POPCHAIN PROJECT

CORE AUDIT

Version 1.0.0

--

Presented by

**Yong-hyu Ban**

Developer, SCV SOFT

**Benjamin Hyokeun Oh**

CEO, SCV SOFT

Aug 9th, 2019

Contents

Introduction

Disclaimer

Executive Summary

Summary of Analysis

Analysis Methods

Analysis Results

1. Static analysis using Synopsis® Coverity® Scan - High impact issues

1.1 CID 242525 - Pointer to local outside scope (use-after-free)

1.2 CID242511 – Resource leak

1.3 CID 242479 - Pointer to local outside scope (use-after-free

1.4 CID242476 – Resource leak

1.5 CID242474 – Resource leak

1.6 CID242447 - Uninitialized local variable

1.7 CID186065 - Out-of-bounds access

2 Static analysis using Synopsis® Coverity® Scan - Medium impact issues

2.1 CID242536 - Dereference after null check

2.2 CID242518 - Dereference after null check

2.3 CID242507 - Explicit null dereferenced

2.4 CID242505 - Explicit null dereferenced

2.5 CID242494 - Dereference after null check

2.6 CID242485 - Dereference null return value

2.7 CID242477 - Dereference after null check

2.8 CID16467 - Improper use of negative value

Conclusions

Introduction

This document includes the results of the audit performed by the SCV SOFT at the request of the POPCHAIN team. And those issues we found has been fixed by the POPCHAIN development ream. The audited code can be found in the public pop-core Github repository, and the version used for this report is commit.

Disclaimer

The audit does not give any warranties on the security of the code. One audit cannot be considered enough. We always recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts. Besides, security audit is not an investment advice.

Executive Summary

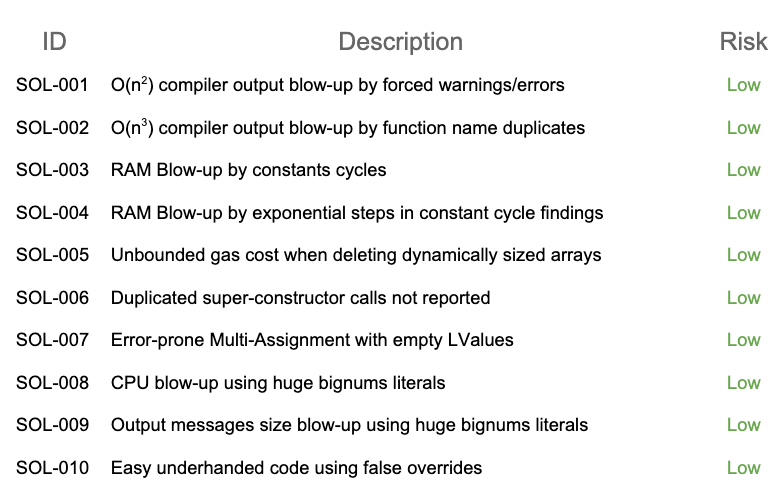
The goal of this audit is to review POPCHAIN’s solidity implementation for its decentralized prediction market, study potential security vulnerabilities, its general design and architecture, and uncover bugs that could compromise the software in production.

During ​​the​​ assessment,​​ SCV SOFT ​​identified ​​0 ​​high-risk​​issues, ​​0 ​​medium-risk​​issues, ​​and​​ 10 low-risk​​issues.​​ The​​ issues​​ identified​​ during ​​the ​​assessment ​​do​​ not ​​lead ​​to​​ the ​​compilation ​​of vulnerable ​​code. ​​Those ​​issues​​ were​​ communicated​​ to ​​the ​​POPCHAIN ​​team​​ and​​ fixed in​​ newer ​​releases.

Summary of Analysis

In this report, we performed our audit according to the procedure described below. Our audit is targeted the full POPCHAIN project except for BerkeleyDB and libboost component.

**The audit showed no critical issues.** And also, those issues have been fixed by POPCHAIN team and SCV SOFT.



참고자료

Analysis Methods

• Full project static analysis using Synopsis® Coverity® Scan

• Manual code inspection

Analysis Result

**1. Static analysis using Synopsis® Coverity® Scan - High impact is- sues**

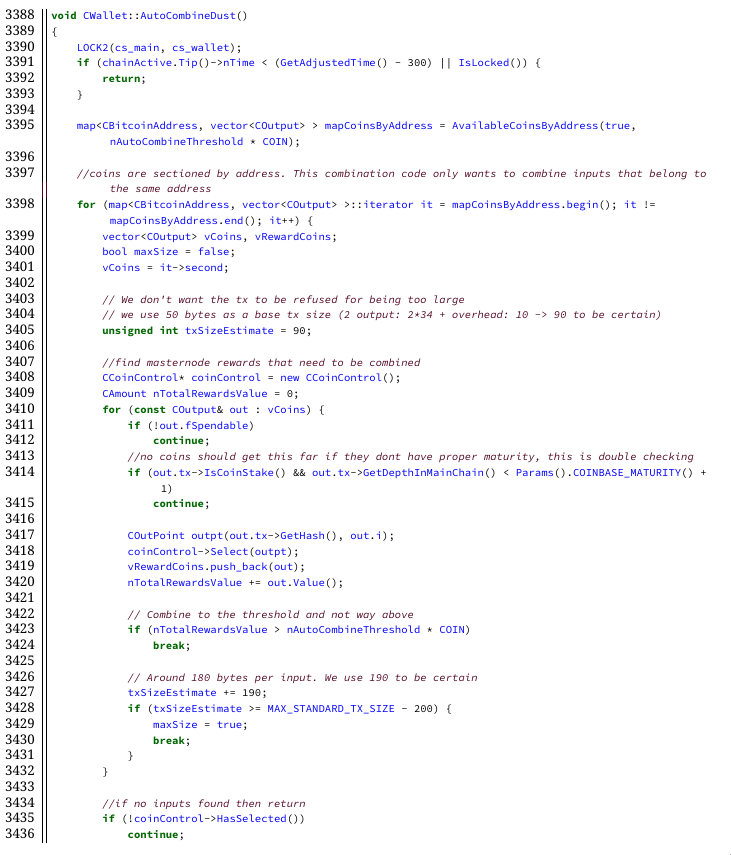
1.1 CID 242525 - Pointer to local outside scope (use-after-free)

In src/main.cpp:AddInvalidSpendsToMap, reference to ’publicSpend’ local variable is used after ’publicSpend’ is freed from the stack. It can be led to data corruption.



1.2 CID 242511 - Resource leak

In src/wallet/wallet.cpp:CWallet::AutoCombineDust, ’coinControl’ is dynamically allocated but not freed when coinControl->HasSelected() is false. It may lead to a memory leak which can disrupt stability.



4.1.3 CID 242479 - Pointer to local outside scope (use-after-free)

In src/miner.cpp:CreateNewBlock, reference to ’publicSpoend’ and ’spendObj’ member variable is used after ’publicSpend’ and ’spendObj’ is freed from the stack. It can be led to data corruption.



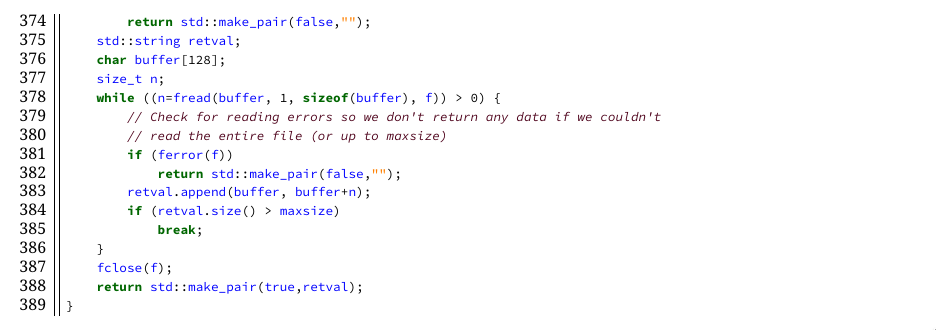
1.4 CID 242476 - Resource leak

In src/wallet/rpcwallet.cpp:serchdzpch, dzpchThreads is dynamically allocated but not freed before control leaves the function. It will cause a memory leak which can disrupt stability.



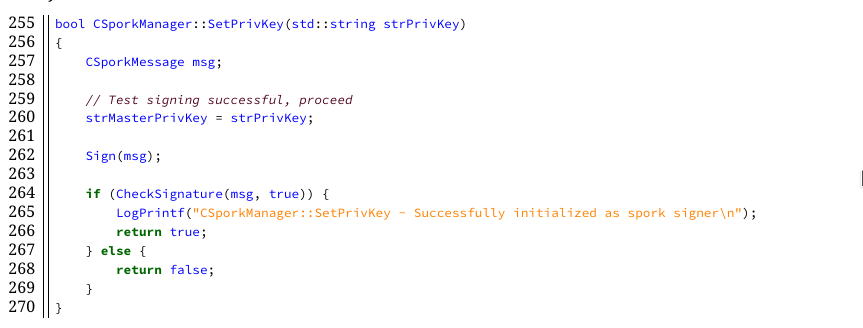
1.5 CID 242474 - Resource leak

In src/torcontrol.cpp:ReadBinaryFile, the file is not properly closed if file descriptor encountered an error. It can cause file descriptor leak which can lead to system instability.



1.6 CID 242447 - Uninitialized local variable

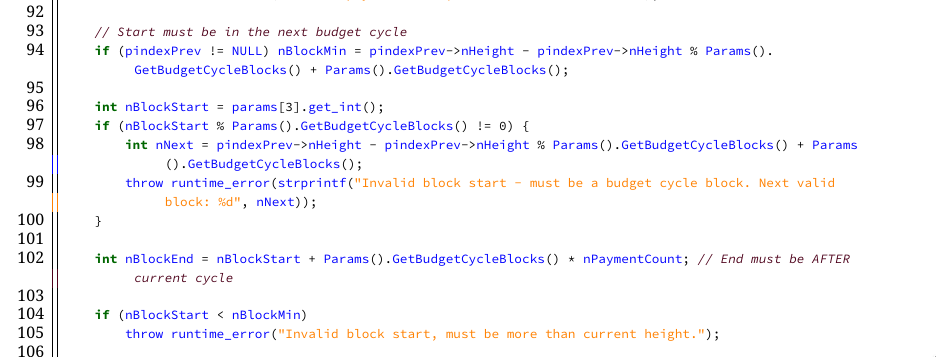
In src/spork.cpp:SetPrivKey uses an uninitialized local variable to test spork signer. It may lead to an undesirable result. It is strongly advised to use a properly initialized object.



**2 Static analysis using Synopsis® Coverity® Scan - Medium impact issues**

2.1 CID 242536 - Dereference after null check

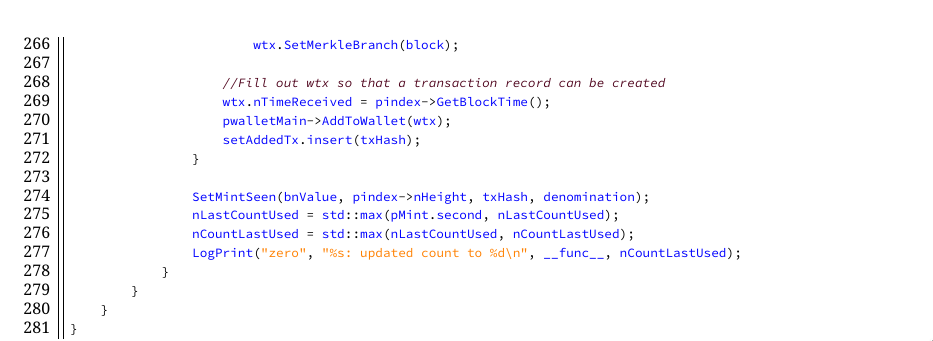
In src/rpc/budget.cpp:preparebudget, pindexPrev can be null in a special case. Though it will not cause a practical problem in production due to its special condition requirement, it is good to fix this problem to improve code quality.



2.2 CID 242518 - Dereference after null check

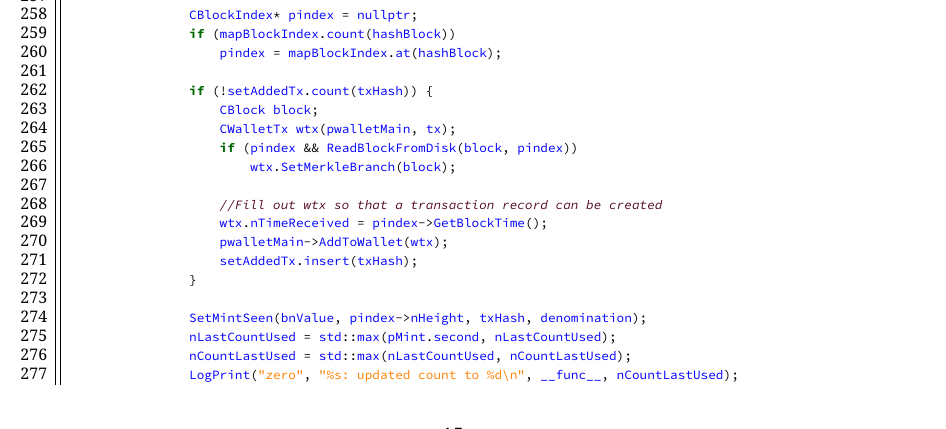
In src/zpch/zpchwallet.cpp:CzPCHWallet::SyncWithChain, in case of pindex is null (some special condition is required), it can cause wallet crash.





2.3 CID 242507 - Explicit null dereferenced

In src/zpch/zpchwallet.cpp:CzPCHWallet::SyncWithChain, in special case if mapBlockIndex.count(hashBlock) is zero, pindex will be null. In this case, it will lead to wallet crash.



2.4 CID 242505 - Explicit null dereferenced

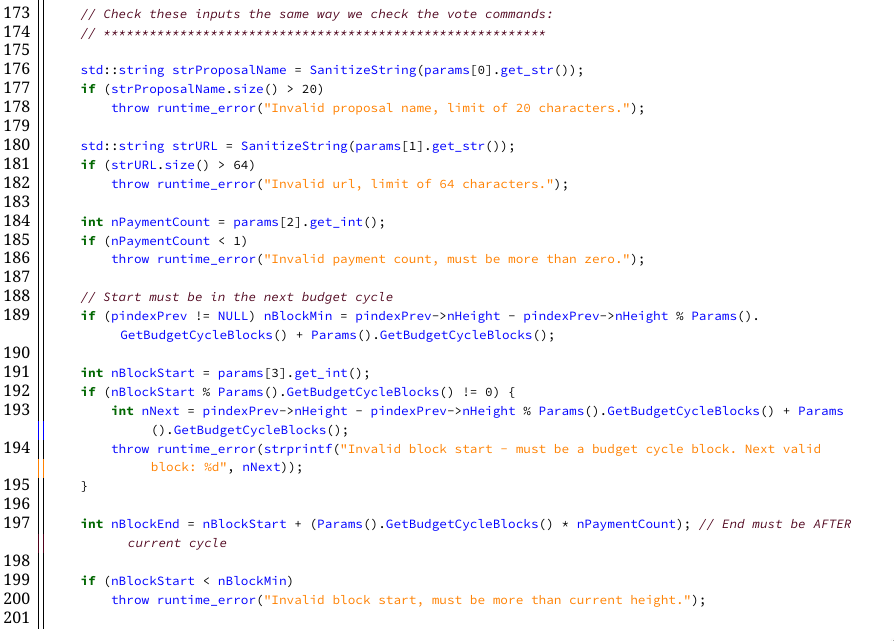
In src/main.cpp:AcceptBlock, in special case when handling genesis block and zerocoin is active, it will lead into null pointer dereference. In this case, it will lead to a wallet crash.

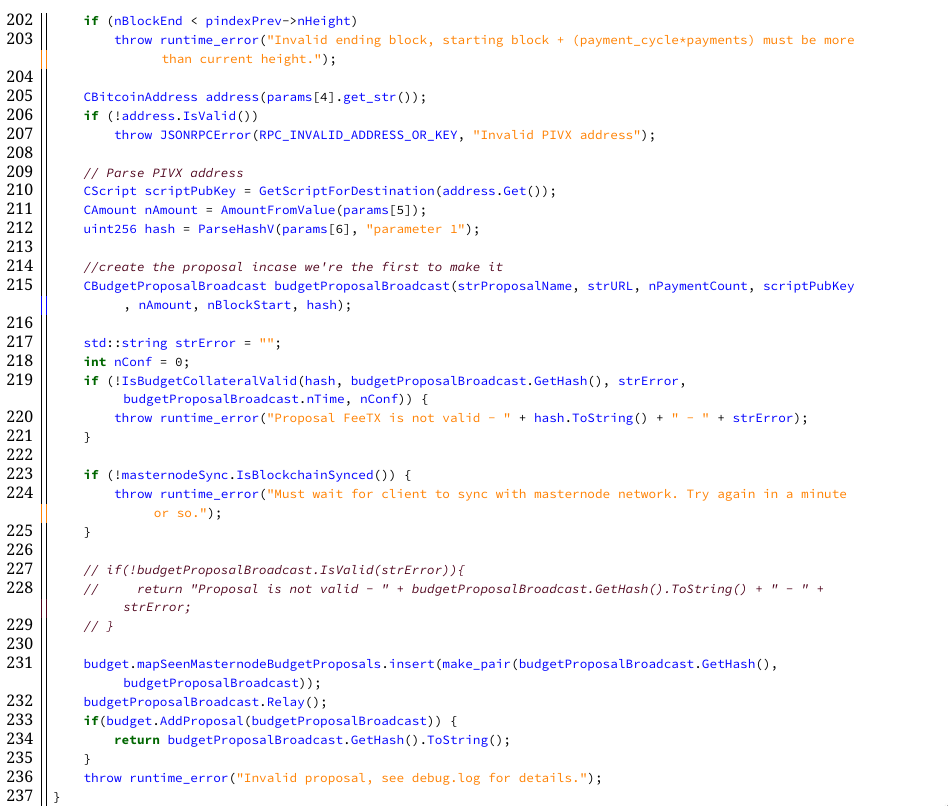




2.5 CID 242494 - Dereference after null check

In src/rpc/budget.cpp:submitbudget, in special case when handling genesis block, it can lead into null pointer dereference. In this case, it will lead to a wallet crash.





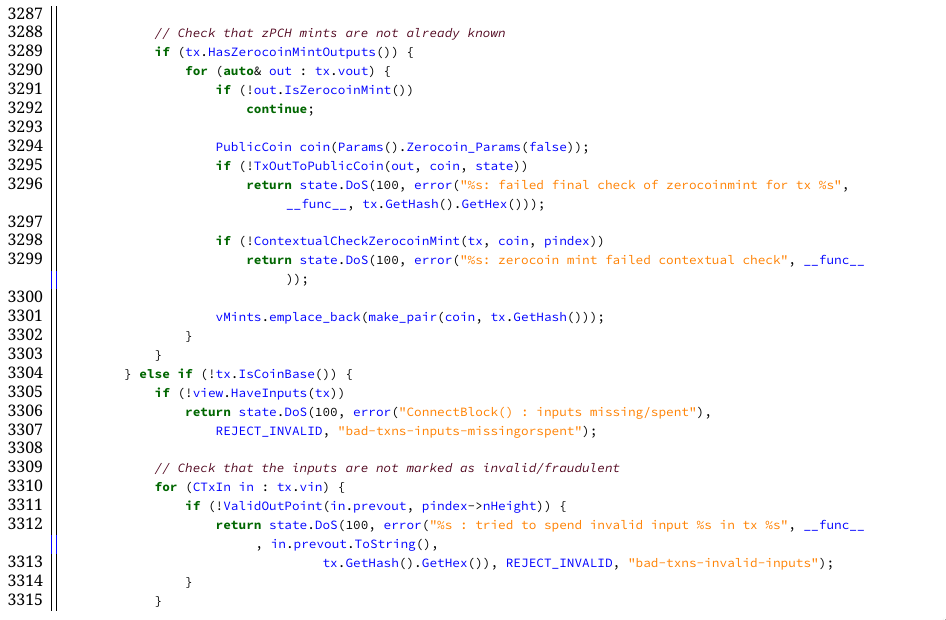
2.6 CID 242485 - Dereference null return value

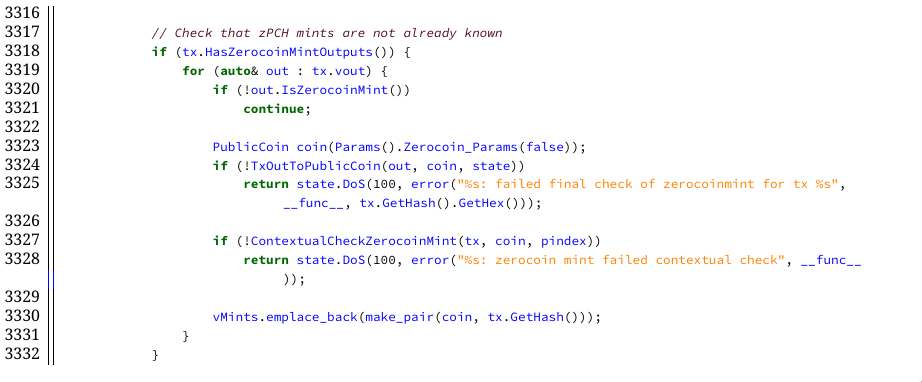
In src/rpc/rawtransaction.cpp:signrawtransaction, when wallet is in regression test mode, there is possibility to crash wallet with crafted request. This vulnerability will not affect production environment because regression test cannot be connected to internet but it is good to fix this problem to improve code quality.



2.7 CID 242477 - Dereference after null check

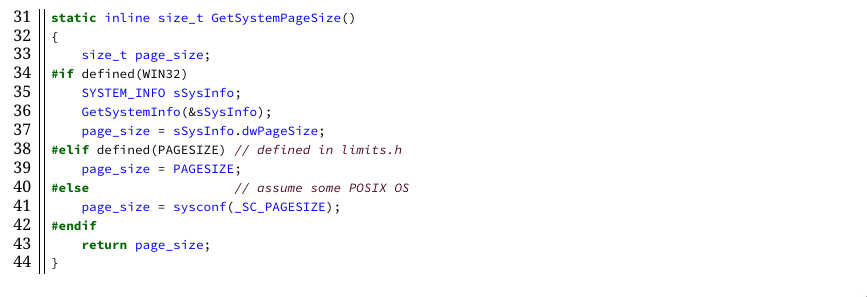
In src/main.cpp:ConnectBlock, in practically impossible case when wallet tries to con- nect genesis block to somewhere, it can lead into null pointer dereference. It is not possi- ble to trigger crash using this problem but it is good to fix this problem to improve code quality.





2.8 CID 16467 - Improper use of negative value

In src/allocators.cpp:GetSystemPageSize, if there is no PAGESIZE defined in limits.h, it falls back to sysconf(3) syscall. In case of sysconf(3) fails, it returns negative number which will cause severe misbehavior of wallet.



Conclusions

There were some issues on POPCHAIN core code, but those issues had been fixed with this report. POPCHAIN Mainnet would fully work for purpose. This is guaranteed by SCV SOFT vulnerability research team.

Singnature \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_