CONTRACT WOLF

Security Assessment

FUST Staking Dapp

Verified on 06/30/2025

SUMMARY

Project	CHAIN	METHODOLOGY
FUST	Ethereum	Manual & Automatic Analysis
FILES	**DELIVERY**	**TYPE**
Single	1/11/2024	Dapp Audit

2	0	0	1	0	1	0
Total Findings	Critical	Major	Medium	Minor	Informational	Resolved

🟥 0 Critical	An exposure that can affect the dapp's functions in several events that can risk and disrupt the code
🟧 0 Major	An opening & exposure to manipulate the code in an unwanted manner
🟧 1 Medium	An opening that could affect the outcome in executing the code in a specific situation
⬜ 0 Minor	An opening but doesn't have an impact on the functionality of the code
🟦 1 Informational	An opening that consists information but will not risk or affect the code
🟩 0 Resolved	ContractWolf's findings has been acknowledged & resolved by the project

STATUS ✔ **AUDIT PASSED**

TABLE OF CONTENTS | Staking Dapp

DISCLAIMER | Staking Dapp

ContractWolf audits and reports should not be considered as a form of project's "Advertisement" and does not cover any interaction and assessment from "Project Code" to "External Code"

ContractWolf does not provide any warranty on its released report and should not be used as a decision to invest into audited projects.

ContractWolf provides a transparent report to all its "Clients" and to its "Clients Participants" and will not claim any guarantee of bug-free code within its **DAPP**.

ContractWolf's presence is to analyze, audit and assess the Client's Dapp to find any underlying risk and to eliminate any logic and flow errors within its code.

Each company or project should be liable to its security flaws and functionalities.

SCOPE OF WORK | Staking Dapp

FUST's team has agreed and provided us with the files that need to be tested. The scope of audit is the main dapp.

The goal of this engagement is to identify if there is a possibility of security flaws in the implementation of dapp and its systems.

ContractWolf will be focusing on dapp issues and functionalities along with the project claims from smart contract to their website, whitepaper, repository which has been provided by **FUST**.

AUDITING APPROACH | Staking Dapp

Every line of code along with its functionalities will undergo manual review to check for security issues, quality of logic and dapp scope of inheritance. The manual review will be done by our team that will document any issues that they discovered.

METHODOLOGY

The auditing process follows a routine series of steps :

1. Code review that includes the following :
   - Review of the specifications, sources and instructions provided to ContractWolf to make sure we understand the size, scope and functionality of the DAPP.
   - Manual review of code. Our team will have a process of reading the code line-by-line with the intention of identifying potential vulnerabilities, underlying and hidden security flaws.

2. Testing and automated analysis that includes :
   - Testing the DAPP's function with common test cases and scenarios to ensure that it returns the expected results.

3. Best practices and ethical review. The team will review the dapp with the aim to improve efficiency, effectiveness, clarifications, maintainability, security and control within the dapp.

4. Recommendations to help the project take steps to eliminate or minimize threats and secure the dapp.

DAPP DETAILS | Staking Dapp

FUST is a utility token with standard tokenomics which is part of the FUSD ecosystem. FUSD is an appreciating stable token due to launch in the next few weeks and FUST will be used to mine free FUSD using a staking protocol we are calling the Fusion Miner.

SOURCE

Source	*Sent Via local-files*

FINDINGS | Staking Dapp

Total Findings	Critical	Major	Medium	Minor	Informational	Resolved
2	0	0	1	0	1	0

This report has been prepared to state the issues and vulnerabilities for FUST Dapp through this audit. The goal of this report findings is to identify specifically and fix any underlying issues and errors

Frontend

ID	Title	File & Line #	Severity	Status
DCW-008	Gas Griefing	StakingManager.js L: 76	Informational	• Pending
DCW-019	Incorrect Unstaking Logic	-	Medium	• Pending

PENETRATION ATTACKS | Staking Dapp

Dapp Weakness Classification and Test Cases

ID	Description	Status
DCW-001	Malware Scan	Passed
DCW-002	Phishing	Passed
DCW-003	Missing HTTP Headers	Passed
DCW-004	Valid SSL Certificate	Passed
DCW-005	Firewalls(Drop & Deny)	Passed
DCW-006	Potential SQL Injection	Passed
DCW-007	Framework Version	Passed
DCW-008	Gas Griefing	Informational
DCW-009	Address Approval	Passed
DCW-010	Address Draining	Passed
DCW-011	Insecure API Usage	Passed
DCW-012	Error Handling	Passed
DCW-013	Memory Leak	Passed
DCW-014	Lack of Input Validation	Passed
DCW-015	Potential Backdoor	Passed
DCW-016	Sensitive Data Exposure	Passed
DCW-017	Request Limit	Passed
DCW-018	Overflow or Precision Loss	Passed
DCW-019	Unintended Behavior	Medium

FIXES & RECOMMENDATION

DCW-008 | Gas Griefing (StakingManager.js)

Hard-coding 5M gas may lead to overspending or out-of-gas if network conditions change.

```
      const tx = await this.stakingContract.stake(wei, {
        gasLimit: 5000000,
      });
```

Recommendation:

Remove the explicit `gasLimit` and let Ethers/BNB(or the network) estimate for you or use a small margin above the estimate :

```
  async stake(amount) {
    try {
      // Parse input amount to Wei
      const wei = ethers.utils.parseEther(amount.toString());

      // Step 1: ensure allowance
      const approveTx = await this.fustToken.approve(this.stakingAddress, wei);
      await approveTx.wait();
      this.toast?.success("Approval successful");

      // Step 2: estimate gas for staking
      const estimated = await this.stakingContract.estimateGas.stake(wei);
      // add a 20% buffer
      const gasLimit = estimated.mul(120).div(100);

      // Step 3: execute stake with dynamic gasLimit
      const tx = await this.stakingContract.stake(wei, { gasLimit });
      await tx.wait();
      this.toast?.success("Staked successfully");
    } catch (err) {
      console.error("Staking failed", err);
      this.toast?.error("Staking failed");
      throw err;
    }
  }
```

DCW-019 | Incorrect Unstaking Logic

`unstake()` is called without an amount parameter.
Smart contract will unstake entire balance regardless of input.

Recommendation :
Modify manager & Smart Contract to support partial unstaking or remove the amount and just leave the button altogether.

AUDIT COMMENTS | Staking Dapp

Dapp audit comment for a non-technical perspective

🟢 Project has been marked as SAFE to be interacted with by any EVM wallets *(06-30-2025)*
🟢 DAPP has no backdoors
🟢 DAPP cannot drain wallets via approval

CONTRACTWOLF

Blockchain Security - Smart Contract Audits