Adevar Labs

Prepared for:

Tempest Finance

February 09, 2026

Fogo-Lock Audit Report

Version: after fix review

Table of Contents

© 2026 Adevar Labs. All rights reserved.

1. Executive Summary

About Tempest Finance

This protocol aims to implement Fogo-Lock, a vesting and escrow program designed to manage the distribution of tokens over time. The core logic allows creators to lock tokens in an escrow account that unlocks based on a defined start time, cliff, and frequency.

The Tempest Finance team has added two new functions that were audited through this engagement: claim_with_session and create_vesting_escrow_with_session. These additions allow a user to interact with the program using a temporary session signer instead of their main wallet for every transaction.

Audit Summary

The table below summarizes identified vulnerabilities by risk level and remediation status.

Risk Level	Count	Fixed	Acknowledged
Critical	0	0	0
High	0	0	0
Medium	0	0	0
Low	0	0	0

Enhancement opportunities: 3

Overall Security Posture

The Tempest Finance team demonstrated a strong understanding of the Solana Anchor framework and the original Jup-Lock codebase. By limiting modifications to session-based wrapper functions around the existing claim and create vesting escrow operations, the team employed a minimal-footprint development strategy that preserves the battle-tested core logic while enabling new functionality.

After Fix Review Summary

The protocol has implemented two enhancement opportunities: removing an unnecessary mutable account declaration and eliminating a redundant account load to reduce compute unit usage. Additionally, they have acknowledged the session-based events with signer context enhancement.

Before Fix Review Summary

The audit identified no critical, high, medium, or low severity vulnerabilities. Three enhancement opportunities were identified to improve code quality: removing an unnecessary mutable account declaration, enriching session-based events with signer context for better off-chain indexing, and eliminating a redundant account load to reduce compute unit usage.

Launch Recommendations

© 2026 Adevar Labs. All rights reserved.

Before Fix Review Summary

I. Mandatory before launch:

- No mandatory actions required as no critical, high, or medium severity issues were identified.

II. Strongly recommended:

- Consider implementing the 3 enhancement opportunities to improve code quality and reduce compute unit consumption.
- Ensure comprehensive test coverage for the new session-based instructions.

III. Post-Launch Monitoring:

- Monitor transaction logs for session-based operations to ensure proper functionality.
- Set up alerts for unusual patterns in vesting escrow creation or claim activities.

Audit Scope

The audit scope was performed on the diff between the forked commit hash and before-fix review commit hash.

- **Repository:** https://github.com/Tempest-Finance/fogo-locker
- **Before-Fix Review Commit Hash:** 6822157c99253244f330a1fbb27541e31dfc9013
- **After-Fix Review Commit Hash:** c405ebd242141dafbed8476ebe9dc987de9695ab
- **Files/Modules in Scope:**
  - programs/
- **Forked repository:** https://github.com/jup-ag/jup-lock
- **Forked commit hash:** d123ed778262858d6031a7c555424d796c5bef90

© 2026 Adevar Labs. All rights reserved.

2. Assumptions and Considerations

Audit Assumptions

I. Paymaster will not get refunded on escrow close: payer and user_pubkey are separate in the context of handle_create_vesting_escrow_with_session. In handle_close_vesting_escrow, the lamports will unconditionally get refunded to creator (user_pubkey), despite that account potentially not having paid for the escrow creation.

II. Creating a vesting escrow with a Token 2022 mint will DoS: SPL Token 2022 transfers are currently not supported by Fogo. Therefore, any handle_create_vesting_escrow_with_session function call made with an SPL Token 2022 mint will fail.

Trust Assumptions

III. Fogo infrastructure: The Fogo Sessions off-chain infrastructure is trusted to securely manage session lifecycle, including creation, scoping, and expiration.

© 2026 Adevar Labs. All rights reserved.

3. Severity Definitions

Each issue identified in this report is assigned a severity level based on two dimensions: **Impact** and **Likelihood**. These dimensions help project our team's understanding of both the potential consequences of a vulnerability and how likely a vulnerability is to be discovered and exploited in the real world.

Note: Enhancements represent non-blocking improvements–typically usability, observability, or defense-in-depth tweaks that do not pose an immediate asset risk but would improve the product's reliability and/or user experience if implemented.

Impact

Impact reflects the potential consequences of the issue–particularly on **project funds**, **user funds**, and the **availability or integrity** of the protocol.

- **High Impact**: Successful exploitation could result in a complete loss of user or protocol funds, disruption of core protocol functionality, or permanent loss of control over critical components.
- **Medium Impact**: Exploitation could cause significant disruption or partial loss of funds, but not a total compromise. May impact some users or non-core functionality.
- **Low Impact**: The issue has minor or negligible consequences. It may affect edge cases, expose metadata, or degrade performance slightly without putting funds or core logic at serious risk.

Likelihood

Likelihood reflects how easy a vulnerability is to discover and exploit by an attacker, as well as how economically attractive the exploit is to an attacker.

- **High Likelihood**: The vulnerability is trivially exploitable. This means it can be exploited by a wide range of actors without privileged access rights, with minimal capital requirements and low financial risks.
- **Medium Likelihood**: This type of vulnerability can be found and exploited with moderate effort. It might require a significant capital investment, but with manageable financial risk.
- **Low Likelihood**: Exploitation of these vulnerabilities is often technically unfeasible or requires highly specialized conditions. They may require extraordinary effort or a significant financial risk for an attacker, with a high chance of failure and minimal potential return.

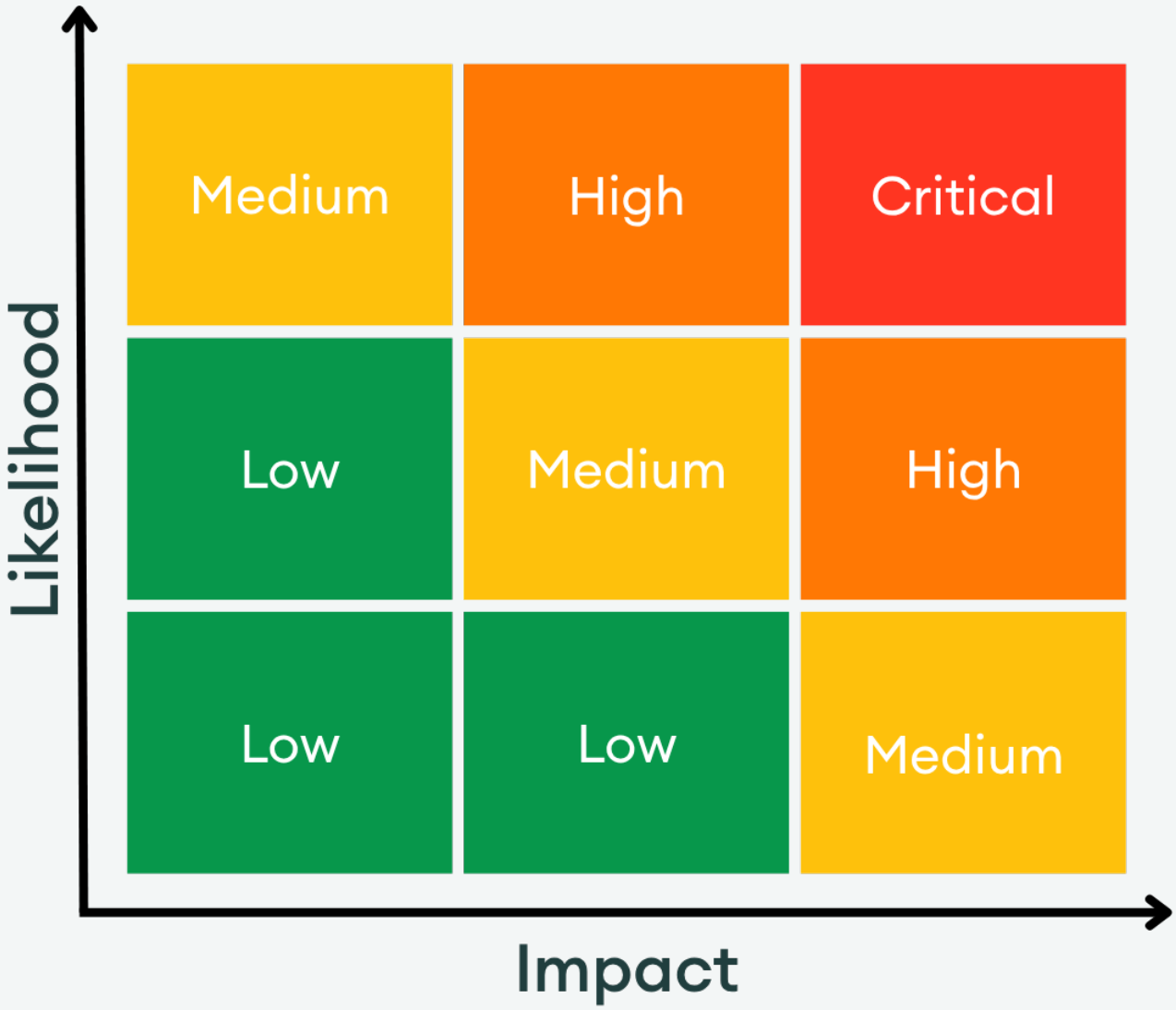
Severity Classification Matrix

By combining **Impact** and **Likelihood**, we assign a severity level using the matrix below:

- **Critical**: High impact + high likelihood (e.g. a bug that could allow anyone to drain a substansial amount of protocol funds with minimal effort)
- **High**: High impact with medium likelihood, or medium impact with high likelihood
- **Medium**: Moderate impact and/or discoverability
- **Low**: Minimal impact or unlikely to be exploited

This structured approach helps teams prioritize fixes and mitigate the most dangerous threats first.

© 2026 Adevar Labs. All rights reserved.



Severity Matrix

© 2026 Adevar Labs. All rights reserved.

4. Enhancement Opportunities

E01: Dropping unnecessary mut on base follows minimal-privilege account access

Location: https://github.com/Tempest-Finance/fogo-locker/blob/6822157c99253244f330a1fbb27541e31dfc9013/programs/locker/src/instructions/escrow_instructions/create_vesting_escrow_with_session.rs#L15-L16

>_ create_vesting_escrow_with_session.rs RUST

```rust
#[derive(Accounts)]
pub struct CreateVestingEscrowWithSessionCtx<'info> {
    #[account(mut)]
    pub base: Signer<'info>,

    #[account(
```

Description:

In CreateVestingEscrowWithSessionCtx, the base account is declared as mut.
However, base is only used as:
1. A signer (to prevent PDA seed front-running).
2. A PDA seed for the escrow account (seeds = [b"escrow".as_ref(), base.key().as_ref()]).
It is never debited, credited, or reallocated.

Potential Benefit:

Follows the principle of least privilege: accounts should only be granted the access level they actually need.

Recommendation:

Change the account declaration to remove mut.
The same cleanup could be applied to the original CreateVestingEscrowCtx and CreateVestingEscrow2Ctx.

Developer Response:

CreateVestingEscrowWithSessionCtx base account declaration has been modified to remove mut.

© 2026 Adevar Labs. All rights reserved.

E02: Session-based operations emit identical events as direct-signer operations

Location: https://github.com/Tempest-Finance/fogo-locker/blob/6822157c99253244f330a1fbb27541e31dfc9013/programs/locker/src/instructions/escrow_instructions/create_vesting_escrow_with_session.rs#L141-L152

>_ create_vesting_escrow_with_session.rs RUST

```rust
        cancel_mode,
    } = params;
    emit_cpi!(EventCreateVestingEscrow {
        vesting_start_time,
        cliff_time,
        frequency,
        cliff_unlock_amount,
        amount_per_period,
        number_of_period,
        recipient: ctx.accounts.recipient.key(),
        escrow: ctx.accounts.escrow.key(),
        update_recipient_mode,
        cancel_mode,
    });
    Ok(())
}
```

Description:

The two new session-based instructions (claim_with_session and create_vesting_escrow_with_session) emit the exact same EventClaim and EventCreateVestingEscrow events as their direct-signer counterparts (claim2, create_vesting_escrow2).

Potential Benefit:

Enriching events with signer context would allow off-chain indexers to separately track session-delegated and direct operations for analytics dashboards.

Recommendation:

Extend the event structs, or emit a supplementary event to include the actual signer and whether a session was used.

Developer Response:

Event modification is not necessary.

© 2026 Adevar Labs. All rights reserved.

E03: Removing double load will improve CU usage

Location: https://github.com/Tempest-Finance/fogo-locker/blob/6822157c99253244f330a1fbb27541e31dfc9013/programs/locker/src/instructions/escrow_instructions/claim_with_session.rs#L59

>_ claim_with_session.rs RUST

```
57:
58:     {
59:         let escrow = ctx.accounts.escrow.load()?;
60:         require!(
61:             escrow.recipient == user_pubkey,
```

Description:

The escrow account data is obtained first with load as an immutable reference for verifying the escrow recipient against the user_pubkey, then again with load_mut to handle the claim.

Potential Benefit:

Removing the redundant load can improve CU usage.

Recommendation:

This duplicate load is unnecessary and the escrow account can be loaded once mutably. The check and claim can both be performed on this mutable reference.

Developer Response:

Removed the immutable load as suggested.

© 2026 Adevar Labs. All rights reserved.

About Us

About Adevar Labs

Adevar Labs is a boutique blockchain security firm specializing in web3 audits.

Built by a mix of experienced professionals in traditional enterprise and crypto natives who have contributed to some of the most critical projects in blockchain infrastructure.

Our team's background spans companies like Bitdefender, Asymmetric Research, Quantstamp, Chainproof, and Juicebox, and includes experience securing smart contracts, bridges, and L1 and L2 protocols across ecosystems like Solana, Ethereum, Polkadot, Cosmos, and MultiversX.

With over 100 audits completed and a portfolio that includes custom fuzzers, exploit modeling, and runtime testing frameworks, Adevar Labs brings both depth and precision to every engagement.

Our auditors have discovered critical vulnerabilities, built high-impact tooling, and placed in top positions in premier audit competitions including Code4rena and Sherlock.

Team members hold distinctions such as PhDs in software protection, and key roles at Fortune 500 companies, and leadership of flagship conferences like ETH Bucharest.

With team members having publications with over 1,100 academic citations and trusted by projects securing over $500M in on-chain value, Adevar Labs blends elite technical rigor with real-world security impact.

We also collaborate with some of the best independent security researchers in the web3 space.

Projects may optionally request specific contributors to be part of their audit.

Audit Methodology

Our audit methodology is specialized to provide thorough security assessments of Solana programs. We focus explicitly on rigorous manual analysis, detailed threat modeling, and careful validation of implemented fixes to ensure your programs operate securely and as intended.

1. Program Context and Architecture Analysis

Our auditors begin by deeply examining your Solana program's documentation, intended functionality, and account design. We meticulously map out program interactions, instruction processing flows, and state management logic. Special attention is given to understanding how your program interfaces with critical Solana system programs, such as the SPL Token Program, Stake Program, and System Program. Last but not least we check external integrations with other Solana projects and verify if the inputs and return values are handled properly.

2. Threat Modeling

We conduct targeted threat modeling tailored specifically for Solana's execution environment. We carefully define attacker capabilities and identify potential vulnerabilities that may arise from Solana-specific issues, including but not limited to:

- Unauthorized account data manipulation
- Improper ownership or signer verification
- Misuse of Program-Derived Addresses (PDAs)

© 2026 Adevar Labs. All rights reserved.

- Incorrect use of Cross-Program Invocations (CPI)
- Failure to adequately handle account privileges or account states
- Risks stemming from rent-exemption and account initialization logic

3. In-depth Manual Security Review

Our experienced auditors perform an extensive manual security review of your Rust-based Solana programs. This involves a comprehensive line-by-line inspection of source code, focusing on common Solana vulnerabilities including, but not limited to:

- Missing or insufficient ownership checks
- Inadequate signer checks
- Incorrect handling of CPI calls and invocation privileges
- Arithmetic and integer overflow or underflow errors
- Unsafe deserialization and serialization of account data structures
- Improper token transfers and SPL-token logic issues
- Logic flaws in financial operations or state transitions
- Edge-case handling in instruction input validation
- Potential denial-of-service vectors related to transaction execution and account handling

During this stage, we clearly document any discovered vulnerabilities, including detailed descriptions, precise severity ratings, and recommendations for secure implementation. In situations where it is not clear how the vulnerability might be exploited we may also include a detailed proof-of-concept exploit including code snippets and instructions on how the exploit could be performed.

4. Detailed Fix Review and Validation

After the initial audit and your team's subsequent remediation efforts, we perform a comprehensive fix review to ensure the vulnerabilities identified have been effectively resolved.

We verify each fix individually, confirming that:

- Corrections effectively eliminate the security risks
- Changes do not inadvertently introduce new vulnerabilities or regressions
- The fixes align closely with Solana best practices and secure coding guidelines

Our detailed validation ensures that security improvements are robust, complete, and aligned with best practices specific to the Solana development ecosystem.

Confidentiality Notice

This report, including its content, data, and underlying methodologies, is subject to the confidentiality and feedback provisions in your agreement with Adevar Labs. These materials are not to be disclosed, extracted, copied, or distributed except to the extent expressly authorized by Adevar Labs.

Legal Disclaimer

The review and this report are provided by Adevar Labs on an as-is, where-is, and as-available basis. To the fullest extent permitted by law, Adevar Labs disclaims all warranties, expressed or implied, in

© 2026 Adevar Labs. All rights reserved.

connection with this report, its content, and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

You agree that access to and/or use of the report and other results of the review, including any associated services, products, protocols, platforms, content, and materials, will be at your sole risk.

FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE. This report is based on the scope of materials and documentation provided for a limited review at the time provided.

You acknowledge that blockchain technology remains under development and is subject to unknown risks and flaws. Adevar Labs does not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party, or any open-source or third-party software, code, libraries, materials, or information accessible through the report. As with the purchase or use of a product or service in any environment, you should use your best judgment and exercise caution where appropriate.

© 2026 Adevar Labs. All rights reserved.