SOLID GROUP

X

FOOTBALL STARS.IO

Security Assessment

June 27th, 2021

For:
FootballStars

Disclaimer

SolidGroup reports are not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team. Solid group do not cover testing or auditing the integration with external contract or services (such as Unicrypt, Uniswap, PancakeSwap etc’…)

SolidGroup Audits do not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technology proprietors. SolidGroup Audits should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.

SolidGroup Reports represent an extensive auditing 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. SolidGroup’s position is that each company and individual are responsible for their own due diligence and continuous security. SolidGroup in no way claims any guarantee of security or functionality of the technology we agree to analyze.

About FootballStars

FootballStars Fans are connecting with their heroes like never before. Sports clubs and sporting mega stars are big business. They collect billions worldwide through endorsements and TV deals. It sometimes seems like the fans are being left behind while clubs and players Cash In. the advent of FootballStars means that players and clubs can now reconnect with fans on a global scal

The FootballStars Marketplace will interconnect fans, football players and clubs from around the world, under one platform. FootballStars is a deflationary, community lead hybrid DEFI/NFT project. Aiming to be the personal connection between fans and real life football stars and clubs Having Already Garnished A Vast Array Of Contacts In The Sports Industry, FootballStars Has The Competitive Edge, To Quickly Become The Leading Sports Blockchain Platform.

🌍 Website | 📱 Telegram | 🐦 Twitter | Instagram

About Solid Group

Solid Group is a blockchain consulting and auditing service provider, founded by 3 cybersecurity experts with a passion for thinking out of the box, learning, and sharing knowledge. Every project goes through a meticulous process and is viewed by at least two partners, thereby achieving a high level of credibility and professionalism. Our group is partnered with multiple organizations and launchpads that have a combined market cap of over 400 million USD.

📣Telegram | 🗣Telegram discussion group |🐦 Twitter | 🛡 Contact for audit | 🤖 Audit Checker bot | Medium

Description

The bridge contract on ETH network.

Files in Scope

Contract Name	Contract Address
Initial Revision	https://etherscan.io/address/0x90b0ced725077a935782e09a47807a66420bd5c6

Vulnerability Summary

● Informational severity Issues	2
● Low severity issues	2
● Medium severity issues	2
● High severity issues	4

Privilege Functions

- The owner can withdraw any number of tokens/eth that was sent to the contract by having a consensus among all owners.

- The owner can control the number of tokens that are received by the user on the receiving side of the bridge. Which can be different from the amount that was sent by the user on the other side.

- The bridge is managed by the team. They can close the bridge whenever they like.

General Warnings

- We audit the bridge contract. **The bridge management server is the one that handles the transfer between the two networks. Which was not audited by us.**
  Our recommendation is to audit the code of the bridge management server since there may be found potentially high severity issues.
- FTS token was not initially audited by us, assuming the integration
- The audit covers only the on-chain **contract** on the Ethereum network. the bridge management server has not been audited by us. There is no guarantee that the bridge management server is bug-free **and can't be exploited by a malicious actor.** We did raise our concern regarding the security of the bridge management server.
- Solid Group assumes that the integration with FTS contract was tested and treats the FTS token contract code as Blackbox since it wasn't initially audited by Solid Group. The team was asked to do proper testing for the integration with FTS token contract and especially test the behavior in extreme cases.

Issue #1	Type	Severity	Location	Status
	Volatile Code	● Medium	recieveTokens	❌ Not Fixed

Description

recieveTokens should always work, even if it fails to send commission to one of the owners, to ensure that investors' funds are safe. If the function is critical (such as recieveTokens) always make sure its error cases are handled gracefully!

```
for (uint i = 0; i < owners.length; i++) {
            address payable owner = payable(owners[i]);
            uint256 commission = commissions[i];
            owner.transfer(commission);
   }
```

Recommendation

Use try catch when calling transfer.

Issue #2	Type	Severity	Location	Status
	Logical Issue Gas Optimization	● Informational	recieveTokens	❌ Not Fixed

Description

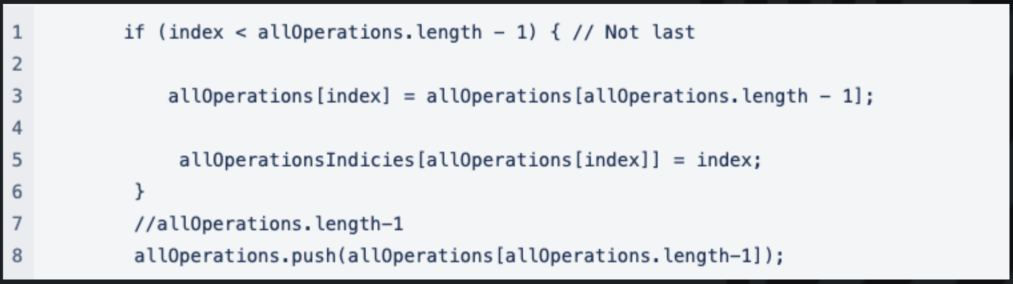
amountToSent should be declared as a local variable to save on gas fees.

```
amountToSent = tokensRecievedButNotSent[msg.sender] - tokensSent[msg.sender];
```

Issue #3	Type	Severity	Location	Status
	Volatile Code	● High	deleteOperation	✗ Not Fixed

Description

This block of code removed allOperations[index] by setting it to the last element. Line 7 (which is commented out) simply deletes the last element which was moved down the array by lines 3 and 4, and instead, you reinsert the last element to the array on line 8



```
1         if (index < allOperations.length - 1) { // Not last
2
3             allOperations[index] = allOperations[allOperations.length - 1];
4
5              allOperationsIndicies[allOperations[index]] = index;
6          }
7          //allOperations.length-1
8          allOperations.push(allOperations[allOperations.length-1]);
```

Recommendation

Remove line 8 and uncomment line 7

Issue #4	Type	Severity	Location	Status
	Volatile Code	● High		✗ Not Fixed

Description

The code is vulnerable to overflow.

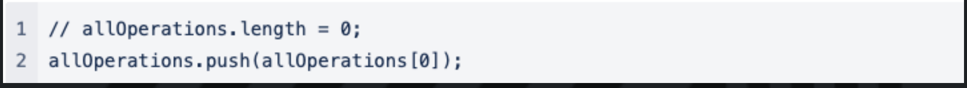
Recommendation

Consider using safemath library

Issue #5	Type	Severity	Location	Status
	Volatile Code	● High	transferOwnershipWithHowMany	✗ Not Fixed

Description

Same as Issue #3

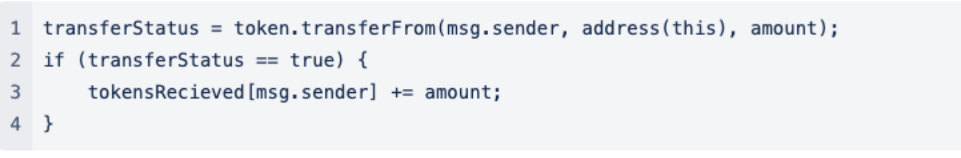


```
// allOperations.length = 0;
allOperations.push(allOperations[0]);
```

Issue #6	Type	Severity	Location	Status
	Best Practice	● Low	transferOwnershipWithHowMany	✗ Not Fixed

Description

It’s customary to revert in case a token transfer failed, to undo any side effects.

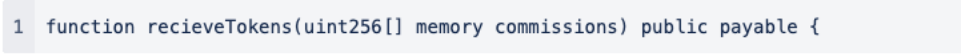


```
transferStatus = token.transferFrom(msg.sender, address(this), amount);
if (transferStatus == true) {
    tokensRecieved[msg.sender] += amount;
}
```

Issue #7	Type	Severity	Location	Status
	Logical Issue	● Medium	recieveTokens	✗ Not Fixed

Description

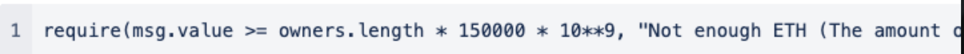
commission can be bypassed by the user.



```
function recieveTokens(uint256[] memory commissions) public payable {
```

Issue #8	Type	Severity	Location	Status
	Logical Issue	● Low	recieveTokens	✗ Not Fixed

Description



```
require(msg.value >= owners.length * 150000 * 10**9, "Not enough ETH (The amount o
```

This magic number is equal to 3.00000218e-7$ according to the current ETH price, which is practically zero.

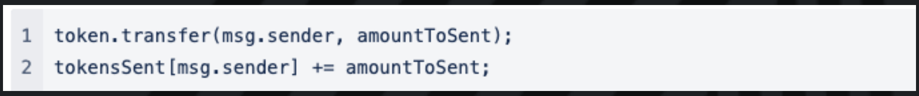
Recommendation

Consider using a sensible minimum or determining the price in a more dynamic way (e.g. specifying the required commission per holder in writeTransaction)

Issue #9	Type	Severity	Location	Status
	Volatile Code	● High	recieveTokens	✗ Not Fixed

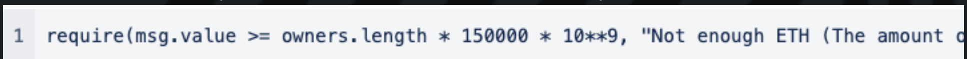
Description

The function recieveTokens should revert if it failed to transfer the tokens to msg.sender.



```
token.transfer(msg.sender, amountToSent);
tokensSent[msg.sender] += amountToSent;
```

tokensSent will be updated as if the tokens were already sent



```
require(msg.value >= owners.length * 150000 * 10**9, "Not enough ETH (The amount o
```

This magic number is equal to 3.00000218e-7$ according to the current ETH price, which is practically zero.

Recommendation

Use require.