

## Summary

**Auditors:** 0xWeiss (Marc Weiss)

**Client:** Sybil Samurai

**Report Delivered:** 8<sup>th</sup> November, 2023

## Protocol Summary

Protocol Name	Sybil Samurai
Language	Solidity
Codebase	<a href="https://github.com/alexflorence/ss-scoping">https://github.com/alexflorence/ss-scoping</a>
Commit	1ca785f4a2ddffcdb54ab3ff9483b96d3be86951
Previous Audits	None






## About 0xWeiss

0xWeiss is an independent security researcher. Having found numerous security vulnerabilities in various defi protocols, he does his best to contribute to the blockchain ecosystem and its protocols by putting time and effort into security research & reviews. Reach out on Twitter @[0xWeiss](#)

## Audit Summary

Sybil Samurai engaged 0xWeiss through Hyacinth to review the security of its codebase. From the 7th of November to the 8th of November, 0xWeiss reviewed the source code in scope. At the end, there were 7 issues identified. All findings have been recorded in the following report. Notice that the examined smart contracts are not resistant to internal exploit. For a detailed understanding of risk severity, source code vulnerability, and potential attack vectors, refer to the complete audit report below.

## Vulnerability Summary

Severity	Total	Pending	Acknowledged	Par. resolved	Resolved
 HIGH	1	0	1	0	0
 MEDIUM	0	0	0	0	0
 LOW	2	0	1	0	1
 INF	3	0	1	0	2
 GOV	1	0	1	0	0

## Audit Scope

ID	File Path
ASCV	contracts/AbstractSamuraiClaimVerifier.sol
STD	contracts/SamuraiTokenDistributor.sol
ISS	contracts/interfaces/ISybilSamurai.sol

## Severity Classification








Severity	Classification
● HIGH	Exploitable, causing loss/manipulation of assets or data.
● MEDIUM	Risk of future exploits that may or may not impact the smart contract execution.
● LOW	Minor code errors that may or may not impact the smart contract execution.
● INF	No impact issues. Code improvement

## Methodology

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross-referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

## Findings and Resolutions

ID	Category	Severity		Status
ASCV-1	Flash Loan		<b>HIGH</b>	Acknowledged
ASCV-2	Incorrect Design		<b>LOW</b>	Resolved
ASCV-3	Incorrect Architecture		<b>LOW</b>	Acknowledged
ASCV-4	Un-used import		<b>INF</b>	Resolved
ISS-1	Un-used import		<b>INF</b>	Resolved
GLOBAL-1	Architectural Error		<b>INF</b>	Acknowledged
ASCV-5	Admin Privileges		<b>GOV</b>	Acknowledged

## ASCV-1 | Rewards can be claimed by flash loaning the NFT

Severity	Category	Status
● HIGH	Flash Loan	Acknowledged

### Description of the issue

Currently anyone can flashloan samurai NFTs and claim the rewards for that specific NFTs without actually owning it. Actual owners of the NFTs will lose the rewards for their Samurais if decided to supply them to protocols that allow for NFT flash loans.

```
if (sybilSamuraiNFT.ownerOf(samuraiId) == msg.sender &&  
!claimProcessed[token][samuraiId]) {  
    claimableTokens +=  
    _tokensPerSamuraiTier[sybilSamuraiNFT.tokenIdToTier(samuraiId)];  
}
```

### Recommendation

Consider building a protection mechanism on the NFT contract so that you can't own the NFT for less than 1 block when planning to add tiered rewards for owning the NFT.

### Resolution

Acknowledged

## ASCV-2 | sybilSamuraiNFT should be immutable.

Severity	Category	Status
● LOW	Incorrect Design	Resolved

### Description of the issue

The sybilSamuraiNFT stands for the address of the samurai NFT collection. This address is set in the constructor and never changes:

```
ISybilSamurai public sybilSamuraiNFT;
```

### Recommendation

Consider making the address immutable.

### Resolution

Fixed

## ASCV-3 | Use a 2-step ownership transfer.

Severity	Category	Status
● LOW	Incorrect Architecture	Acknowledged

### Description of the issue

Sybil Samurai uses a single-step access control transfer pattern. This means that if the current owner account transfers ownership with an incorrect address, then this owner role will be lost forever along with all the functionality that depends on it.

### Recommendation

Follow the pattern from OpenZeppelin's [Ownable2Step](#) and implement a two-step transfer pattern for the action.

### Resolution

Acknowledged

## ASCV-4 | Un-used ERC721 import

Severity	Category	Status
● INF	Un-used import	Resolved

### Description of the issue

The ASCV contract imports the IERC721 interface, and it is never being used:

```
import "@openzeppelin/contracts/token/ERC721/IERC721.sol"; // @audit-issue
INF unused import
```

### Recommendation

Delete the import.

### Resolution

Fixed



## ISS-1 | Un-used ERC20 import

Severity	Category	Status
● INF	Un-used import	Resolved

### Description of the issue

The ISS interface imports the IERC721 interface and it is never being used:

```
import "@openzeppelin/contracts/token/ERC20/IERC20.sol"; // @audit-issue  
Interface not used
```

### Recommendation

Delete the import.

### Resolution

Fixed

## GLOBAL-1 | Transfer-tax tokens are not supported.

Severity	Category	Status
● INF	Architectural Error	Acknowledged

### Description of the issue

The architecture of the codebase has several spots that will not work with transfer-tax tokens. This will break the entire system.

### Recommendation

Consider not using such tokens.

### Resolution

Acknowledged

## ASCV-5 | Impacts of private-key leakage

Severity	Category	Status
● GOV	Governance	Acknowledged

### Description of the issue

The owner role has certain privileges in the codebase that allow them to withdraw all the tokens from the contract and set the rewards for all the NFT tiers.

### Recommendation

Use a strong internal pipeline using multi-signature wallets to manage the key state changes of the codebase.

### Resolution

Acknowledged

## DISCLAIMER

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Marc Weiss to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology. Blockchain technology and cryptographic assets present a high level of ongoing risk.

My position is that each company and individual are responsible for their own due diligence and continuous security. My goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze. Therefore, I do not guarantee the explicit security of the audited smart contract, regardless of the verdict.