Security Analysis Report

Hawksight Protocol v 0.0.1

Apr 6th, 2022

by Soteria Automatic Auditor Software (beta)

SOTERIA

Summary

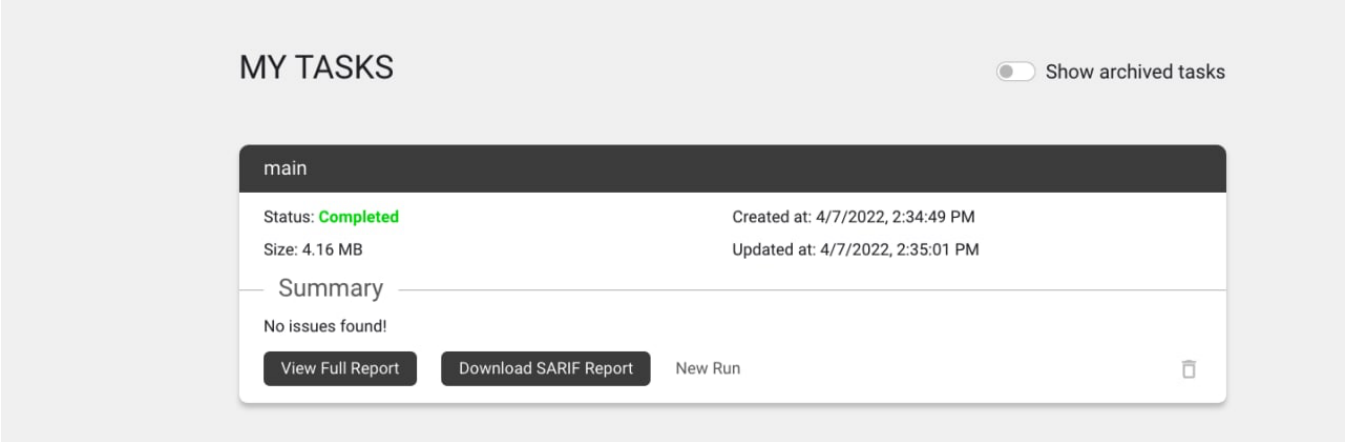
Soteria Auto Auditor Software ("**Soteria Software**") was used by Hawksight (the "**Client**") to conduct security analysis of the Hawksight Dapp V0.0.1 Solana smart contract program. The artifact of the analysis was the source code of the following on-chain smart contract excluding tests in a private repository:

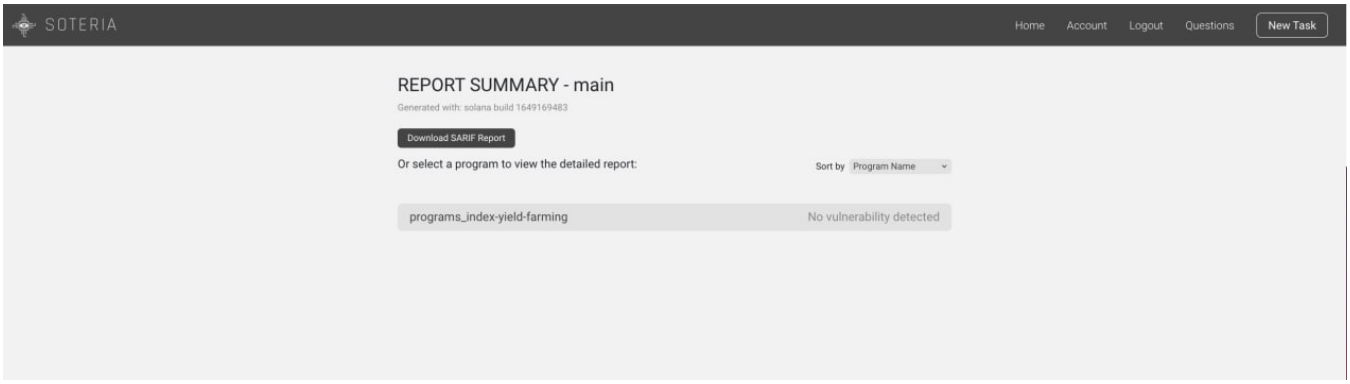
- Tag `v0.0.1`
- Commit `b6c82e33cea36e5c35a906c9d90e5e6b58f92abd`

The analysis revealed 0 potential issues.

This report presents the output from Soteria Software.

Analysis Output from Soteria Software





DISCLAIMER

This report ("**Report**") includes the results of a security analysis, by **Soteria Auto Auditor Software** (beta), 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 Soteria Auto Auditor Software

Soteria Auto Auditor Software (beta) extracts essential code structure and relationships into a set of databases that enable sophisticated analysis of source code. Soteria 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, Soteria Software can scan 36 types of security vulnerabilities, including Missing Signer Check, Missing Owner Check, etc. Please refer to Appendix A for more information.

About Soteria

Founded by leading academics in the field of software security and senior industrial veterans, Soteria 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 Soteria, 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.

Appendix A – Soteria Vulnerabilities and Exposures (SVE)

SVE	Checker	Description	Examples
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 is not securely closed	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 div-by-zero error	Example
SVE1013	AccountReInitialization	The account is 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

SVE	Checker	Description	Examples
SVE1016	ArbitraryCPI	The spl_token account may be arbitrary	Example
SVE1017	MaliciousSimulation	The program may contain malicious simulation	Example
SVE1018	UnsafeSysVarAPI	The sysvar instructions API is unsafe and deprecated (wormhole exploit)	Example
SVE1019	UnvalidatedAccount	The account is not 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

SVE	Checker	Description	Examples
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 flash loan attacks	Example
SVE1028	InconsistentRounding	The arithmetics here have inconsistent rounding	Example
SVE1029	CastTruncation	The cast operation here may lose precision due to truncation	Example
SVE2001	IncorrectLogic	Loop break instead of continue (jet-v1 exploit)	Example
SVE2002	IncorrectCalculation	Liquidation condition should be > instead of >=	Example
SVE2003	ExponentialCalculation	The calculation has exponential complexity	Example
SVE3001	BestSecurityPractice	The code does not follow best security practices	Example

SVE	Checker	Description	Examples
SVE3002	RedundantCode	The code is redundant or unused	Example
SVE3003	InconsistentAnchor	The program uses Anchor inconsistently across different instructions	Example
SVE3004	InconsistentConfig	The configuration and initialization data are inconsistent	Example