Snap Application Security Audit Report

Table Of Contents

1 Executive Summary

2 Audit Methodology

3 Project Overview

3.1 Project Introduction

3.2 Vulnerability Information

3.3 Vulnerability Summary

4 Audit Result

5 Statement

1 Executive Summary

On 2023.11.10, the SlowMist security team received the TrendMicro team's security audit application for Chainsafer Snap, developed the audit plan according to the agreement of both parties and the characteristics of the project, and finally issued the security audit report.

The SlowMist security team adopts the strategy of "black/grey box lead, white box assists" to conduct a complete security test on the project in the way closest to the real attack.

The test method information:

Test method	Description
Black box testing	Conduct security tests from an attacker's perspective externally.
Grey box testing	Conduct security testing on code modules through the scripting tool, observing the internal running status, mining weaknesses.
White box testing	Based on the open source code, non-open source code, to detect whether there are vulnerabilities in programs such as nodes, SDK, etc.

The vulnerability severity level information:

Level	Description
Critical	Critical severity vulnerabilities will have a significant impact on the security of the project, and it is strongly recommended to fix the critical vulnerabilities.
High	High severity vulnerabilities will affect the normal operation of the project. It is strongly recommended to fix high-risk vulnerabilities.
Medium	Medium severity vulnerability will affect the operation of the project. It is recommended to fix medium-risk vulnerabilities.
Low	Low severity vulnerabilities may affect the operation of the project in certain scenarios. It is suggested that the project team should evaluate and consider whether these vulnerabilities need to be fixed.
Weakness	There are safety risks theoretically, but it is extremely difficult to reproduce in engineering.
Suggestion	There are better practices for coding or architecture.

2 Audit Methodology

The security audit process of SlowMist security team for the application includes two steps:

- The codes are scanned/tested for commonly known and more specific vulnerabilities using automated analysis tools.
- Manual audit of the codes for security issues. The application is manually analyzed to look for any potential issues.

The following is a list of security audit items considered during an audit:

Serial Number.	Audit Items
1	Snaps user interface security audit
2	Snaps permissions security audit
3	Insecure entropy source audit
4	Cryptography security audit
5	Cross-Site Scripting security audit
6	Third-party components security audit
7	Communication encryption security audit
8	Business design security audit
9	Architecture design security audit
13	Web API security audit
14	DNSSEC security audit
15	SSL/TLS security audit

3 Project Overview

3.1 Project Introduction

Trend Micro ChainSafer provides advanced blockchain security with AI-powered algorithms that proactively detect and alert against suspicious activities. Leveraging Trend Micro's cybersecurity database, Trend Micro ChainSafer ensures your online transactions are safe and of the highest quality.

Audit Version

app-metasec-chainsafer-snap.zip(SHA256):

605a2487e9a3956a19b91c46a55dbf5f5e788a2ec9b693f448d6c19939dfe010

Fixed Version

app-metasec-chainsafer-snap_2023_11_20.zip(SHA256):

b3f4b25f56d78e6c0dba3b3252fd8cd0ccc7591992a8a04586da6afc46e3dbd8

3.2 Vulnerability Information

The following is the status of the vulnerabilities found in this audit:

NO	Title	Category	Level	Status
N1	an issue in obtaining ChainID	Business design security audit	Low	Fixed
N2	wrong prompt	Business design security audit	Suggestion	Fixed
N3	Improve the detection rules	Others	Suggestion	Fixed

3.3 Vulnerability Summary

[N1] [Low] an issue in obtaining ChainID

Category: Business design security audit

Content

There is an error in the code implementation of obtaining network_id when chainId.split(':')[1].length != 2 and

chainId.split(':')[1] != 1 Then the chainId obtained is the wrong chainId, For example, Goerli's ChainID is 5, but the ChainID obtained according to the following code logic is 1.

Code location: app-metasec-chainsafer-snap/src/controllers/chainsafer.ts

```
export const postTransactionSimulation = async (
    chainId: string,
    transaction: Json
): Promise<[IPostTransactionSimulationResponseParsed, IResponseError]> => {
    let result: IPostTransactionSimulationResponseParsed =
        {} as IPostTransactionSimulationResponseParsed
    let error: IResponseError = null

    let payload = {
        network_id: chainId.split(':')[1].length == 2 ? chainId.split(':')[1] : '1',
        from: transaction['from'] || '',
        to: transaction['to'] || '',
        call_data: transaction['data'] || '',
        value: parseInt(transaction['value'], 16) || 0,
        gas: parseInt(transaction['gas'], 16) || 0,
    } as IPostTransactionSimulationRequestPayload

    try {
        result = await pgw.postTransactionSimulation(payload)
    } catch (e) {
        logger.error(`${JSON.stringify(e)}`)
        error = e
    }

    return [result, error]
}
```

Solution

It is recommended to use the "supported chainId list" to judge chainId.split(':')[1] to avoid obtaining the wrong chainId.

Status

Fixed

[N2] [Suggestion] wrong prompt

Category: Business design security audit

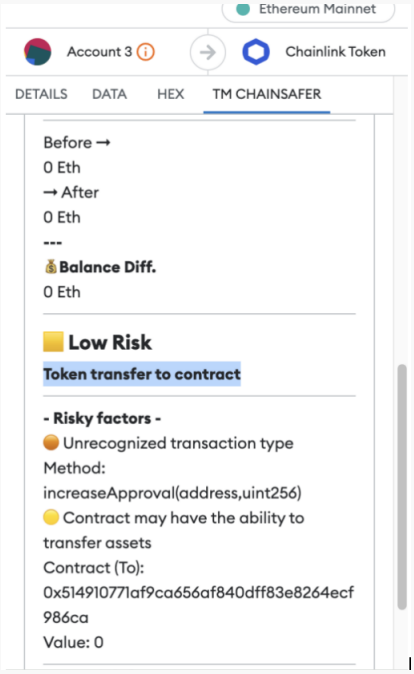
Content

The prompt of convertToSimulationPanel is wrong. When transferring ERC20 token or native coin, the prompt is about NFT.

Code location: app-metasec-chainsafer-snap/src/helpers/panels/simulationPanel.ts

```
    return panel([
        text(`**You're about to buy a NFT via a smart contract.**`),
        divider(),
        ...paymentDetailPanel,
        ...tokenChangePanel,
        ...contractPanel,
        ...balanceChangePanel,
    ])
```

This is an increaseApproval operation, but it is recognized as Token transfer to contract.

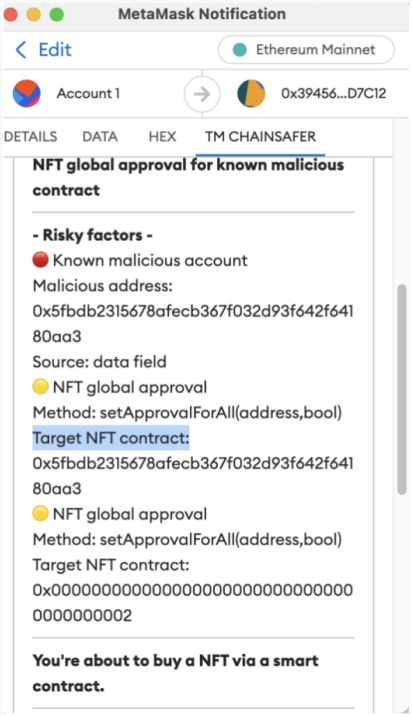


PoC: increaseApproval(address,uint256)

```
window.ethereum.request({"method":"eth_sendTransaction","params":[{"from":<Your wallet address>,"maxPriorityFeePerGas":null,"maxFeePerGas":null,"data":"0xd73dd623000000000000000000000003b6878c8f6e9ba26e0ee1a938a35b2d21763d9e800000000000000000000000000000000000000000000000038d7ea4c68000","to":"0x514910771af9ca656af840dff83e8264ecf986ca"}]})
```

Data of "target nft contract" is wrong, The address 0x5fbdb2315678afecb367f032d93f642f64180aa3 is a malicious address that has been marked as such. It is being used as the value for the "to" parameter of the setApprovalForAll

function, instead of the target NFT contract.



PoC: setApprovalForAll(address,bool)

```
window.ethereum.request({"method":"eth_sendTransaction","params":[{"from":<Your wallet address>,"maxPriorityFeePerGas":null,"maxFeePerGas":null,"data":"0xa22cb465000000000000000000000005fbdb2315678afecb367f032d93f642f64180aa3000000000000000000000000000000000000000000000000000000000000000001","to":"0x769272677fab02575e84945f03eca517acc544cc"}]})
```

Solution

It is recommended to check the prompt to ensure that the information of the prompt is correctly identified and displayed.

Status

Fixed

[N3] [Suggestion] Improve the detection rules

Category: Others

Content

There are three common functions to increase the amount of allowance, but currently only approve (address, uint256) are supported and can be bypassed.

PoC: increaseAllowance(address,uint256)

```
window.ethereum.request({"method":"eth_sendTransaction","params":[{"from":<Your
wallet
address>,"maxPriorityFeePerGas":null,"maxFeePerGas":null,"data":"0x395093510000000000
00000000000003b6878c8f6e9ba26e0ee1a938a35b2d21763d9e800000000000000000000000000000000
0000000000000000038d7ea4c68000","to":"0xa0b86991c6218b36c1d19d4a2e9eb0ce3606eb48"}]})
```

PoC: increaseApproval(address,uint256)

```
window.ethereum.request({"method":"eth_sendTransaction","params":[{"from":<Your
wallet
address>,"maxPriorityFeePerGas":null,"maxFeePerGas":null,"data":"0xd73dd62300000000000
00000000000003b6878c8f6e9ba26e0ee1a938a35b2d21763d9e800000000000000000000000000000000
0000000000000000038d7ea4c68000","to":"0x514910771af9ca656af840dff83e8264ecf986ca"}]})
```

PoC: approve(address,uint256)

```
window.ethereum.request({"method":"eth_sendTransaction","params":[{"from":<Your
wallet
address>,"maxPriorityFeePerGas":null,"maxFeePerGas":null,"data":"0x095ea7b300000000000
00000000000003b6878c8f6e9ba26e0ee1a938a35b2d21763d9e800000000000000000000000000000000
0000000000000000038d7ea4c68000","to":"0x514910771af9ca656af840dff83e8264ecf986ca"}]})
```

Code location: app-metasec-chainsafer-snap/src/constants/content.ts

```
transaction_risks: {
        factor_url_blocklist: 'Known malicious website',
        factor_url_ai_scam: 'AI-flagged suspicious website',
        factor_address_blocklist: 'Known malicious account',
        factor_eth_sign: 'High-risk signature type',
        factor_contract_not_public: 'Unverifiable private contract',
        factor_uncategorized_signature: 'Unrecognized transaction type',
        factor_erc20_transfer: 'ERC-20 token transfer',
        factor_erc20_approve: 'ERC-20 token approval',
        factor_eip712_transfer: 'Signature Request',
        factor_erc721_setapprovalforall: 'NFT global approval',
        factor_ssl_short_create: 'Recent SSL certificate',
        factor_ssl_short_available: 'SSL certificate expires soon',
        factor_domain_short_create: 'Recent website domain',
        factor_domain_short_available: 'Short-term domain validity',
        factor_p2p_transfer: 'Transfer to account',
        factor_contract_transfer: 'Transfer to contract',
```

```
        factor_ssl_domain_mismatch: 'SSL certificate-domain mismatch',
        factor_payable_contract_transfer: 'Contract may have the ability to transfer
assets',
        factor_withdraw_ape_coin: 'Withdraw Ape Coin',
    },
```

Solution

It is recommended to add the following detection rules for increaseApproval(address,uint256),

increaseAllowance(address,uint256).

Status

Fixed

4 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002311140001	SlowMist Security Team	2023.11.10 - 2023.11.14	Passed

Summary conclusion: The SlowMist security team uses a manual and SlowMist team's analysis tool to audit the project, during the audit work we found 1 low-risk vulnerabilities, 2 suggestions, and all issues have been fixed.

5 Statement

SlowMist issues this report with reference to the facts that have occurred or existed before the issuance of this report, and only assumes corresponding responsibility based on these.

For the facts that occurred or existed after the issuance, SlowMist is not able to judge the security status of this project, and is not responsible for them. The security audit analysis and other contents of this report are based on the documents and materials provided to SlowMist by the information provider till the date of the insurance report (referred to as "provided information"). SlowMist assumes: The information provided is not missing, tampered with, deleted or concealed. If the information provided is missing, tampered with, deleted, concealed, or inconsistent with the actual situation, the SlowMist shall not be liable for any loss or adverse effect resulting therefrom. SlowMist only conducts the agreed security audit on the security situation of the project and issues this report. SlowMist is not responsible for the background and other conditions of the project.

Official Website

www.slowmist.com

E-mail

team@slowmist.com

Twitter

@SlowMist_Team

Github

https://github.com/slowmist