HACKEN

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Customer: Premia
Date: September 15th, 2021

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Document

Name	Smart Contract Code Review and Security Analysis Report for Premia-Audit Report
Approved by	Andrew Matiukhin | CTO Hacken OU
Type	Libraries
Platform	Ethereum / Solidity
Methods	Architecture Review, Functional Testing, Computer-Aided Verification, Manual Review
Git repository	https://github.com/solidstate-network/solidstate-solidity/tree/e23bd67e28a8435ea9f2cca2cfa2c87cd7febd18
Timeline	30 JUNE 2021 - 15 SEPTEMBER 2021
Changelog	7 JULY 2021 - INITIAL AUDIT 15 SEPTEMBER 2021 - SECOND REVIEW

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Table of contents

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Introduction

Hacken OÜ (Consultant) was contracted by Premia (Customer) to conduct a Smart Contract Code Review and Security Analysis. This report presents the findings of the security assessment of Customer's smart contract and its code review conducted on September 15th, 2021.

Scope

The scope of the project is the smart contracts in Git repository:

```
https://github.com/solidstate-network/solidstate-
solidity/tree/e23bd67e28a8435ea9f2cca2cfa2c87cd7febd18

Scope:
access
    IERC173.sol                                 md5: 699641ce6870af9ebb29937eaa1038a8
    Ownable.sol                                 md5: 78e7e0da1819b5cf034c7970a632a804
    OwnableInternal.sol                         md5: 77a017c4972bb8e4981fde51bb288897
    OwnableStorage.sol                          md5: b3ee2917624bb07d3393db9de29839a7
    SafeOwnable.sol                             md5: 501ed4cc07708f6f1c7af9aad8fa24e2
    SafeOwnableInternal.sol                     md5: 63be404448441c48dc6923dbe193ef83
    SafeOwnableStorage.sol                      md5: ca9a8f7c80b860aefbfed75902468dd6
cryptography
    ECDSA.sol                                   md5: ee91173f818a174a78bbfcf52e99ba72
factory
    CloneFactory.sol                            md5: 68796b62de8ec6f138430648b5b5f82b
    Factory.sol                                 md5: 4538c7d8c13b497c34d107a1f865d774
    MetamorphicFactory.sol                      md5: 91fedbc4d7e9ad0d436ab28b4dca3543
    MetamorphicFactoryStorage.sol               md5: 11d9c8deac475485430ebd162d698ed5
    MinimalProxyFactory.sol                     md5: e94890c318eae280335583d01a9e0ff6
introspection
    ERC165.sol                                  md5: 01b086278a9b7f4366c2b13a01a706c2
    ERC165Storage.sol                           md5: c8228e7b63d90cee8ad9d2788bb9bb5b
    IERC165.sol                                 md5: 5052ec684f450bd2bd7162e6aec4b3d4
multisig
    ECDSAMultisigWallet.sol                     md5: def602f841ac86868eebedca1a82ca77
    ECDSAMultisigWalletStorage.sol              md5: 2089debbfbf3b00326a83434bfa781c2
proxy
    diamond
        Diamond.sol                             md5: 1f57aa8f8f453fbc34e4b78a42b06b5f
        DiamondBase.sol                         md5: 14ffca0a92600c93b47614a0c94fcc58
        DiamondBaseStorage.sol                  md5: 625587974f0108828dd02173f2713486
        DiamondCuttable.sol                     md5: 163ee7b24931f148a0a22ea400e5066f
        DiamondLoupe.sol                        md5: 9a70eb910a01e628aac85fbd360d6feb
        IDiamondCuttable.sol                    md5: 50d66452c6ce7d7c479df2b2f3f8d7ba
        IDiamondLoupe.sol                       md5: 55b62c729403e7ed29aeef5ce25e72d3
    managed
        ManagedProxy.sol                        md5: 26f7403590c56a84cbe7a3d77bea77d0
        ManagedProxyOwnable.sol                 md5: b028a6df790a5d9214d2a62f2f10e21d
    Proxy.sol                                   md5: 9c3ccd780b82cf4f659f00b2b2c60bfb
signature
    ERC1271Base.sol                             md5: 2768fc79c60193c0054629213c0cd039
    ERC1271Ownable.sol                          md5: 5175b6f60da8cac10c8a87f8778ac318
    ERC1271Stored.sol                           md5: bedc1ed8aa859901adcdcaaff1c615e2
```

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

```
    ERC1271StoredStorage.sol                 md5: d68140d4d5885729b46df1f552358ded
    IERC1271.sol                             md5: e9fa5c1422cb11122d9184aae8eef580
token
    ERC20
        ERC20                                md5: 918208b4fdbb840fa0a869836fbaa3d0
        ERC20Base                            md5: dca0741ba2b67ecd78d16b2e1356b53e
        ERC20BaseStorage                     md5: 6c9dcbd97f0af529d7bf7b1f5fafd924
        ERC20Extended                        md5: 72849e2870c2fc7b2b95866c085294d4
        ERC20ImplicitApproval                md5: 613efe276cd29f475be57539b2324d45
        ERC20ImplicitApprovalStorage         md5: 9ec6578d1b4dc52346df0037ad80c5f0
        ERC20Metadata                        md5: 5b801d9b0eae7a5101f460a8e87ed37d
        ERC20MetadataStorage                 md5: 437b363df8e7ac040705bc933d72df44
        ERC20Mock                            md5: 690927fbd9c94c2078ac031b799b0e81
        ERC20Permit                          md5: 45d9e6cb4d89fa967fa01332881c22f8
        ERC20PermitStorage                   md5: 04ea4eaa957263da25118bc986a6c854
        ERC20Snapshot                        md5: 8159665c653c8f51296ffce71c26b74e
        ERC20SnapshotStorage                 md5: 7f98801ba01ca5deb0334b5f337cddaa
        IERC20                               md5: 6ece315959fdb94316075d8a5a17fa6d
        IERC20Metadata                       md5: 608d816b8b63af2016da6c9ae4ce7eeb
        IERC2612                             md5: dac6d185ab00f99fae82430fd3d76dab
    ERC1155
        ERC1155.sol                          md5: 09bb2891abb71d4fc65c004166202b5f
        ERC1155Base.sol                      md5: 1ccbb6f7b7cb727316dc60ef608e2aac
        ERC1155BaseStorage.sol               md5: 7b64673c87f941fe93f22d94f51b10a1
        ERC1155Enumerable.sol                md5: 6b3a360e8335769b90531fdf5556c95f
        ERC1155EnumerableStorage.sol         md5: 411835dd8ebe1e1da4da2c14b0b33460
        IERC1155.sol                         md5: 0e4b5f388749daddf223da035c5921e8
        IERC1155Enumerable.sol               md5: 94c94aed76e1492785b2f4161fca6d9b
        IERC1155Receiver.sol                 md5: 720dbb2ddd482ebe71bc5702bc5fe446
    ERC1404
        ERC1404.sol                          md5: 98fe11317cc54a386507ad3b3ad67bd1
        ERC1404Base.sol                      md5: 52def4d5f8eb49e600fa429244ba2e51
        ERC1404Storage.sol                   md5: b7877dd6d1a41a7a61a7973520c90803
        IERC1404.sol                         md5: d2516b9ee8c5cc852a65f6a28b63552a
utils
   AddressUtils.sol                          md5: 55656181f92ac06ef926e4ca7de3ba6f
   Array.sol                                 md5: c8fe5a859f0d5db2fcf11dbd1e69e8d0
   EnumerableMap.sol                         md5: 48a1a863caaf2c3d935b7162cc9be8d5
   EnumerableSet.sol                         md5: 66c0307050bbe444f20d4e24cdc0e619
   IMulticall.sol                            md5: fc6d148df26538c0239f0555b30db781
   IWETH.sol                                 md5: d1d00ec1b6e5df65b101af35e3b77fa0
   Math.sol                                  md5: 31636344a6144930c78d8e8cc5e34791
   Multicall.sol                             md5: 2b3359f564adbd858280c0577da2e749
   MulticallMock.sol                         md5: 6ceb9bad3e83982788241121d124ccb2
   ReentrancyGuard.sol                       md5: 7ca02fc2e2f55d314ba0d79b16c1b836
   ReentrancyGuardMock.sol                   md5: 19a98ed3fada0753ad4d8297b7913c2d
   ReentrancyGuardStorage.sol                md5: e6ddf93991f847247b36f53376d742d0
   SafeCast.sol                              md5: 207ff46a858928d744f8182faf822c20
   SafeERC20.sol                             md5: 1a430b9ebac9605c71573b55706c56ca
   UintUtils.sol                             md5: f0957b4073fdb5d5b4e4a4637f72fc97
```

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

We have scanned these smart contracts for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that are considered:

Category	Check Item
Code review	▪ Reentrancy ▪ Ownership Takeover ▪ Timestamp Dependence ▪ Gas Limit and Loops ▪ DoS with (Unexpected) Throw ▪ DoS with Block Gas Limit ▪ Transaction-Ordering Dependence ▪ Style guide violation ▪ Costly Loop ▪ ERC20 API violation ▪ Unchecked external call ▪ Unchecked math ▪ Unsafe type inference ▪ Implicit visibility level ▪ Deployment Consistency ▪ Repository Consistency ▪ Data Consistency
Functional review	▪ Business Logics Review ▪ Functionality Checks ▪ Access Control & Authorization ▪ Escrow manipulation ▪ Token Supply manipulation ▪ Asset’s integrity ▪ User Balances manipulation ▪ Kill-Switch Mechanism ▪ Operation Trails & Event Generation

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Executive Summary

According to the assessment, the Customer's smart contracts are secured but have some styling and other recommendations.

Insecure	Poor secured	Secured	Well-secured

You are here

Our team performed an analysis of code functionality, manual audit, and automated checks with Mythril and Slither. All issues found during automated analysis were manually reviewed, and important vulnerabilities are presented in the Audit overview section. All found issues can be found in the Audit overview section.

Security engineers found **3** low and **3** informational issues during the first review.

After the second review security engineers found that all issues were **resolved**.

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to assets loss or data manipulations.
High	High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to assets loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution
Lowest / Code Style / Best Practice	Lowest-level vulnerabilities, code style violations, and info statements can't affect smart contract execution and can be ignored.

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Audit overview

Critical

No Critical severity issues were found.

High

No High severity issues were found.

Medium

No Medium severity issues were found.

Low

1. **Issue**: Lack of comments

   Comments in libraries and abstract contracts, which are written to be inherited by others, are very important.

   Fixed before the second review

2. **Issue:** Return value ignored

   ECDSAMultisigWalletStorage ignores return value of the methods EnumerableSet.add and EnumerableSet.remove. Theoretically, in the case there would be an issue and method would return false, storage will continue as if the operation was successful

   Recommendation: Please consider adding a return value check

   Fixed before the second review

3. **Issue:** Return value ignored

   ERC1155Enumerable ignores return value of the methods EnumerableSet.add and EnumerableSet.remove. Theoretically, in the case there would be an issue and the method would return false, storage will continue as if the operation was successful

   Recommendation: Please consider adding a return value check

   Fixed before the second review

Lowest / Code style / Best Practice

1. **Issue:** Code Layout

   Solidity has recommendations about code layout. That’s about

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

indentation, blank lines, maximum line length, order of functions, etc.

Recommendation: Please consider following solidity code layout recommendations.

Fixed before the second review

2. **Issue:** Contract could be abstract

   Some contracts could be declared abstract since those should never be deployed stand-alone.

   Such contracts are:
   - Ownable
   - SafeOwnable
   - DiamondCuttable
   - DiamondLoupe
   - ERC20Snapshot
   - ERC1155Enumerable

   Recommendation: Please consider declaring such contracts abstract

   Fixed before the second review

3. **Issue:** Conformance to Solidity naming conventions

   Solidity defines a naming convention that should be followed. Constants should be written in UPPER_CASE_WITH_UNDERSCORES

   Recommendation: Please consider following the Solidity naming convention.

   Fixed before the second review

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Conclusion

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools.

The audit report contains all found security vulnerabilities and other issues in the reviewed code.

Security engineers found **2** low and **2** informational issues during the first review.

After the second review security engineers found that all issues were **resolved**.

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Disclaimers

Hacken Disclaimer

The smart contracts given for audit have been analyzed in accordance with the best industry practices at the date of this report, in relation to cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The audit makes no statements or warranties on security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bugfree status or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only - we recommend proceeding with several independent audits and a public bug bounty program to ensure security of smart contracts.

Technical Disclaimer

Smart contracts are deployed and executed on the blockchain platform. The platform, its programming language, and other software related to the smart contract can have its vulnerabilities that can lead to hacks. Thus, the audit can't guarantee the explicit security of the audited smart contracts.