Cryptonics

Continuous Audit Report 2

pTokens Bridges

OCTOBER 15, 2020

Table of Contents

Disclaimer

THE CONTENT OF THIS AUDIT REPORT IS PROVIDED “AS IS”, WITHOUT REPRESENTATIONS AND WARRANTIES OF ANY KIND.

THE AUTHOR AND HIS EMPLOYER DISCLAIM ANY LIABILITY FOR DAMAGE ARISING OUT OF, OR IN CONNECTION WITH, THIS AUDIT REPORT.

This audit has been performed by

Philip Stanislaus

Cryptonics Consulting S.L.
Ramiro de Maeztu 7
46022 Valencia
SPAIN

https://cryptonics.consulting/
info@cryptonics.consulting

Summary of Findings

No	Description	Severity	Status
1	EOS block submission does not enforce submission of blocks with schedule/producer set changes	Major	Acknowledged
2	EOS block validation in core-private does not check whether block has been confirmed by enough producers	Medium	Acknowledged
3	ERC20 on EOS debug nonces are incremented before checking potential errors	Minor	Resolved
4	ERC20 on EOS debug nonces incrementing and reading happens not atomic	Minor	Resolved
5	Duplicate usage of contract address DB key	Minor	Resolved
6	Potential panics	Minor	Resolved
7	Outdated dependencies	Minor	Acknowledged
8	Code duplication	Minor	Acknowledged
9	Architecture/separation of concerns	Minor	Acknowledged

This report contains 9 findings on 16 pages (including one cover page).

All issues encountered during the audit process have been addressed by the team.

Introduction

Purpose of this Report

Cryptonics Consulting has been engaged to perform a continuous audit of the pTokens 2-way asset transfer bridges, forming part of the pTokens project (https://ptokens.io/). The engagement is ongoing and includes the changes performed to the codebase since the last audit report from September 16, 2020.

The objectives of the audit are as follows:

1. Determine the correct functioning of the contract, in accordance with the project specification.
2. Determine possible vulnerabilities, which could be exploited by an attacker.
3. Determine contract bugs, which might lead to unexpected behavior.
4. Analyze whether best practices have been applied during development.
5. Make recommendations to improve code safety and readability.

This report represents a summary of the findings.

As with any code audit, there is a limit to which vulnerabilities can be found, and unexpected execution paths may still be possible. The author of this report does not guarantee complete coverage (see disclaimer).

Codebase Submitted for the Audit

The audit has been performed on the changes between the following commits in the code provided by the developers in the following GitHub repository:

https://github.com/provable-things/ptokens-core-private

Start commit: `b5a47d70afbbb9f5084efeb1c9807749e1395ad7` (tag `v1.11.0`)
Final commit: `462167c86a37f6efcccd2a261a93af52d6370b83`

This report presents the findings over 232 commits with 238 changed files, containing 9,107 additions and 6,915 deletions.

Methodology

The audit has been performed in the following steps:

1. Gaining an understanding of the code base's intended purpose by reading the available documentation.
2. Automated source code and dependency analysis.
3. Manual line by line analysis of the source code for security vulnerabilities and use of best practice guidelines, including but not limited to:
    a. Race condition analysis
    b. Under- / overflow issues
    c. Key management vulnerabilities
4. Report preparation

Project Overview

The submitted code implements a cross-blockchain bridge that allows assets to be moved between Ethereum and EOS as well as between Bitcoin and EOS. Bitcoin is represented on Ethereum and EOS as pBTC.

The two-way peg works by depositing (locking) BTC on Bitcoin and minting pBTC on Ethereum or EOS and by burning pBTC on Ethereum or EOS and unlocking BTC on Bitcoin. Transactions are relayed across chains through light clients designed to operate in a secure enclave.

Since the last audit report, support for moving ERC20 assets between Ethereum and EOS has been added.

The enclave is designed to be executed in a protected enclave using a trusted execution environment, such as Intel SGX.

Findings

1. EOS block submission does not enforce submission of blocks with schedule/producer set changes

Severity: Major

`submit_eos_block_to_core` in `core-private/src/btc_on_eos/eos/submit_eos_block.rs` does not enforce that it receives blocks with schedule changes. This allows blocks to be accepted from a block producer that has been removed from the schedule, opening the possibility of double-spending.

Recommendation

Fail if a block is submitted for a schedule that has not been confirmed by a block with schedule changes.

Status: Acknowledged

Due to the lack of library support caused by moving schedule changes into `header_extenison` in EOS 2.0, the current version does not support parsing schedule changes. The issue described above is currently worked around by supplying schedule changes in a semi-trusted manner.

2. EOS block validation in `core-private` does not check whether a block has been confirmed by enough producers

Severity: Medium

In the current implementation, `core-private` processes the supplied EOS block without any checks for its finality. This means that the core could be currently looking at a fork that might not end up in the canonical chain, leading to an inconsistent database or even to double-spending. While this issue is mitigated by the fact that the current implementation of `eos-syncer` filters actions by for irreversibility in `eos-syncer/lib/get-redeem-actions.js:36`, it adds a dependency on that component working as expected, requiring an additional trusted component in the system.

Recommendation

Only process irreversible blocks in `core-private`. Until real-time BFT has been added to EOS, the best approach is to track the finalized block (called last irreversible block or LIB in EOS) by ensuring that ⅔ + 1 block producers (15 out of the 21) have confirmed the current chain.

Status: Acknowledged

The current implementation of relying on `eos-syncer` to supply only irreversible blocks is intended for the alpha version of the bridge only. It is intended to submit blocks with relevant actions along with the subsequent blocks required for that relevant block to achieve irreversibility in order that the core knows that the block in question has reached finality.

3. ERC20 on EOS debug nonces are incremented before checking potential errors

Severity: Minor

Nonces in the ERC20 on EOS debug functions `debug_get_perc20_migration_tx`, `debug_get_add_supported_token_tx` and `debug_get_remove_supported_token_tx` are incremented before all pre-conditions for the signatures are checked. If any of the conditions are not satisfied, the nonce would have been incremented, without a signed transaction. That would result in any further transaction having a wrong nonce. Examples for failing conditions:

1. `debug_get_perc20_migration_tx`: a missing gas price in the database
2. `debug_get_add_supported_token_tx`: a missing `eos_erc20_smart_contract_address` in the database
3. `debug_get_remove_supported_token_tx`: a missing `eos_erc20_smart_contract_address` in the database

Recommendation

Only increment the nonce once all conditions have been checked.

Status: Resolved

Fixed in `f0f04b98c3b8d0724c68ba2fcf3e35a0921c43fb` and `e00a5c543aabf7e06fcd411734a7a63dc7dec9ec`.

4. ERC20 on EOS debug nonces incrementing and reading happens not atomic

Severity: Minor

Nonces in the ERC20 on EOS debug functions `debug_get_perc20_migration_tx`, `debug_get_add_supported_token_tx` and `debug_get_remove_supported_token_tx` are incremented in a database transaction, but read after the transaction is ended. Consequently, another process could also increment the nonce before the read, resulting in a skipped nonce and two transactions with the same nonce.

Recommendation

Include both writing and reading of the nonce in the transaction.

Status: Resolved

Fixed in `f0f04b98c3b8d0724c68ba2fcf3e35a0921c43fb` and `e00a5c543aabf7e06fcd411734a7a63dc7dec9ec`.

5. Duplicate usage of contract address DB key

Severity: Minor

Both the ERC777 smart contract address and the ERC20 smart contract address are stored under the same key `ETH_SMART_CONTRACT_ADDRESS_KEY` in the database (`src/chains/eth/eth_database_utils.rs:509` and `524`). This will cause issues in case both feature with different smart contracts in the same core at some point in the future. It should not cause issues with multiple cores running on the same database if distinct database prefixes are used.

Recommendation

Use distinct keys for ERC777 and ERC20 smart contract addresses.

Status: Resolved

Fixed in `9858dc31fc10fd4601cc374f4762086de496cd24`, `6e5f5248880b3032d6800cf313656b3cf144c345`, `065e1e6be3600529f7cfe4ae9edcac7844c2e2b1`, `fa247523491dde83b4c8b87cbc8b3dea56c84e2a` and `6ead7e973d61fd7d3aaa56bc0d91a1a38f88d76f`.

6. Potential panics

Severity: Minor

There are still a few places in the codebase where panics can occur. It is generally preferred to explicitly handle errors by returning a `Result`, which is done in most of the code. That improves debugging and allows graceful shutdowns.

- In `src/chains/eth/eth_metadata.rs:86`, unwrap is used.
- In `src/base58.rs:140`, assert_eq is used.

Recommendation

Return a `Result` instead and handle the error case.

Status: Resolved

Fixed in `7237d4055d3b6a9e0eec7d35de523b9f67eea72d`.

7. Outdated dependencies

Severity: Minor

Many dependencies are outdated, which could introduce security risks. A full list of the outdated dependencies has been added to Appendix A.

Recommendation

Upgrade dependencies.

Status: Acknowledged

Will be upgraded in the future.

8. Code duplication

Severity: Minor

There is duplicated code in multiple places across the codebase, most notably between `core-private/src/btc_on_eos` and `core-private/src/btc_on_eth`. This duplication has been documented in `core-private/src/notes`. While this is not a vulnerability per se, it makes the codebase hard to maintain and increases the likelihood of bugs being introduced by changes that are not applied consistently across the duplicates. While some shared functionality has been factored out since the last audit report, there is still a lot of common behavior duplicated across those directories, which makes maintenance hard and can introduce regressions of bugs over time. Even worse, it could lead to separate implementations of the same functionality, which increases maintenance burden even more.

Recommendation

Extract shared behavior into shared crates/packages. Use a tool like duplo to find code duplication. De-duplication of the codebase does not just improve maintainability, it also speeds up the addition of bridges to further blockchains.

Status: Acknowledged

Refactoring of code has been continued since the last audit, but not finished before this audit report was released.

9. Architecture/separation of concerns

Severity: Minor

The separation of logic into bridge and chain directories simplified the architecture a lot. Unfortunately, there are still some places where the separation is broken. Fox example, `src/chains/eth/eth_state.rs` is in the chains directory, but contains logic specific to the EOS bridges, for instance `eos_transactions`. This mix of logic makes the codebase harder to reason about and makes long term maintenance harder. We recommend extracting the bridge specific logic for a cleaner separation of concerns.

Additionally, there are some types that implement behaviour for different manifestations of those types, requiring runtime checks which manifestation is currently supposed to be used. An example for that is the `EthLog` type, which contains different types of logs. For instance, each call to `get_btc_on_eth_redeem_amount` needs to do a runtime check (`check_is_btc_on_eth_redeem`) whether the correct type is present. This could cause bugs, if a check is missed. We recommend using specific types instead in order to let the Rust compiler do the checks. That will improve the robustness of the code, while improving separation of concerns at the same time.

Recommendation

Use generic types that are used across bridges and add bridge-specific types for the individual bridges. For sharing logic, traits and generic `impl` blocks can be used.

Status: Acknowledged

Will be considered during the next refactoring.

Appendix A: Outdated dependencies

```
> cargo outdated
Name                                    Project                     Compat   Latest   Kind         Platform
----                                    -------                     ------   ------   ----         --------
atty->hermit-abi                        0.1.16                      0.1.17   0.1.17   Normal       cfg(target_os = "hermit")
atty->libc                              0.2.77                      0.2.79   0.2.79   Normal       cfg(unix)
base64->byteorder                       1.3.2                       ---      1.3.4    Normal       ---
bitcoin->bitcoin_hashes                 0.7.0                       ---      0.7.6    Normal       ---
bitcoin->byteorder                      1.3.2                       ---      1.3.4    Normal       ---
bitcoin_hashes                          0.7.0                       ---      0.9.0    Normal       ---
bitcoin_hashes->byteorder               1.3.2                       ---      Removed  Normal       ---
bitcoin_hashes->serde                   1.0.102                     ---      Removed  Normal       ---
blake2b_simd->arrayref                  0.3.6                       ---      Removed  Normal       ---
blake2b_simd->arrayvec                  0.5.1                       ---      Removed  Normal       ---
blake2b_simd->constant_time_eq          0.1.5                       ---      Removed  Normal       ---
byteorder                               1.3.2                       ---      1.3.4    Normal       ---
chrono                                  0.4.9                       ---      0.4.19   Normal       ---
chrono->libc                            0.2.77                      0.2.79   0.2.79   Normal       ---
crossbeam-utils->autocfg                1.0.1                       ---      Removed  Build        ---
crossbeam-utils->cfg-if                 0.1.10                      ---      Removed  Normal       ---
crossbeam-utils->lazy_static            1.4.0                       ---      Removed  Normal       ---
derive_more                             0.99.10                     0.99.11  0.99.11  Normal       ---
derive_more->proc-macro2                1.0.21                      1.0.24   1.0.24   Normal       ---
derive_more->syn                        1.0.41                      1.0.44   1.0.44   Normal       ---
dirs->cfg-if                            0.1.10                      ---      Removed  Normal       ---
dirs->dirs-sys                          0.3.5                       ---      Removed  Normal       ---
dirs-sys->libc                          0.2.77                      0.2.79   Removed  Normal       cfg(unix)
dirs-sys->redox_users                   0.3.5                       ---      Removed  Normal       cfg(target_os = "redox")
dirs-sys->winapi                        0.3.9                       ---      Removed  Normal       cfg(windows)
eos-keys->bitcoin_hashes                0.7.0                       ---      0.7.6    Normal       ---
eos-keys->byteorder                     1.3.2                       ---      1.3.4    Normal       ---
eos-keys->rustc-hex                     2.0.1                       ---      2.1.0    Normal       ---
eos-primitives->bitcoin_hashes          0.7.0                       ---      0.7.6    Normal       ---
eos-primitives->chrono                  0.4.9                       ---      0.4.19   Normal       ---
eos-primitives->hex                     0.4.0                       ---      0.4.2    Normal       ---
eos-primitives->serde                   1.0.102                     ---      1.0.116  Normal       ---
ethabi->rustc-hex                       2.0.1                       ---      2.1.0    Normal       ---
ethabi->serde                           1.0.102                     ---      1.0.116  Normal       ---
ethabi->serde_json                      1.0.40                      ---      1.0.59   Normal       ---
fixed-hash->byteorder                   1.3.2                       ---      1.3.4    Normal       ---
fixed-hash->rand                        0.7.2                       ---      0.7.3    Normal       ---
fixed-hash->rustc-hex                   2.0.1                       ---      2.1.0    Normal       ---
getrandom->cfg-if                       0.1.10                      ---      Removed  Normal       ---
getrandom->libc                         0.2.77                      0.2.79   0.2.79   Normal       cfg(unix)
getrandom->libc                         0.2.77                      0.2.79   Removed  Normal       cfg(unix)
getrandom->wasi                         0.9.0+wasi-snapshot-preview1 ---     Removed  Normal       cfg(target_os = "wasi")
hermit-abi->libc                        0.2.77                      0.2.79   0.2.79   Normal       ---
hex                                     0.4.0                       ---      0.4.2    Normal       ---
impl-rlp->rlp                           0.4.2                       ---      0.4.6    Normal       ---
impl-serde->serde                       1.0.102                     ---      1.0.116  Normal       ---
log                                     0.4.8                       ---      0.4.11   Normal       ---
parity-scale-codec->serde               1.0.102                     ---      1.0.116  Normal       ---
proc-macro2->unicode-xid                0.2.1                       ---      Removed  Normal       ---
quote->proc-macro2                      1.0.21                      1.0.24   1.0.24   Normal       ---
quote->proc-macro2                      1.0.21                      1.0.24   Removed  Normal       ---
rand                                    0.7.2                       ---      0.7.3    Normal       ---
rand->libc                              0.2.77                      0.2.79   0.2.79   Normal       cfg(unix)
rand_jitter->libc                           0.2.77                          0.2.79   0.2.79   Normal       cfg(any(target_os =
"macos", target_os = "ios"))
rand_os->libc                           0.2.77                      0.2.79   0.2.79   Normal       cfg(unix)
redox_users->getrandom                  0.1.15                      ---      Removed  Normal       ---
redox_users->redox_syscall              0.1.57                      ---      Removed  Normal       ---
redox_users->rust-argon2                0.8.2                       ---      Removed  Normal       ---
rlp                                     0.4.2                       ---      0.4.6    Normal       ---
rlp->rustc-hex                          2.0.1                       ---      2.1.0    Normal       ---
rust-argon2->base64                     0.12.3                      ---      Removed  Normal       ---
rust-argon2->blake2b_simd               0.5.10                      ---      Removed  Normal       ---
rust-argon2->constant_time_eq           0.1.5                       ---      Removed  Normal       ---
rust-argon2->crossbeam-utils            0.7.2                       ---      Removed  Normal       ---
rustc-hex                               2.0.1                       ---      2.1.0    Normal       ---
secp256k1->cc                           1.0.60                      1.0.61   1.0.61   Build        ---
serde                                   1.0.102                     ---      1.0.116  Normal       ---
serde->serde_derive                     1.0.101                     ---      1.0.116  Normal       ---
serde->serde_derive                     1.0.101                     ---      Removed  Normal       ---
serde-big-array->serde                  1.0.102                     ---      1.0.116  Normal       ---
serde-big-array->serde_derive           1.0.101                     ---      1.0.116  Normal       ---
serde_derive                            1.0.101                     ---      1.0.116  Normal       ---
serde_derive->proc-macro2               1.0.21                      1.0.24   1.0.24   Normal       ---
```

serde_derive->proc-macro2	1.0.21	1.0.24	Removed	Normal	---
serde_derive->quote	1.0.7	---	Removed	Normal	---
serde_derive->syn	1.0.41	1.0.44	1.0.44	Normal	---
serde_derive->syn	1.0.41	1.0.44	Removed	Normal	---
serde_json	1.0.40	---	1.0.59	Normal	---
serde_json->serde	1.0.102	---	1.0.116	Normal	---
simple_logger	1.9.0	1.11.0	1.11.0	Development	---
simple_logger->chrono	0.4.9	---	0.4.19	Normal	---
simple_logger->log	0.4.8	---	0.4.11	Normal	---
simplelog	0.7.3	---	0.8.0	Normal	---
simplelog->chrono	0.4.9	---	0.4.19	Normal	---
simplelog->log	0.4.8	---	0.4.11	Normal	---
simplelog->term	0.6.1	---	Removed	Normal	---
syn->proc-macro2	1.0.21	1.0.24	1.0.24	Normal	---
syn->proc-macro2	1.0.21	1.0.24	Removed	Normal	---
syn->quote	1.0.7	---	Removed	Normal	---
syn->unicode-xid	0.2.1	---	Removed	Normal	---
term->dirs	2.0.2	---	Removed	Normal	---
term->winapi	0.3.9	---	Removed	Normal	cfg(windows)
time->libc	0.2.77	0.2.79	0.2.79	Normal	---
tiny-keccak	1.5.0	---	2.0.2	Normal	---
uint->byteorder	1.3.2	---	1.3.4	Normal	---
uint->rustc-hex	2.0.1	---	2.1.0	Normal	---
winapi->winapi-i686-pc-windows-gnu	0.4.0	---	Removed	Normal	i686-pc-windows-gnu
winapi->winapi-x86_64-pc-windows-gnu	0.4.0	---	Removed	Normal	x86_64-pc-windows-gnu