Audited by

CredShields

Smart Contract Audit

September 22, 2025 • CONFIDENTIAL

Description

This document details the process and result of the Smart Contract audit performed by CredShields Technologies PTE. LTD. on behalf of BitSafe SRL between September 15th, 2025, and September 18th, 2025. A retest was performed on September 19th, 2025.

Author
Shashank (Co-founder, CredShields) shashank@CredShields.com

Reviewers
Aditya Dixit (Research Team Lead), Shreyas Koli(Auditor), Naman Jain (Auditor), Sanket Salavi (Auditor), Neel Shah (Auditor), Prasad Kuri (Auditor)

Prepared for
BitSafe SRL

Table of Contents

1. Executive Summary --------------------

BitSafe SRL engaged CredShields to perform a smart contract audit from September 15th, 2025, to September 18th, 2025. During this timeframe, four (4) vulnerabilities were identified. **A retest was performed on September 19th, 2025, and all the bugs have been addressed.**

During the audit, Zero (0) vulnerabilities were found with a severity rating of either High or Critical. These vulnerabilities represent the greatest immediate risk to "BitSafe SRL" and should be prioritized for remediation.

The table below shows the in-scope assets and a breakdown of findings by severity per asset. Section 2.3 contains more information on how severity is calculated.

Assets in Scope	Critical	High	Medium	Low	info	Gas	Σ
FUN Token Giveaway Contracts	0	0	0	3	0	1	**4**
	0	**0**	**0**	**3**	**0**	**1**	**4**

Table: Vulnerabilities Per Asset in Scope

The CredShields team conducted the security audit to focus on identifying vulnerabilities in the FUN Token Giveaway Contract's scope during the testing window while abiding by the policies set forth by BitSafe SRL's team.

State of Security

To maintain a robust security posture, it is essential to continuously review and improve upon current security processes. Utilizing CredShields' continuous audit feature allows both BitSafe SRL's internal security and development teams to not only identify specific vulnerabilities but also gain a deeper understanding of the current security threat landscape.

To ensure that vulnerabilities are not introduced when new features are added or code is refactored, we recommend conducting regular security assessments. Additionally, by analyzing the root cause of resolved vulnerabilities, the internal teams at BitSafe SRL can implement both manual and automated procedures to eliminate entire classes of vulnerabilities in the future. By taking a proactive approach, BitSafe SRL can future-proof its security posture and protect its assets.

2. The Methodology --------------------

BitSafe SRL engaged CredShields to perform the FUN Token Giveaway Contract audit. The following sections cover how the engagement was put together and executed.

2.1 Preparation Phase

The CredShields team meticulously reviewed all provided documents and comments in the smart contract code to gain a thorough understanding of the contract's features and functionalities. They meticulously examined all functions and created a mind map to systematically identify potential security vulnerabilities, prioritizing those that were more critical and business-sensitive for the refactored code. To confirm their findings, the team deployed a self-hosted version of the smart contract and performed verifications and validations during the audit phase.

A testing window from September 15th, 2025, to September 18th, 2025, was agreed upon during the preparation phase.

2.1.1 Scope

During the preparation phase, the following scope for the engagement was agreed upon:

IN SCOPE ASSETS
https://github.com/FunTokenHubs/5m-staking-contracts/blob/a2b8c42ac2c1c43f2304e459595bfdd5e2ea09ff

2.1.2 Documentation

Documentation was not required as the code was self-sufficient for understanding the project.

2.1.3 Audit Goals

CredShields employs a combination of in-house tools and thorough manual review processes to deliver comprehensive smart contract security audits. The majority of the audit involves manual inspection of the contract's source code, guided by OWASP's Smart Contract Security Weakness Enumeration (SCWE) framework and an extended, self-developed checklist built from industry best practices. The team focuses on deeply understanding the contract's core logic, designing targeted test cases, and assessing business logic for potential vulnerabilities across OWASP's identified weakness classes.

CredShields aligns its auditing methodology with the OWASP Smart Contract Security projects, including the Smart Contract Security Verification Standard (SCSVS), the Smart Contract Weakness Enumeration (SCWE), and the Smart Contract Secure Testing Guide (SCSTG). These frameworks, actively contributed to and co-developed by the CredShields team, aim to bring consistency, clarity, and depth to smart contract security assessments. By adhering to these OWASP standards, we ensure that each audit is performed against a transparent, community-driven, and technically robust baseline. This approach enables us to deliver structured, high-quality audits that address both common and complex smart contract vulnerabilities systematically.

2.2 Retesting Phase

BitSafe SRL is actively partnering with CredShields to validate the remediations implemented towards the discovered vulnerabilities.

2.3 Vulnerability classification and severity

CredShields follows OWASP's Risk Rating Methodology to determine the risk associated with discovered vulnerabilities. This approach considers two factors - Likelihood and Impact - which are evaluated with three possible values - **Low**, **Medium**, and **High**, based on factors such as Threat agents, Vulnerability factors, and Technical and Business Impacts. The overall severity of the risk is calculated by combining the likelihood and impact estimates.

Overall Risk Severity				
Impact	HIGH	● Medium	● High	● Critical
	MEDIUM	● Low	● Medium	● High
	LOW	● None	● Low	● Medium
		LOW	MEDIUM	HIGH
Likelihood				

Overall, the categories can be defined as described below -

1. Informational

We prioritize technical excellence and pay attention to detail in our coding practices. Our guidelines, standards, and best practices help ensure software stability and reliability. Informational vulnerabilities are opportunities for improvement and do not pose a direct risk to the contract. Code maintainers should use their own judgment on whether to address them.

2. Low

Low-risk vulnerabilities are those that either have a small impact or can't be exploited repeatedly or those the client considers insignificant based on their specific business circumstances.

3. Medium

Medium-severity vulnerabilities are those caused by weak or flawed logic in the code and can lead to exfiltration or modification of private user information. These vulnerabilities can harm the client's reputation under certain conditions and should be fixed within a specified timeframe.

4. **High**

   High-severity vulnerabilities pose a significant risk to the Smart Contract and the organization. They can result in the loss of funds for some users, may or may not require specific conditions, and are more complex to exploit. These vulnerabilities can harm the client's reputation and should be fixed immediately.

5. **Critical**

   Critical issues are directly exploitable bugs or security vulnerabilities that do not require specific conditions. They often result in the loss of funds and Ether from Smart Contracts or users and put sensitive user information at risk of compromise or modification. The client's reputation and financial stability will be severely impacted if these issues are not addressed immediately.

6. **Gas**

   To address the risk and volatility of smart contracts and the use of gas as a method of payment, CredShields has introduced a "Gas" severity category. This category deals with optimizing code and refactoring to conserve gas.

2.4 CredShields staff

The following individual at CredShields managed this engagement and produced this report:

- Shashank, Co-founder CredShields shashank@CredShields.com

Please feel free to contact this individual with any questions or concerns you have about the engagement or this document.

3. Findings Summary -------------------

This chapter contains the results of the security assessment. Findings are sorted by their severity and grouped by asset and OWASP SCWE classification. Each asset section includes a summary highlighting the key risks and observations. The table in the executive summary presents the total number of identified security vulnerabilities per asset, categorized by risk severity based on the OWASP Smart Contract Security Weakness Enumeration framework.

3.1 Findings Overview

3.1.1 Vulnerability Summary

During the security assessment, 4 security vulnerabilities were identified in the asset.

VULNERABILITY TITLE	SEVERITY	SCWE | Vulnerability Type
Excess Interest Accrual Beyond One-Year Lock Duration	Low	Business Logic Issue (SCWE-001)
Admin Can Skip Milestones	Low	Business Logic Issue (SCWE-001)
Admin can Change the Deadline and Alter Users Reward Weights	Low	Business Logic Issue (SCWE-001)
Cheaper Conditional Operators	Gas	Gas Optimization

Table: Findings in Smart Contracts

4. Remediation Status ------------------

BitSafe SRL is actively partnering with CredShields from this engagement to validate the discovered vulnerabilities' remediations. **A retest was performed on August 21st, 2025, and all the issues have been addressed.**
Also, the table shows the remediation status of each finding.

VULNERABILITY TITLE	SEVERITY	REMEDIATION STATUS
Excess Interest Accrual Beyond One-Year Lock Duration	Low	Fixed [September 19, 2025]
Admin Can Skip Milestones	Low	Fixed [September 19, 2025]
Admin can Change the Deadline and Alter Users Reward Weights	Low	Fixed [September 19, 2025]
Cheaper Conditional Operators	Gas	Fixed [September 19, 2025]

Table: Summary of findings and status of remediation

5. Bug Reports -----------------------

Bug ID #L001 [Fixed]

Excess Interest Accrual Beyond One-Year Lock Duration

Vulnerability Type
Business Logic (SC03-LogicErrors)

Severity
Low

Description
The withdrawWithInterest() function is designed to allow users to withdraw their staked tokens with interest if no milestones are reached. Here, lockDuration is derived from the elapsed time since staking. However, if a user waits beyond one year, the calculation continues to accrue interest beyond the intended one-year maximum.From the user's perspective, this may seem fair since they waited longer. But from the protocol's business perspective, staking is only intended to be locked for one year maximum. Allowing indefinite accrual results in unintended payouts that exceed the designed reward schedule.

Affected Code

- https://github.com/FunTokenHubs/5m-staking-contracts/blob/a2b8c42ac2c1c43f2304e459595bfdd5e2ea09ff/contracts/FUNGiveaway.sol#L271-L307

Impacts
Users who delay withdrawal past one year can receive more than the intended annual interest. This creates a misalignment with the protocol's business rules, potentially increasing token outflow.

Remediation
It is recommended to introduce a cap on the lockDuration used in interest calculations. Limit the reward period to a maximum of one year

Retest
This issue has been fixed by introducing a new maxInterestDuration variable and did required check around it.

Bug ID #L002 [Fixed]

Admin Can Skip Milestones

Vulnerability Type
Business Logic (SC03-LogicErrors)

Severity
Low

Description
The manuallyUnlockMilestone() function allows the DEFAULT_ADMIN_ROLE to manually trigger any milestone. While this is intended for administrative flexibility, the function does not enforce sequential milestone triggering. If the admin skips certain milestones and directly triggers a higher milestone, the skipped milestones remain untriggered. Since users only receive rewards from milestones that are triggered, any skipped milestones will result in users losing access to rewards they would otherwise be entitled to. This behavior depends entirely on how the protocol expects milestone management to be performed and may or may not align with business requirements.

Affected Code

- https://github.com/FunTokenHubs/5m-staking-contracts/blob/a2b8c42ac2c1c43f2304e459595bfdd5e2ea09ff/contracts/FUNGiveaway.sol#L380-L409

Impacts
Users may be denied milestone rewards if the admin chooses to skip intermediate milestones. This does not directly compromise funds or allow malicious exploitation by regular users, but it introduces a trust dependency on the admin's correct usage of the function.

Remediation
To align milestone triggering with user reward expectations, enforce sequential milestone unlocking. The contract could require that milestones be triggered in order, without skipping

Retest
This issue has been fixed by introducing a loop for triggering in between milestones.

Bug ID #L003 [Fixed]

Admin can Change the Deadline and Alter Users Reward Weights

Vulnerability Type
Business Logic (SC03-LogicErrors)

Severity
Low

Description
updateDeadline() lets an admin set a new deadline for all milestones at any time. A user's staking weight in lockTokens() is amount * (deadline - block.timestamp). Because weight depends on the deadline, changing it after users have staked changes the economics they staked under. Existing locks keep their stored weight, but the global milestoneEligibleWeight[i] and future users' weights are now based on a different deadline. This can make rewards uneven compared to what users expected at stake time.

Affected Code

- https://github.com/FunTokenHubs/5m-staking-contracts/blob/a2b8c42ac2c1c43f2304e459595bfdd5e2ea09ff/contracts/FUNGiveaway.sol#L420-L430

Impacts
Some users may earn more or less than intended because the reference deadline moved after they staked. This does not create a direct exploit or fund theft, but it can cause fairness complaints and inconsistent reward distribution versus the original program terms.

Remediation
Freeze deadlines once staking has begun, or only allow extending/shortening for future locks. If deadlines must change mid-program, recompute affected aggregates consistently: either re-snapshot user weights using the new deadline for everyone, or store per-lock immutable deadlines at stake time and exclude deadline changes from already staked positions.

Retest
This issue has been fixed by removing the function.

Bug ID #G001 [Fixed]

Cheaper Conditional Operators

Vulnerability Type
Gas Optimization

Severity
Gas

Description
Upon reviewing the code, it has been observed that the contract uses conditional statements involving comparisons with unsigned integer variables. Specifically, the contract employs the conditional operators x != 0 and x > 0 interchangeably. However, it's important to note that during compilation, x != 0 is generally more cost-effective than x > 0 for unsigned integers within conditional statements.

Affected Code

- https://github.com/FunTokenHubs/5m-staking-contracts/blob/a2b8c42ac2c1c43f2304e459595bfdd5e2ea09ff/contracts/GeomeanOracle.sol#L185

Impacts
Employing x != 0 in conditional statements can result in reduced gas consumption compared to using x > 0. This optimization contributes to cost-effectiveness in contract interactions.

Remediation
Whenever possible, use the x != 0 conditional operator instead of x > 0 for unsigned integer variables in conditional statements.

Retest
This issue has been fixed.

6. The Disclosure -----------------------

The Reports provided by CredShields are not an endorsement or condemnation of any specific project or team and do not guarantee the security of any specific project. The contents of this report are not intended to be used to make decisions about buying or selling tokens, products, services, or any other assets and should not be interpreted as such.

Emerging technologies such as Smart Contracts and Solidity carry a high level of technical risk and uncertainty. CredShields does not provide any warranty or representation about the quality of code, the business model or the proprietors of any such business model, or the legal compliance of any business. The report is not intended to be used as investment advice and should not be relied upon as such.

CredShields Audit team is not responsible for any decisions or actions taken by any third party based on the report.

YOUR SECURE FUTURE STARTS HERE

CRED SHiELDS

At CredShields, we're more than just auditors. We're your strategic partner in ensuring a secure Web3 future. Our commitment to your success extends beyond the report, offering ongoing support and guidance to protect your digital assets

Audited by CRED SHiELDS