**Chapter Five**

**RESULTS AND DISCUSSION**

This section presents, analyzes, and interprets the results of the study in developing a Blockchain Implementation for Secured Vaccine Certificates. It also presents the result from OWASP ZAP and Slither Solidity Security Scans.

**5.1 Functionality of the Newly Developed System**

**5.1.1 Patient Registration**

Graphical user interface, application

Description automatically generated

This is the catalyst for a patient to be signed-up in the system. It will trigger generation of public and private keys in the backend which is crucial for encryption and uploading to IPFS.

**5.1.2 Patient Login**

Graphical user interface, application

Description automatically generated

The patient will be required to input email (as username) and password to be able to access related vaccine records and details.

**5.1.3 Patient Home Screen**

Qr code

Description automatically generated

Upper part of the home screen is the patient’s profile data: Full name, profile photo, address and patient code. Patient QR code will also be displayed. This code will be used later if patient decides to get vaccinated. It will signify that the patient is currently registered to the system.

Qr code

Description automatically generated

Lower part of the home screen is for vaccination details. If the patient already got vaccinated, this section will be displayed. It will show a summary of vaccine doses and a QR code for these details. This code can be used for validation of third party such as establishments to validated if patient was indeed vaccinated.

The ‘Download Certificate’ button will download the vaccine certificate file as follows:

Qr code

Description automatically generated

Graphical user interface, application

Description automatically generated**5.1.4 Vaccine Record Creation**

For medical personnel doing the vaccination, this page will be available for them. A valid Patient QR code is required before the system allows encoding of vaccination dose detail. Once details are confirmed, a MetaMask (blockchain plugin) will popup to confirm the transaction. This will log the transaction to the blockchain

**5.1.5 Vaccine Certificate Validation**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

For third-party validators that would want to authenticate a Vaccine Certificate file, this page will be available. It will require the user to upload the file and will display a prompt that would tell if file is valid or not (within the blockchain logs context)

Graphical user interface

Description automatically generated**5.1.6 Scan Summary QR Code**

Another way to validate is via summary QR code. For third-party validators that would want quick details of patient’s vaccine records, it will require a summary QR code from a patient and will display related details. QR code will be invalid if app founds out it’s not existing within the blockchain logs.

**5.2 Security Scans**

The researcher employed two scans to audit the application’s functionality. This will ensure that the application complies with existing standard for Solidity and Open Web Applications. Because the application size is fairly small, we will only focus the scans on the blockchain and web application aspects.

**Smart Contract Weakness Classification Registry**

The Smart Contract Weakness Classification Registry (SWC Registry) is an implementation of the weakness classification scheme proposed in [EIP-1470](https://github.com/ethereum/EIPs/issues/1469). It is loosely aligned to the terminologies and structure used in the Common Weakness Enumeration ([CWE](https://cwe.mitre.org/)) while overlaying a wide range of weakness variants that are specific to smart contracts.

The goals of this project are as follows:

* Provide a straightforward way to classify security issues in smart contract systems.
* Define a common language for describing security issues in smart contract systems' architecture, design, or code.
* Serve as a way to train and increase performance for smart contract security analysis tools.

The chosen scans, Securify and Slither, based their supported vulnerabilities from SWC Registry

**5.2.1 Securify**

The Securify tool is a static analyzer tool for Ethereum Solidity contracts. This tool scans the contract code and finds the security vulnerability patterns in the code. After scanning, it generates a report along with descriptions of each vulnerability it has found and provides an idea of how to solve each vulnerability.

**Supported Vulnerabilities**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Pattern name** | **Severity** | **Slither ID** | **SWC ID** |
| 1 | TODAmount | Critical | - | [SWC-114](https://swcregistry.io/docs/SWC-114) |
| 2 | TODReceiver | Critical | - | [SWC-114](https://swcregistry.io/docs/SWC-114) |
| 3 | TODTransfer | Critical | - | [SWC-114](https://swcregistry.io/docs/SWC-114) |
| 4 | UnrestrictedWrite | Critical | - | [SWC-124](https://swcregistry.io/docs/SWC-124) |
| 5 | RightToLeftOverride | High | rtlo | [SWC-130](https://swcregistry.io/docs/SWC-130) |
| 6 | ShadowedStateVariable | High | shadowing-state, shadowing-abstract | [SWC-119](https://swcregistry.io/docs/SWC-119) |
| 7 | UnrestrictedSelfdestruct | High | suicidal | [SWC-106](https://swcregistry.io/docs/SWC-106) |
| 8 | UninitializedStateVariable | High | uninitialized-state | [SWC-109](https://swcregistry.io/docs/SWC-109) |
| 9 | UninitializedStorage | High | uninitialized-storage | [SWC-109](https://swcregistry.io/docs/SWC-109) |
| 10 | UnrestrictedDelegateCall | High | controlled-delegatecall | [SWC-112](https://swcregistry.io/docs/SWC-112) |
| 11 | DAO | High | reentrancy-eth | [SWC-107](https://swcregistry.io/docs/SWC-107) |
| 12 | ERC20Interface | Medium | erc20-interface | - |
| 13 | ERC721Interface | Medium | erc721-interface | - |
| 14 | IncorrectEquality | Medium | incorrect-equality | [SWC-132](https://swcregistry.io/docs/SWC-132) |
| 15 | LockedEther | Medium | locked-ether | - |
| 16 | ReentrancyNoETH | Medium | reentrancy-no-eth | [SWC-107](https://swcregistry.io/docs/SWC-107) |
| 17 | TxOrigin | Medium | tx-origin | [SWC-115](https://swcregistry.io/docs/SWC-115) |
| 18 | UnhandledException | Medium | unchecked-lowlevel | - |
| 19 | UnrestrictedEtherFlow | Medium | unchecked-send | [SWC-105](https://swcregistry.io/docs/SWC-105) |
| 20 | UninitializedLocal | Medium | uninitialized-local | [SWC-109](https://swcregistry.io/docs/SWC-109) |
| 21 | UnusedReturn | Medium | unused-return | [SWC-104](https://swcregistry.io/docs/SWC-104) |
| 22 | ShadowedBuiltin | Low | shadowing-builtin | - |
| 23 | ShadowedLocalVariable | Low | shadowing-local | - |
| 24 | CallToDefaultConstructor? | Low | void-cst | - |
| 25 | CallInLoop | Low | calls-loop | [SWC-104](https://swcregistry.io/docs/SWC-104) |
| 26 | ReentrancyBenign | Low | reentrancy-benign | [SWC-107](https://swcregistry.io/docs/SWC-107) |
| 27 | Timestamp | Low | timestamp | [SWC-116](https://swcregistry.io/docs/SWC-116) |
| 28 | AssemblyUsage | Info | assembly | - |
| 29 | ERC20Indexed | Info | erc20-indexed | - |
| 30 | LowLevelCalls | Info | low-level-calls | - |
| 31 | NamingConvention | Info | naming-convention | - |
| 32 | SolcVersion | Info | solc-version | [SWC-103](https://swcregistry.io/docs/SWC-103) |
| 33 | UnusedStateVariable | Info | unused-state | - |
| 34 | TooManyDigits | Info | too-many-digits | - |
| 35 | ConstableStates | Info | constable-states | - |
| 36 | ExternalFunctions | Info | external-function | - |
| 37 | StateVariablesDefaultVisibility | Info | - | [SWC-108](https://swcregistry.io/docs/SWC-108) |

**Scan Result:**

Severity: MEDIUM

Pattern: Missing Input Validation

Description: Method arguments must be sanitized before they are used

in computations.

Type: Violation

Contract: Certificate

Line: 19

Source:

>

> function isFileHashUserIdExists(string calldata \_fileHash, uint256 \_userId)

> ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

> external

Severity: MEDIUM

Pattern: Missing Input Validation

Description: Method arguments must be sanitized before they are used

in computations.

Type: Violation

Contract: Certificate

Line: 33

Source:

>

> function saveUserIdHashes(

> ^^^^^^^^^^^^^^^^^^^^^^^^^^

> string calldata \_fileHash,

Severity: MEDIUM

Pattern: Missing Input Validation

Description: Method arguments must be sanitized before they are used

in computations.

Type: Violation

Contract: Certificate

Line: 45

Source:

>

> function isSummaryHashUserIdExists(

> ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

> string calldata \_summaryHash,

Severity: LOW

Pattern: Solidity pragma directives

Description: Avoid complex solidity version pragma statements.

Type: Violation

Contract: None

Line: 3

Source:

>

> pragma solidity >=0.5.0 <0.9.0;

> ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Reported Vulnerabilities:

|  |  |
| --- | --- |
| Critical | 0 |
| High | 0 |
| Medium | 3 |
| Low | 1 |
| Informational | 0 |
| Total | 4 |

All the three Medium vulnerabilities fall under “Missing Input Validation - Method arguments must be sanitized before they are used in computation”. The researcher chose to bypass this vulnerability as sanitation is already done on the web browser by using a front-end framework. The update/create transaction in Solidity is also guarded by an authorization modifier, thus only letting known entity to the blockchain make a successful transaction.

The single Low vulnerability is categorized on “Solidity pragma directives - Avoid complex solidity version pragma statements”. The researcher also skipped this vulnerability as this is not a security threat and more of a namespace convention or best practice. Since the application is still on Proof-of-Concept stage, the pragma versions used was a range to keep an open option when porting the testing from Remix (web) and local blockchain network.

**5.2.2 Slither**

Slither is a Solidity static analysis framework written in Python 3. It runs a suite of vulnerability detectors, prints visual information about contract details, and provides an API to easily write custom analyses. Slither enables developers to find vulnerabilities, enhance their code comprehension, and quickly prototype custom analyses.

**Supported Vulnerabilities**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Detector** | **What it Detects** | **Impact** | **Confidence** |
| 1 | abiencoderv2-array | [Storage abiencoderv2 array](https://github.com/crytic/slither/wiki/Detector-Documentation#storage-abiencoderv2-array) | High | High |
| 2 | array-by-reference | [Modifying storage array by value](https://github.com/crytic/slither/wiki/Detector-Documentation#modifying-storage-array-by-value) | High | High |
| 3 | incorrect-shift | [The order of parameters in a shift instruction is incorrect.](https://github.com/crytic/slither/wiki/Detector-Documentation#shift-parameter-mixup) | High | High |
| 4 | multiple-constructors | [Multiple constructor schemes](https://github.com/crytic/slither/wiki/Detector-Documentation#multiple-constructor-schemes) | High | High |
| 5 | name-reused | [Contract's name reused](https://github.com/crytic/slither/wiki/Detector-Documentation#name-reused) | High | High |
| 6 | public-mappings-nested | [Public mappings with nested variables](https://github.com/crytic/slither/wiki/Detector-Documentation#public-mappings-with-nested-variables) | High | High |
| 7 | rtlo | [Right-To-Left-Override control character is used](https://github.com/crytic/slither/wiki/Detector-Documentation#right-to-left-override-character) | High | High |
| 8 | shadowing-state | [State variables shadowing](https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing) | High | High |
| 9 | suicidal | [Functions allowing anyone to destruct the contract](https://github.com/crytic/slither/wiki/Detector-Documentation#suicidal) | High | High |
| 10 | uninitialized-state | [Uninitialized state variables](https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-state-variables) | High | High |
| 11 | uninitialized-storage | [Uninitialized storage variables](https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-storage-variables) | High | High |
| 12 | unprotected-upgrade | [Unprotected upgradeable contract](https://github.com/crytic/slither/wiki/Detector-Documentation#unprotected-upgradeable-contract) | High | High |
| 13 | arbitrary-send | [Functions that send Ether to arbitrary destinations](https://github.com/crytic/slither/wiki/Detector-Documentation#functions-that-send-ether-to-arbitrary-destinations) | High | Medium |
| 14 | controlled-array-length | [Tainted array length assignment](https://github.com/crytic/slither/wiki/Detector-Documentation#array-length-assignment) | High | Medium |
| 15 | controlled-delegatecall | [Controlled delegatecall destination](https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall) | High | Medium |
| 16 | delegatecall-loop | [Payable functions using delegatecall inside a loop](https://github.com/crytic/slither/wiki/Detector-Documentation/#payable-functions-using-delegatecall-inside-a-loop) | High | Medium |
| 17 | msg-value-loop | [msg.value inside a loop](https://github.com/crytic/slither/wiki/Detector-Documentation/#msgvalue-inside-a-loop) | High | Medium |
| 18 | reentrancy-eth | [Reentrancy vulnerabilities (theft of ethers)](https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities) | High | Medium |
| 19 | storage-array | [Signed storage integer array compiler bug](https://github.com/crytic/slither/wiki/Detector-Documentation#storage-signed-integer-array) | High | Medium |
| 20 | unchecked-transfer | [Unchecked tokens transfer](https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer) | High | Medium |
| 21 | weak-prng | [Weak PRNG](https://github.com/crytic/slither/wiki/Detector-Documentation#weak-PRNG) | High | Medium |
| 22 | enum-conversion | [Detect dangerous enum conversion](https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-enum-conversion) | Medium | High |
| 23 | erc20-interface | [Incorrect ERC20 interfaces](https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-erc20-interface) | Medium | High |
| 24 | erc721-interface | [Incorrect ERC721 interfaces](https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-erc721-interface) | Medium | High |
| 25 | incorrect-equality | [Dangerous strict equalities](https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities) | Medium | High |
| 26 | locked-ether | [Contracts that lock ether](https://github.com/crytic/slither/wiki/Detector-Documentation#contracts-that-lock-ether) | Medium | High |
| 27 | mapping-deletion | [Deletion on mapping containing a structure](https://github.com/crytic/slither/wiki/Detector-Documentation#deletion-on-mapping-containing-a-structure) | Medium | High |
| 28 | shadowing-abstract | [State variables shadowing from abstract contracts](https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing-from-abstract-contracts) | Medium | High |
| 29 | tautology | [Tautology or contradiction](https://github.com/crytic/slither/wiki/Detector-Documentation#tautology-or-contradiction) | Medium | High |
| 30 | write-after-write | [Unused write](https://github.com/crytic/slither/wiki/Detector-Documentation#write-after-write) | Medium | High |
| 31 | boolean-cst | [Misuse of Boolean constant](https://github.com/crytic/slither/wiki/Detector-Documentation#misuse-of-a-boolean-constant) | Medium | Medium |
| 32 | constant-function-asm | [Constant functions using assembly code](https://github.com/crytic/slither/wiki/Detector-Documentation#constant-functions-using-assembly-code) | Medium | Medium |
| 33 | constant-function-state | [Constant functions changing the state](https://github.com/crytic/slither/wiki/Detector-Documentation#constant-functions-changing-the-state) | Medium | Medium |
| 34 | divide-before-multiply | [Imprecise arithmetic operations order](https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply) | Medium | Medium |
| 35 | reentrancy-no-eth | [Reentrancy vulnerabilities (no theft of ethers)](https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1) | Medium | Medium |
| 36 | reused-constructor | [Reused base constructor](https://github.com/crytic/slither/wiki/Detector-Documentation#reused-base-constructors) | Medium | Medium |
| 37 | tx-origin | [Dangerous usage of tx.origin](https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-usage-of-txorigin) | Medium | Medium |
| 38 | unchecked-lowlevel | [Unchecked low-level calls](https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-low-level-calls) | Medium | Medium |
| 39 | unchecked-send | [Unchecked send](https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-send) | Medium | Medium |
| 40 | uninitialized-local | [Uninitialized local variables](https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables) | Medium | Medium |
| 41 | unused-return | [Unused return values](https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return) | Medium | Medium |
| 42 | incorrect-modifier | [Modifiers that can return the default value](https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-modifier) | Low | High |
| 43 | shadowing-builtin | [Built-in symbol shadowing](https://github.com/crytic/slither/wiki/Detector-Documentation#builtin-symbol-shadowing) | Low | High |
| 44 | shadowing-local | [Local variables shadowing](https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing) | Low | High |
| 45 | uninitialized-fptr-cst | [Uninitialized function pointer calls in constructors](https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-function-pointers-in-constructors) | Low | High |
| 46 | variable-scope | [Local variables used prior their declaration](https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables) | Low | High |
| 47 | void-cst | [Constructor called not implemented](https://github.com/crytic/slither/wiki/Detector-Documentation#void-constructor) | Low | High |
| 48 | calls-loop | [Multiple calls in a loop](https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-a-loop) | Low | Medium |
| 49 | events-access | [Missing Events Access Control](https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-access-control) | Low | Medium |
| 50 | events-maths | [Missing Events Arithmetic](https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic) | Low | Medium |
| 51 | incorrect-unary | [Dangerous unary expressions](https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-unary-expressions) | Low | Medium |
| 52 | missing-zero-check | [Missing Zero Address Validation](https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation) | Low | Medium |
| 53 | reentrancy-benign | [Benign reentrancy vulnerabilities](https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2) | Low | Medium |
| 54 | reentrancy-events | [Reentrancy vulnerabilities leading to out-of-order Events](https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3) | Low | Medium |
| 55 | timestamp | [Dangerous usage of block.timestamp](https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp) | Low | Medium |
| 56 | assembly | [Assembly usage](https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage) | Informational | High |
| 57 | assert-state-change | [Assert state change](https://github.com/crytic/slither/wiki/Detector-Documentation#assert-state-change) | Informational | High |
| 58 | boolean-equal | [Comparison to boolean constant](https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality) | Informational | High |
| 59 | deprecated-standards | [Deprecated Solidity Standards](https://github.com/crytic/slither/wiki/Detector-Documentation#deprecated-standards) | Informational | High |
| 60 | erc20-indexed | [Un-indexed ERC20 event parameters](https://github.com/crytic/slither/wiki/Detector-Documentation#unindexed-erc20-event-parameters) | Informational | High |
| 61 | function-init-state | [Function initializing state variables](https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state) | Informational | High |
| 62 | low-level-calls | [Low level calls](https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls) | Informational | High |
| 63 | missing-inheritance | [Missing inheritance](https://github.com/crytic/slither/wiki/Detector-Documentation#missing-inheritance) | Informational | High |
| 64 | naming-convention | [Conformity to Solidity naming conventions](https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions) | Informational | High |
| 65 | pragma | [If different pragma directives are used](https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used) | Informational | High |
| 66 | redundant-statements | [Redundant statements](https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements) | Informational | High |
| 67 | solc-version | [Incorrect Solidity version](https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity) | Informational | High |
| 68 | unimplemented-functions | [Unimplemented functions](https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions) | Informational | High |
| 69 | unused-state | [Unused state variables](https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable) | Informational | High |
| 70 | costly-loop | [Costly operations in a loop](https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop) | Informational | Medium |
| 71 | dead-code | [Functions that are not used](https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code) | Informational | Medium |
| 72 | reentrancy-unlimited-gas | [Reentrancy vulnerabilities through send and transfer](https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-4) | Informational | Medium |
| 73 | similar-names | [Variable names are too similar](https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar) | Informational | Medium |
| 74 | too-many-digits | [Conformance to numeric notation best practices](https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits) | Informational | Medium |
| 75 | constable-states | [State variables that could be declared constant](https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant) | Optimization | High |
| 76 | external-function | [Public function that could be declared external](https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external) | Optimization | High |

**Scan Result:**

Pragma version>=0.5.0<0.9.0 (Certificate2.sol#3) is too complex

Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Certificate2.sol analyzed (1 contracts with 75 detectors), 2 result(s) found

Reported Vulnerabilities:

|  |  |
| --- | --- |
| Critical | 0 |
| High | 0 |
| Medium | 0 |
| Low | 0 |
| Informational | 1 |
| Total | 1 |

The single vulnerability found is “Pragma version is too complex” and is categorized under “Informational”. This is the same finding as with Securify scan pertaining to pragma version.