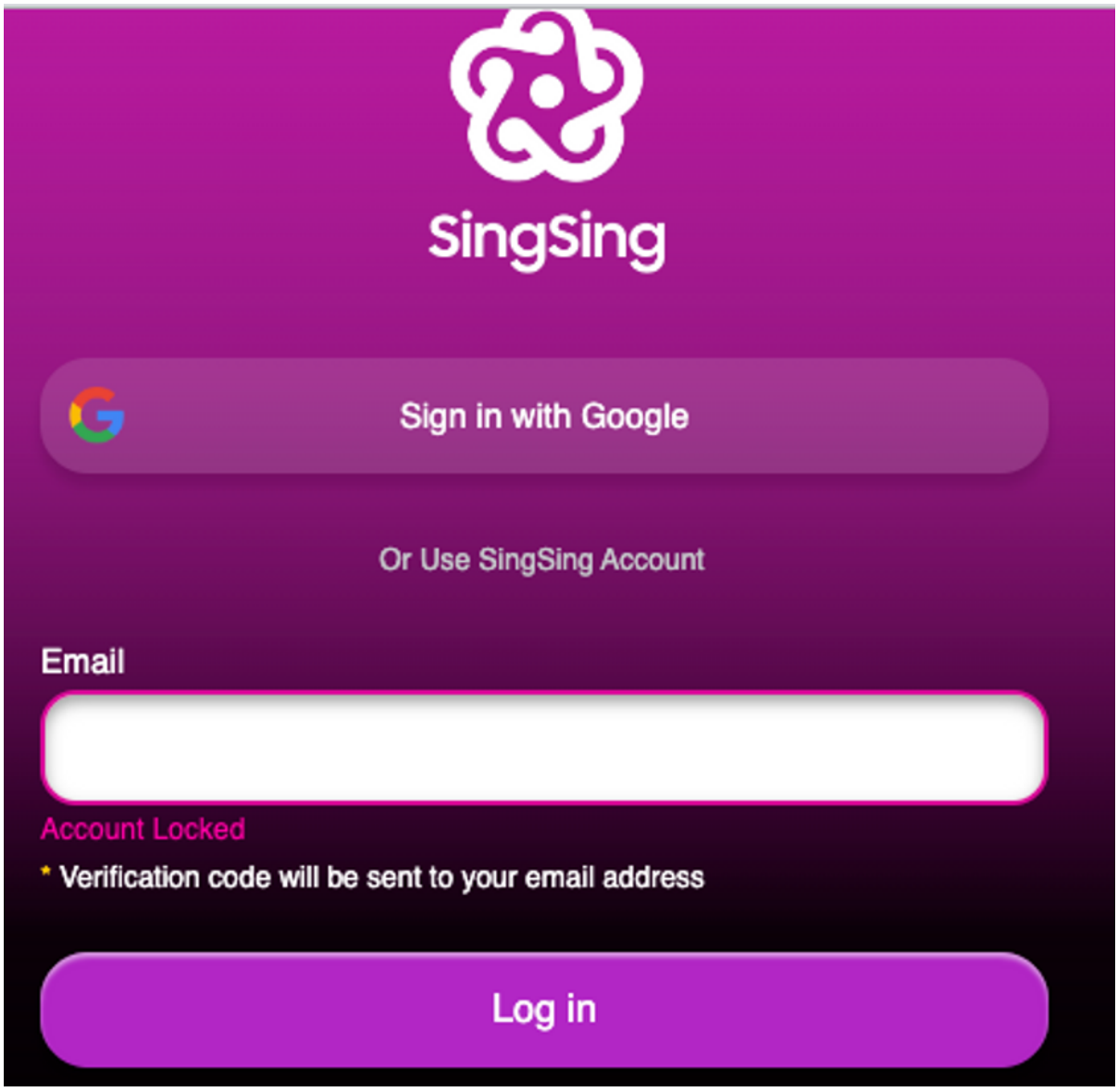
UPDATES

- *Apr 19, 2023*: This issue has been acknowledged by SingSing team. It's noted that the feature is only available on the test server

2.2.7. Mobile API - Locking any user account CRITICAL

Upon successful email login, the system will send a verification code to the user, which they must enter to access their account. In the event of five consecutive failed code attempts, the system will automatically lock the user's account. This presents a significant vulnerability that malicious actors could exploit to intentionally lock legitimate users out of the system.

As a result, this issue poses a severe threat to user experience and the overall reputation of the system. Therefore, it is imperative that this vulnerability be addressed immediately to safeguard the system's integrity and protect user privacy.



RECOMMENDATION

We advise the team to implement a mechanism which prevents users from locking their accounts by entering the wrong code. For example, the system can lock the user's account for a certain period of time after five consecutive failed attempts and send an email to the user to let them know they're been locked out. The only way for the user to unlock their account after that is by clicking the link in the email.

UPDATES

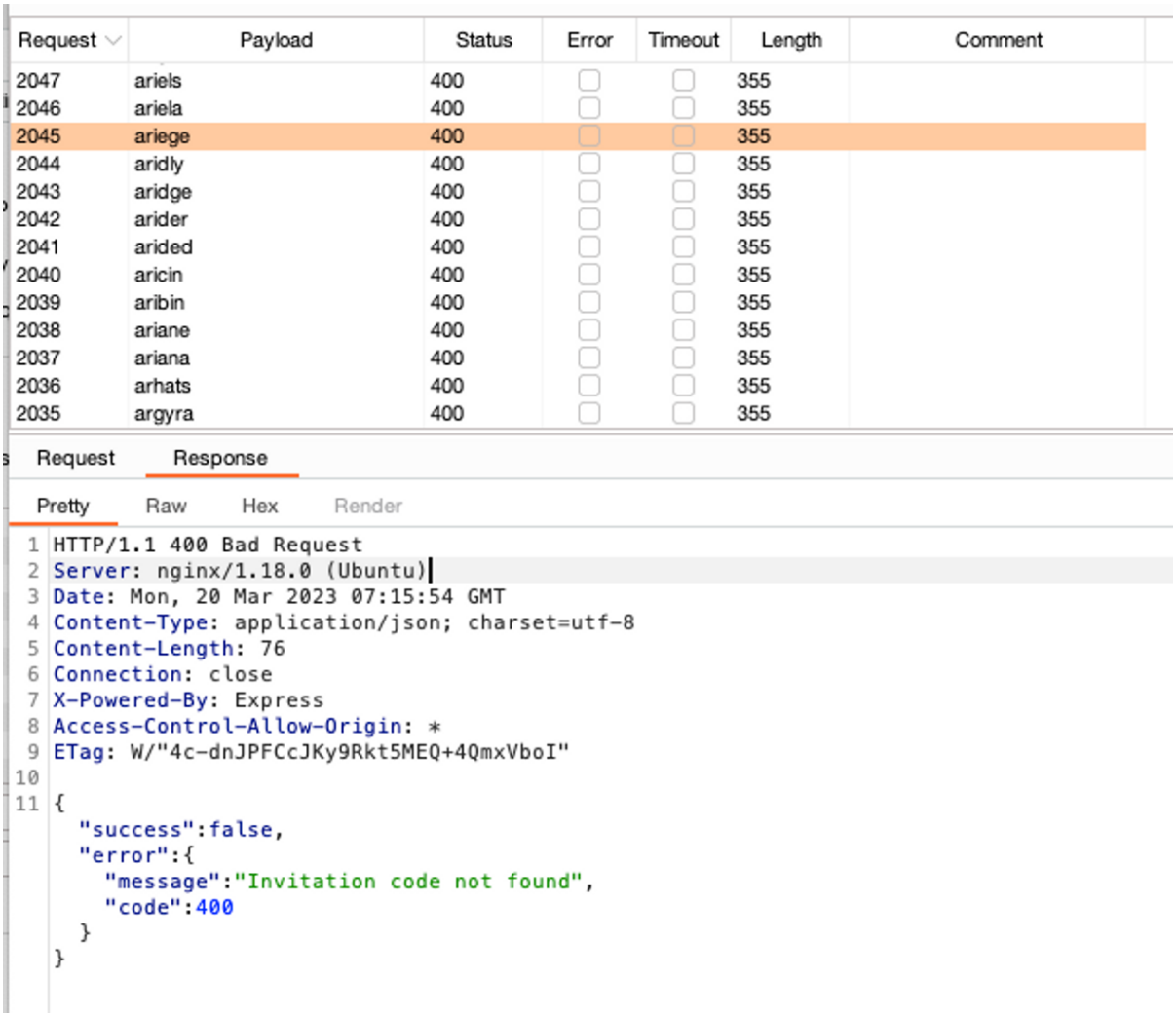
- *Apr 19, 2023*: This issue has been acknowledged by SingSing team.

2.2.8. Mobile API - Old API is not disabled that able to brute force active code of a user CRITICAL

The API endpoint "/user/active-code" is used to activate a user account by providing a four-character code. If the code is present in the white list codes, the user is given a free NFT.

However, the code is easy to guess, and the endpoint has been replaced with a new API "/user/v2/active-code", but the old has not been disabled.

This vulnerability that attackers can exploit by brute-forcing the activation code of user account to obtain a free NFT.



Request	Payload	Status	Error	Timeout	Length	Comment
2047	ariels	400			355	
2046	ariela	400			355	
2045	ariege	400			355	
2044	aridly	400			355	
2043	aridge	400			355	
2042	arider	400			355	
2041	arided	400			355	
2040	aricin	400			355	
2039	aribin	400			355	
2038	ariane	400			355	
2037	ariana	400			355	
2036	arhats	400			355	
2035	argyra	400			355	

Request Response

Pretty Raw Hex Render

```
1 HTTP/1.1 400 Bad Request
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Mon, 20 Mar 2023 07:15:54 GMT
4 Content-Type: application/json; charset=utf-8
5 Content-Length: 76
6 Connection: close
7 X-Powered-By: Express
8 Access-Control-Allow-Origin: *
9 ETag: W/"4c-dnJPFCcJKy9Rkt5MEQ+4QmxVboI"
10
11 {
    "success":false,
    "error":{
      "message":"Invitation code not found",
      "code":400
    }
  }
```

RECOMMENDATION

- Disable the old API endpoint
- To make the activation codes more challenging to guess.

UPDATES

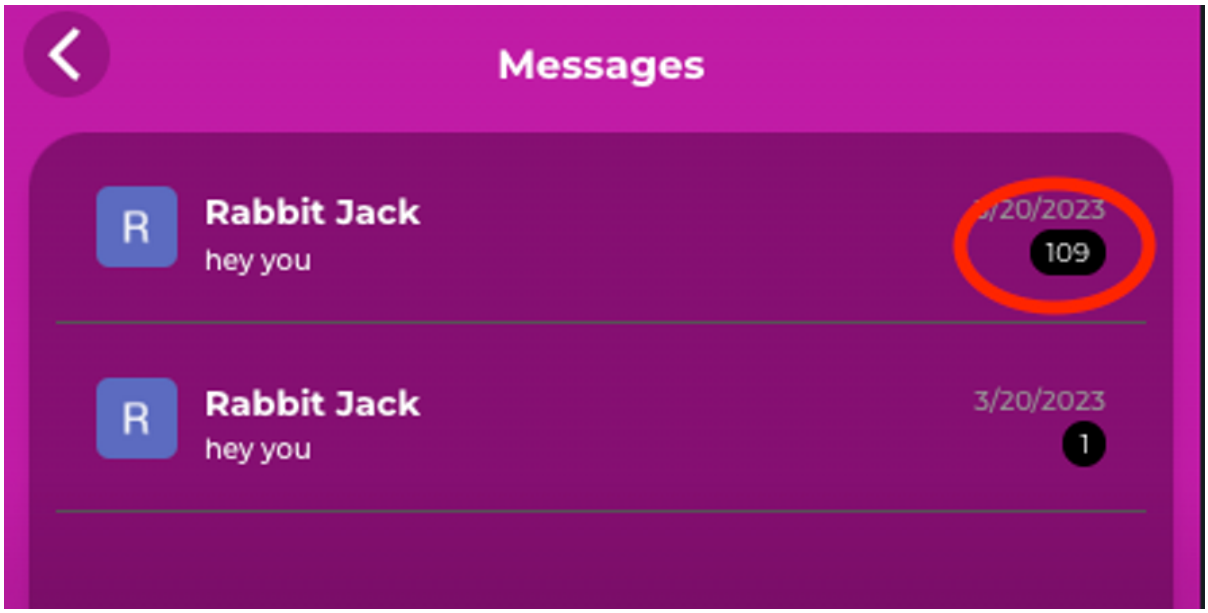
- *Apr 19, 2023*: This issue has been acknowledged and fixed by SingSing team.

2.2.9. Mobile API - Spam message when the system does not have a mechanism to prevent it HIGH

The API endpoint `chat/send` is used to send a message to a user who is a friend of the sender. However, the API does not check whether the current user and the friend user are indeed friends, leaving the system vulnerable to spam messages.

This can pose a significant security risk to the system, phishing attacks, spamming, and malware infections that can compromise sensitive information and lead to fraudulent activities

The vulnerability can also have a negative impact on the business reputation.



RECOMMENDATION

We advise the team to add a guard to check whether the current user and the friend user are indeed friends before sending a message.

UPDATES

- *Apr 19, 2023*: This issue has been acknowledged by SingSing team.

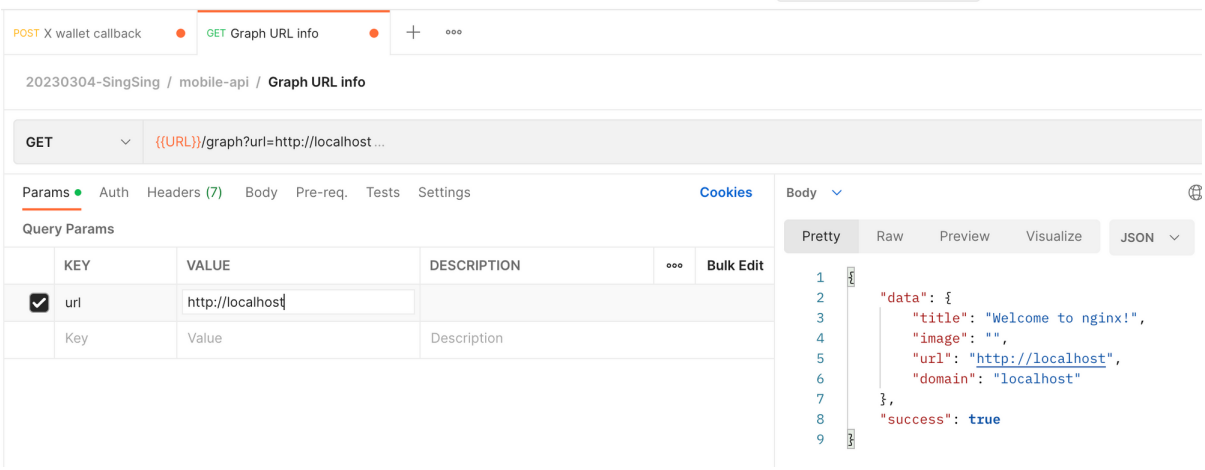
2.2.10. Mobile API - The module Graph does not validate the input raise a SSRF vulnerability that bypass whitelist IP mechanism HIGH

The *graph/query* API endpoint is used to obtain the title and description of a third-party service.

However, the API does not validate the input and uses the input to make a request to a third-party service.

The vulnerability can be exploited by an attacker to perform a variety of attacks, including:

- Discover internal services
- Bypass whitelist IP mechanism
- Scanning internal ports
- And, is a factor that leads to another critical vulnerability.



RECOMMENDATION

We advise the team whitelist the IP address of the third-party service and validate the input before making a request to the third-party service.

UPDATES

- *Apr 19, 2023*: This issue has been acknowledged by SingSing team.

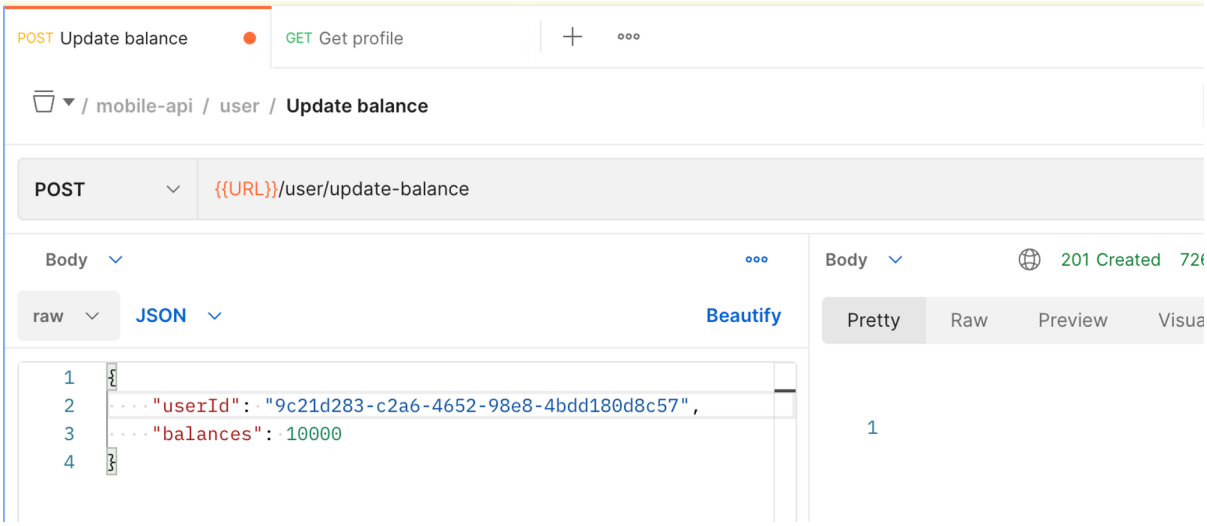
2.2.11. Mobile API - Update balance of any user that emits event to a third party service LOW

The *user/update-balance* API endpoint is used to update the balance of a user in the cache and emit an event to a third-party service. The balance of the user is used to calculate the ranking of the user in a YGG SEA system (the feature under development).

However, the API does not validate the balance parameter, allowing attackers to update the balance of any user and affect the ranking of the user. The security impact of this vulnerability cannot be detected exactly because the feature is still under development.

If the vulnerability is exploited, it can result in the following consequences:

- Manipulation of the ranking of users in the YGG SEA system.
- Damage to the integrity and fairness of the YGG SEA system.



RECOMMENDATION

To mitigate this vulnerability, it is recommended that the '/user/update-balance' API endpoint validates the userId parameter which cannot is modified by users before updating the balance

UPDATES

- *Apr 19, 2023*: This issue has been acknowledged and fixed by SingSing team.

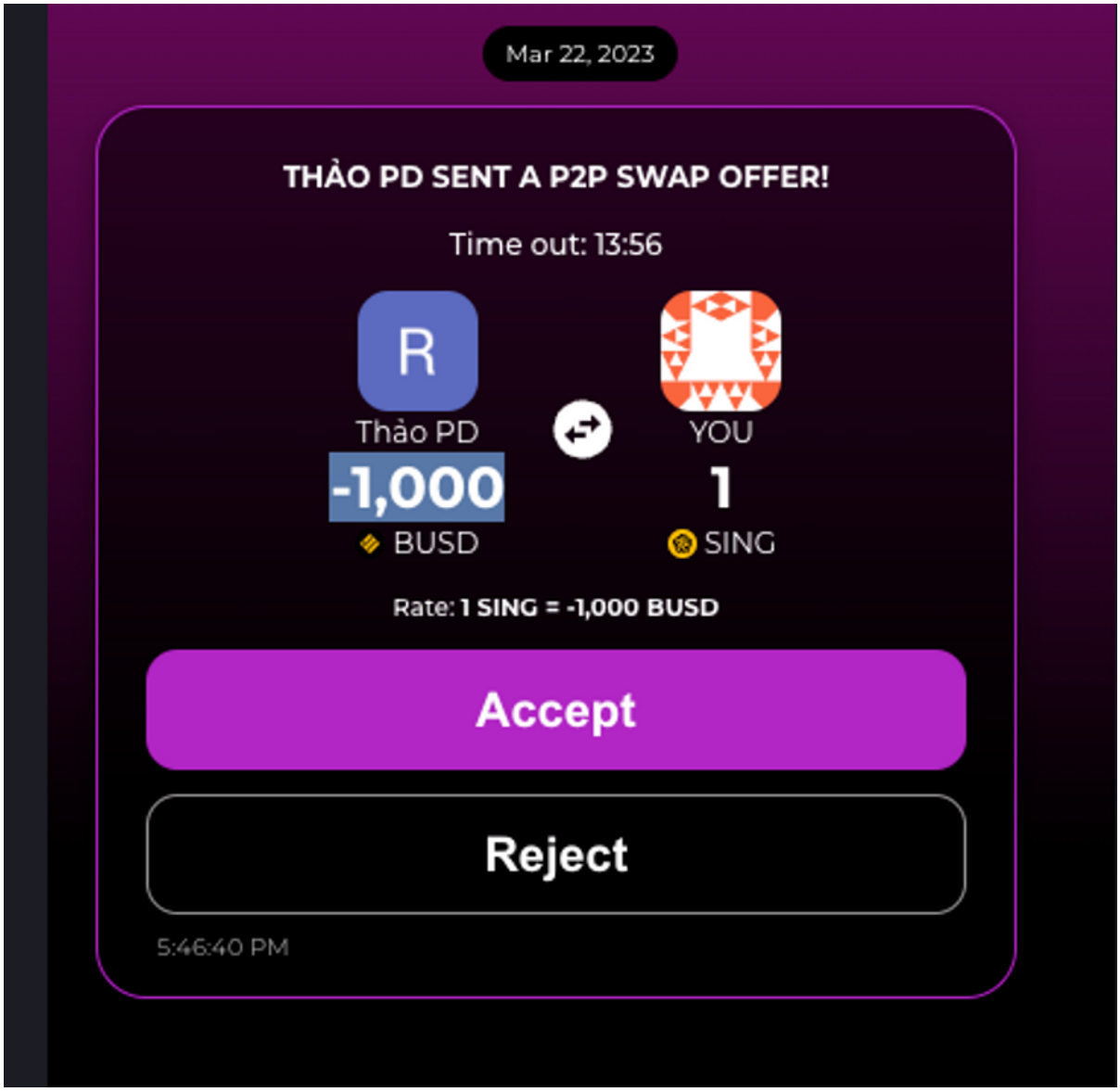
2.2.12. Mobile API - A Swap request in Transaction module does not validate negative amount LOW

The *transaction/swap/request* API endpoint is used to send a swap request to another user to exchange a token on the P2P market.

However, the API does not validate the amount parameter, allowing attackers to send a swap request with a negative amount.

But this vulnerability not impact to the system because another API /transaction/swap/confirm validated the amount parameter.

Although the vulnerability does not have an immediate impact on the system, it is a logical error that should be addressed.



RECOMMENDATION

To mitigate this vulnerability, it is recommended that the '/transaction/swap/request' API endpoint should ensure that the amount parameter is positive before saving the swap request to database.

UPDATES

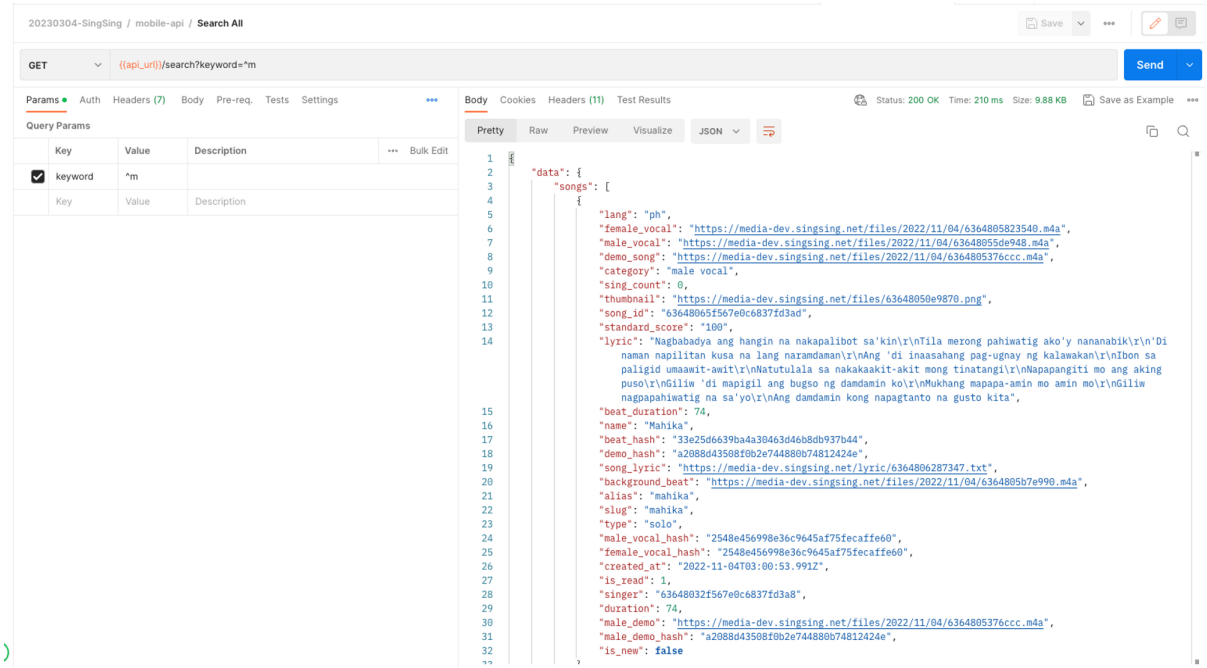
- *Apr 19, 2023*: This issue has been acknowledged and fixed by SingSing team.

2.2.13. Mobile API - A Search API return a plenty unnecessary information INFORMATIVE

The *search* API endpoint is used to search for users by name, username, and email, and songs by name and alias.

The API returns a result of search with information of users, which are restricted to some fields using the `getUserDetail` function.

However, with songs in the results, the API returns plenty of unnecessary information.



RECOMMENDATION

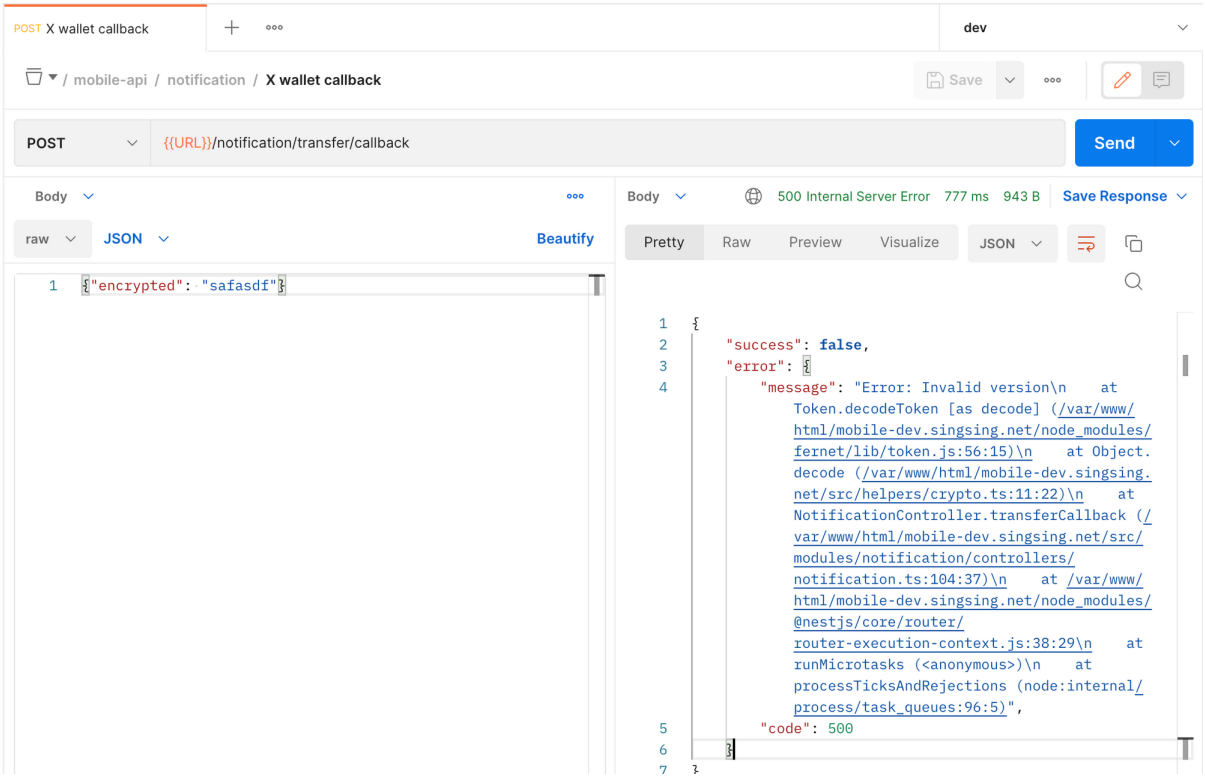
We recommend that the team should use an existing function `utils.getSongDetail` to restrict unnecessary song information.

UPDATES

- *Apr 19, 2023*: This issue has been acknowledged by SingSing team.

2.2.14. Mobile API - Web root disclosure when APIs meet an error INFORMATIVE

When APIs encounter issues and respond with a stack trace that includes a web root, the application does not address the fault.



RECOMMENDATION

We recommend that the team should disable debug mode on the server backend API

UPDATES

- *Apr 19, 2023*: This issue has been acknowledged by SingSing team.

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Apr 19, 2023*	Public Report	Verichains Lab

Table 2. Report versions history