Security Analysis Report

solana-civ

Oct 13, 2023

by sec3 X-ray Auditor

TABLE OF CONTENTS

Summary 3

Disclaimer 4

About sec3 X-ray Auditor 5

About sec3 5

Overview of the Result 6

Program: programs_solciv 7

Appendix A - sec3 Vulnerabilities and Exposures (SVE) 8

Summary

sec3 X-ray Auditor ("sec3 Auditor") was used by Metaversium (the "Client") to conduct security analysis of the **solana-civ** Solana smart contract program. The artifact of the analysis was the source code of the following on-chain smart contract excluding tests in a public repository:

- Commit **c417b2642e6a2792429e2f1f232e0f3a45f499be**

This analysis revealed 1 potential issues, of which 0 are critical.

This report presents the output from sec3 Auditor.

Disclaimer

This report ("**Report**") includes the results of a security analysis, by **Sec3 X-ray Auditor**, of a specific build and/or version of the source code provided by the Client and specified in the Report ("**Assessed Code**").

The sole purpose of the Report is to provide the Client with the results of the security analysis of the Assessed Code. The Report does not apply to any other version and/or build of the Assessed Code.

Regardless of its contents, the Report does not (and shall not be interpreted to) provide any warranty, representation, or covenant that the Assessed Code: (i) is error and/or bug-free, (ii) has no security vulnerabilities, and/or (iii) does not infringe any third-party's rights. The Report is not, and shall not be construed or interpreted, in any manner, as, (i) an endorsement by the Company of the Assessed Code and/or of the Client, or (ii) investment advice or a recommendation to invest in the Assessed Code and/or the Client.

This Report shall be null and void if the Report (or any portion thereof) is altered in any manner.

ABOUT

About sec3 X-ray Auditor

sec3 X-ray Auditor extracts essential code structure and relationships into a set of databases that enable sophisticated analysis of source code. Sec3 Software employs Maximal Concolic Execution (MCE) techniques, amongst others, to provide the ability to systematically explore code paths, encode path conditions and check path invariants.

At the time of this report, sec3 Auditor can scan 60 types of security vulnerabilities, including Missing Signer Check, Missing Owner Check, etc. Please refer to Appendix A for more information.

About sec3

Founded by leading academics in the field of software security and senior industrial veterans, sec3 is a leading blockchain security company that currently focuses on Solana programs. We are also building sophisticated security tools that incorporate static analysis, penetration testing, and formal verification. At sec3, we identify and eliminate security vulnerabilities through the most rigorous process and aided by the most advanced analysis tools. For more information, check out our website and follow us on twitter.

Overview of the Result

By Vulnerability Type

Untrustful Account	1
Unsafe Operation	0
Cosplay Account	0

By Severity

Critical	0
High	1
Medium	0
Low	0

By Program

programs_solciv	1

Program: programs_solciv

Untrustful Account No.1 High

SVE1019 - The account may not be properly validated and may be untrustful

proxycapital-solana-civ-c417b26/programs/solciv/src/instructions/mint.rs:51

```
 45|        associated_token::mint = mint,
 46|        associated_token::authority = owner,
 47|    )]
 48|    pub destination: Account<'info, TokenAccount>,
 49|    /// CHECK: this can be any personal address of the player
 50|    /// it's important to check the signer, while recipient of gems can be any address
>51|    pub owner: AccountInfo<'info>,
 52|    #[account(mut)]
 53|    pub player_account: Account<'info, Player>,
 54|    #[account(mut)]
 55|    pub player: Signer<'info>,
 56|    pub rent: Sysvar<'info, Rent>,
 57|    pub system_program: Program<'info, System>,
```

Appendix A - sec3 Vulnerabilities and Exposures (SVE)

SVE	Checker	Description	Examples
SVE10001	ReentrancyEtherVulnerability	The function may suffer from reentrancy attacks due to the use of call.value, which can invoke an external contract's fallback function	Example
SVE10002	ArbitrarySendERC20	The function may allow an attacker to send from an arbitrary address, instead of from the msg.sender	Example
SVE10003	UnprotectedSelfDestruct	The function may allow an attacker to destruct the contract	Example
SVE10004	MissingCalleeCheck	The function may be missing a check callee != address(this)	Example
SVE1001	MissingSignerCheck	The account is missing signer check	Example
SVE1002	MissingOwnerCheck	The account is missing owner check	Example
SVE1003	IntegerAddOverflow	The add operation may result in overflows	Example
SVE1004	IntegerUnderflow	The sub operation may result in underflows	Example
SVE1005	IntegerMulOverflow	The mul operation may result in overflows	Example
SVE1006	IntegerDivOverflow	The div operation may result in overflows	Example
SVE1007	UnverifiedParsedAccount	The account is not validated before parsing its data	Example

SVE	Checker	Description	Examples
SVE1008	DuplicateMutableAccount	These two accounts are both mutable and may be the same account	Example
SVE1009	InsecureAccountClosing	The account may not be closed securely	Example
SVE1010	TypeFullCosplay	These two account data types are fully compatible and can be used to launch type confusion attacks	Example
SVE1011	TypePartialCosplay	These two account data types are partially compatible and may be exploited by type confusion attacks	Example
SVE1012	DivideByZero	The arithmetic operation may result in a divide-by-zero error	Example
SVE1013	AccountReInitialization	The account may be vulnerable to program re-initialization	Example
SVE1014	BumpSeedNotValidated	The account's bump seed is not validated and may be vulnerable to seed canonicalization attacks	Example
SVE1015	InsecurePDASharing	The PDA sharing with these seeds may be insecure	Example
SVE1016	ArbitraryCPI	The CPI may be vulnerable and invoke an arbitrary program	Example
SVE1017	MaliciousSimulation	The program may contain malicious simulation	Example

SVE	Checker	Description	Examples
SVE1018	UnsafeSysVarAPI	The sysvar instructions API is unsafe and deprecated (wormhole exploit)	Example
SVE1019	UnvalidatedAccount	The account may not be properly validated and may be untrustful	Example
SVE1020	OutdatedDependency	The program has outdated and vulnerable dependencies	Example
SVE1021	UnsafeRust	The program contains unsafe Rust code	Example
SVE1022	OverPayment	The code misses checking to prevent over payment	Example
SVE1023	StalePriceFeed	The code may use a stale price feed (solend loss)	Example
SVE1024	MissInitTokenMint	The init instruction misses minting pool tokens	Example
SVE1025	MissRentExempt	The account misses rent exempt check	Example
SVE1026	MissFreezeAuthority	The account misses checking for freeze authority	Example
SVE1027	FlashLoanRisk	The instruction may suffer from a flashloan risk due to internal price oracle	Example
SVE1028	BidirectionalRounding	The arithmetics here may suffer from bidirectional rounding vulnerabilities	Example
SVE1029	LossyCastTruncation	The cast operation here may lose precision due to truncation	Example
SVE1030	UnvalidatedPDAAccount	The PDA account may not be properly validated and may be untrustful	Example

SVE	Checker	Description	Examples
SVE1031	UnvalidatedDestinationAccount	The account is used as destination in token transfer without validation and it could be the same as the transfer source account	Example
SVE1032	IncorrectAuthorityAccount	The PDA account may be incorrectly used as shared authority and may allow any account to transfer or burn tokens	Example
SVE1033	InsecureAnchorInitIfNeeded	The `init_if_needed` keyword in anchor-lang prior to v0.24.x has a critical security bug	Example
SVE1034	InsecureSPLTokenCPI	The spl_token account may be arbitrary prior to version v3.1.1	Example
SVE1035	InsecureAssociatedTokenAccount	The associated token account is missing PDA key check and may be faked	Example
SVE1036	InsecureAccountRealloc	The account realloc in solana_program prior to v1.10.29 may cause programs to malfunction	Example
SVE1037	PDASeedCollisions	These two PDA accounts may have the same seeds, which may lead to PDA collisions	Example
SVE20001	MissingInitAdminCheck	The init function misses checking admin uniqueness and may allow an attacker to call the init function more than once	Example

SVE	Checker	Description	Examples
SVE20002	BitShiftOverflow	The bit shift operation may result in overflows	Example
SVE20003	DivisionPrecisionLoss	The division operation here may lose precision	Example
SVE20004	VulnerableI128Implementation	The I128 signed integer implementation in Move is not recommended and may be vulnerable. Consider using the built-in Move types only.	Example
SVE2001	IncorrectLoopBreakLogic	Loop break instead of continue (jet-v1 exploit)	Example
SVE2002	IncorrectConditionCheck	Liquidation condition >= should be >	Example
SVE2003	ExponentialCalculation	The calculation has exponential complexity	Example
SVE2004	IncorrectDivisionLogic	Incorrect checked_div instead of checked_ceil_div (spl-token-swap vulnerability: stable curve division)	Example
SVE2005	IncorrectTokenCalculation	The token amount calculation may be incorrect. Consider using the reserves instead of the balances.	Example
SVE3001	BestSecurityPractice	The code does not follow best security practices	Example
SVE3002	CriticalUnusedCode	The code may be redundant or unused, but appears critical	Example
SVE3003	InconsistentAnchor	The program uses Anchor inconsistently across different instructions	Example

SVE	Checker	Description	Examples
SVE3004	InconsistentConfig	The configuration and initialization data are inconsistent	Example
SVE3005	MissingCPIAccountReload	The token account's amount may be incorrect without calling reload after CPI	Example
SVE3006	MissingUnstakeAccessControl	The unstake instruction may be missing an access_control account validation	Example
SVE3007	OrderRaceCondition	The instruction may suffer from a race condition between order cancellation and order recreation by an attacker	Example
SVE3008	NewAccountNotBackwardsCompatible	The account may break the ABI of the deployed on-chain program as it does not exist in the IDL available on Anchor	Example
SVE3009	MutableAccountNotBackwardsCompatible	The mutable account may break the ABI of the deployed on-chain program as it is immutable according to the IDL available on Anchor	Example
SVE3010	ReOrderAccountsNotBackwardsCompatible	These two accounts are reordered in the instruction and may break the ABI of the deployed on-chain program, according to the IDL available on Anchor	Example

Appendix A - SVE List