Zellic

Prepared for
Fernando Otero
Anza Technology, Inc.

Prepared by
Nathanial Lattimer
Maik Robert
Zellic

December 16, 2025

Token2022 and Solana Program Libraries

Solana Program Patch Review

Contents

About Zellic

Zellic is a vulnerability research firm with deep expertise in blockchain security. We specialize in EVM, Move (Aptos and Sui), and Solana as well as Cairo, NEAR, and Cosmos. We review L1s and L2s, cross-chain protocols, wallets and applied cryptography, zero-knowledge circuits, web applications, and more.

Prior to Zellic, we founded the #1 CTF (competitive hacking) team ↗ worldwide in 2020, 2021, and 2023. Our engineers bring a rich set of skills and backgrounds, including cryptography, web security, mobile security, low-level exploitation, and finance. Our background in traditional information security and competitive hacking has enabled us to consistently discover hidden vulnerabilities and develop novel security research, earning us the reputation as the go-to security firm for teams whose rate of innovation outpaces the existing security landscape.

For more on Zellic's ongoing security research initiatives, check out our website zellic.io ↗ and follow @zellic_io ↗ on Twitter. If you are interested in partnering with Zellic, contact us at hello@zellic.io ↗.

1. Introduction

We were asked to review multiple pull requests to Token2022 and Solana Program Libraries from December 8th to December 10th, 2025, which contained various bug fixes and logic changes to programs as well as general maintenance and cleanup of dependencies used.

1.1. Scope

The engagement involved a review of the following targets:

Token2022 and Solana Program Libraries

Type	Rust
Platform	Solana

Target	token-2022
Repository	https://github.com/solana-program/token-2022

Version	`Changes in PR #819 up to commit`
	`ef44ec7ec8560c97f1da7bf86489d010001412ee`

Programs	`program/src/extension/confidential_mint_burn/processor.rs` `program/src/extension/confidential_transfer/processor.rs`

Target	token-2022
Repository	https://github.com/solana-program/token-2022

Version	`Changes in PR #763 up to commit`
	`a11efb22fdff6a9d2ce5e68c447652db72e6df94`

Programs	`program/src/extension/confidential_mint_burn/processor.rs` `program/src/extension/confidential_transfer/processor.rs`

Target	token-2022
Repository	https://github.com/solana-program/token-2022 ↗
Version	`Changes in PR #749 up to commit 826604d5b34a7e32262a384894c0f6ab761896a7`
Programs	`program/src/extension/confidential_transfer_fee/processor.rs`

Target	token-2022
Repository	https://github.com/solana-program/token-2022 ↗
Version	`Changes in PR #546 up to commit 9b6be2d77ef3601ce7fa4a7b9958c2578328b172`
Programs	`program/src/extension/mod.rs`

Target	token-2022
Repository	https://github.com/solana-program/token-2022 ↗
Version	`Changes in PR #522 up to commit ab3ceb751735e8c9a6a226c5bd61353bb8d58eba`
Programs	`clients/rust-legacy/tests/scaled_ui_amount.rs` `program/src/extension/scaled_ui_amount/processor.rs`

Target	token-2022
Repository	https://github.com/solana-program/token-2022 ↗
Version	`Changes in PR #479 up to commit 40f79abca0fdcecb3c165b3df6711a4bfedbbfb6`

Programs	`program/src/extension/confidential_transfer/processor.rs`
	`program/src/extension/mod.rs`
	`program/src/extension/reallocate.rs`
	`program/src/processor.rs`

Target	token-2022
Repository	https://github.com/solana-program/token-2022 ↗
Version	`Changes in PR #424 up to commit 4ea8b14ac9567afd2ed55c9aa754839f46539eb7`
Programs	`program/src/processor.rs`

Target	libraries
Repository	https://github.com/solana-program/libraries ↗
Version	`Changes in PR #164 up to commit b18bc14e9089b13ad573224781f52ac0ea1d3335`
Programs	`tlv-account-resolution/src/state.rs`

Target	token-2022
Repository	https://github.com/solana-program/token-2022 ↗

Version	`Changes in PR #799 up to commit`
	`b370418a7235cf8170fffef903278672f698bd4b`

Programs	`.github/workflows/main.yaml`
	`.github/workflows/publish-rust.yaml`
	`confidential/elgamal-registry-interface/src/instruction.rs`
	`confidential/elgamal-registry-interface/src/lib.rs`
	`confidential/elgamal-registry-interface/src/state.rs`
	`confidential/elgamal-registry/src/entrypoint.rs`
	`confidential/elgamal-registry/src/instruction.rs`
	`confidential/elgamal-registry/src/lib.rs`
	`confidential/elgamal-registry/src/processor.rs`
	`confidential/elgamal-registry/src/state.rs`
	`program/src/extension/confidential_transfer/processor.rs`
	`program/src/lib.rs`

Target	token-2022
Repository	https://github.com/solana-program/token-2022 ↗

Version	`Changes in PR #797 up to commit`
	`bdc8e9ffe33e7e0fbad0e2153cf7d3c82a505919`

Programs	`clients/**.rs`
	`interface/src/error.rs`
	`program/src/entrypoint.rs`
	`program/src/extension/memo_transfer/mod.rs`
	`program/src/lib.rs`
	`program/src/offchain.rs`
	`program/src/onchain.rs`
	`program/src/processor.rs`

Target	token-2022
Repository	https://github.com/solana-program/token-2022 ↗

Version	`Changes in PR #775 up to commit`
	`306a65a580332fcc815b5808217599d38629f1b9`

Programs	`clients/**.rs` `program/**.rs`

Target	token-2022
Repository	https://github.com/solana-program/token-2022 ↗

Version	`Changes in PR #616 up to commit`
	`678c36b797597e317746c721847628f11c41ce93`

Programs	`.github/workflows/main.yml`
	`interface/**.rs`
	`program/**.rs`

Target	token-2022
Repository	https://github.com/solana-program/token-2022 ↗

Version	`Changes in PR #518 up to commit`
	`5bdd18ff5c1cc37bfb3928bb6dc502153a6d7220`

Programs	`.github/workflows/main.yml`
	`program/src/extension/confidential_mint_burn/processor.rs`
	`program/src/extension/confidential_transfer/processor.rs`

Contact Information

The following project manager was associated with the engagement:

Jacob Goreski
Engagement Manager
jacob@zellic.io ↗

The following consultants were engaged to conduct the assessment:

Nathanial Lattimer
Engineer
d0nut@zellic.io ↗

Maik Robert
Engineer
maik@zellic.io ↗

1.2. Disclaimer

This assessment does not provide any warranties about finding all possible issues within its scope; in other words, the evaluation results do not guarantee the absence of any subsequent issues. Zellic, of course, also cannot make guarantees about any code added to the project after the version reviewed during our assessment. Furthermore, because a single assessment can never be considered comprehensive, we always recommend multiple independent assessments paired with a bug bounty program.

For each finding, Zellic provides a recommended solution. All code samples in these recommendations are intended to convey how an issue may be resolved (i.e., the idea), but they may not be tested or functional code. These recommendations are not exhaustive, and we encourage our partners to consider them as a starting point for further discussion. We are happy to provide additional guidance and advice as needed.

Finally, the contents of this assessment report are for informational purposes only; do not construe any information in this report as legal, tax, investment, or financial advice. Nothing contained in this report constitutes a solicitation or endorsement of a project by Zellic.

2. Patch Review

This section documents changes applied to the in-scope programs.

Token2022 PR #819. This PR added CPI guard logic to confidential extensions. No issues have been found.

Token2022 PR #763. Confidential transfers do not work with native tokens. Previously, an `assert!()` was used to check if the `token_account` is a native account. This patch added a proper check and an error message that native tokens are not supported. No issues have been found.

Token2022 PR #749. This patch reorders the logic of confidential transfers by moving the expensive ciphertext operations after basic account checks that have to pass for the transfer to succeed. No issues have been found.

Token2022 PR #546. This PR made changes to an extension-type check. No issues have been found.

Token2022 PR #522. This patch fixes logic in the scaled UI-amount extension by correctly setting the current multiplier to the new multiplier in the update instruction given enough time has passed. No issues have been found.

Token2022 PR #479. The runtime will no longer allow reallocations without zero-initialization once another PR is merged to the runtime. This fixes the usage in Token2022.

Token2022 PR #424. Previously, there was no mechanism to allow the withdrawal of excess lamports from mint accounts with no authority. This adds the ability to withdraw excess lamports if the mint keypair is a signer of the transaction and no authority is set. No issues have been found.

Libraries PR #164. There were too many potential ways that signers could be abused in transfer hooks. This patch demotes all extra accounts to nonsigners to prevent abuse. No issues have been found.

Token2022 PR #799. This is a refactor of the ElGamal registry program. Code has been moved out to an interface crate, as has been done with other crates of the Solana ecosystem. No issues have been found.

Token2022 PR #797. This PR was a version bump of the used SDK from V2 to V3. No issues have been found.

Token2022 PR #775. This patch marks reexports as deprecated to ensure the interface crates are used in the future. No issues have been found.

Token2022 PR #616. This refactored code by moving logic into interfaces to improve Solana development experience. No issues have been found.

Token2022 PR #518. This patch gates code behind the `zk-ops` feature. No issues have been found.