



**SOLIDProof**  
*Bring trust into your projects*

**Blockchain Security | Smart Contract Audits | KYC**

MADE IN GERMANY

**StellaSwap**

-

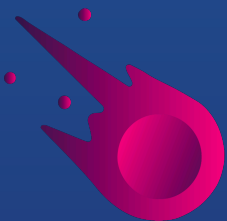
**Stable AMM**

**Audit**

**Security Assessment**

**20. April, 2022**

**For**



**StellaSwap**

Disclaimer	3
Description	5
Project Engagement	5
Logo	5
Contract Link	5
Methodology	7
Used Code from other Frameworks/Smart Contracts (direct imports)	8
Tested Contract Files	9
Source Lines	10
Risk Level	10
Capabilities	11
Inheritance Graph	13
CallGraph	14
Scope of Work/Verify Claims	15
Modifiers and public functions	21
Source Units in Scope	23
Critical issues	25
High issues	25
Medium issues	25
Low issues	25
Informational issues	27
Commented Code exist	29
Audit Comments	29
SWC Attacks	30

# Disclaimer

SolidProof.io reports are not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team. SolidProof.io do not cover testing or auditing the integration with external contract or services (such as Uniswap, Uniswap, PancakeSwap etc’...)

**SolidProof.io Audits do not provide any warranty or guarantee regarding the absolute bug- free nature of the technology analyzed, nor do they provide any indication of the technology proprietors. SolidProof Audits should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.**

SolidProof.io Reports represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology. Blockchain technology and cryptographic assets present a high level of ongoing risk. SolidProof’s position is that each company and individual are responsible for their own due diligence and continuous security. SolidProof in no way claims any guarantee of security or functionality of the technology we agree to analyze.

Version	Date	Description
1.0	19. April 2022	<ul style="list-style-type: none"><li>• Layout project</li><li>• Automated- /Manual-Security Testing</li><li>• Summary</li></ul>
	20. Audit 2022	<ul style="list-style-type: none"><li>• Check layout of source code</li><li>• Finishing report</li></ul>

## **Network**

Moonbeam (Polkadot)

## **Website**

<https://stellaswap.com/>

## **Telegram**

<https://t.me/stellaswap>

## **Twitter**

<https://twitter.com/StellaSwap>

## **Github**

<https://github.com/stellaswap>

## **Reddit**

<https://www.reddit.com/user/stellaswap>

## **Medium**

<https://stellaswap.medium.com/>

## Description

All your DeFi needs in one place.

Swap, earn and build on Moonbeam's leading DEX

## Project Engagement

During the 18th of April 2022, **StellaSwap Team** engaged Solidproof.io to audit smart contracts that they created. The engagement was technical in nature and focused on identifying security flaws in the design and implementation of the contracts. They provided Solidproof.io with access to their code repository and whitepaper.

### Logo



# StellaSwap

### Contract Link

**v1.0**

- Github
  - <https://github.com/stellaswap/stable-amm/>
  - Commit: 9bd2d5deece603d50e1391a7cd4171efa2d2d43f

# Vulnerability & Risk Level

Risk represents the probability that a certain source-threat will exploit vulnerability, and the impact of that event on the organization or system. Risk Level is computed based on CVSS version 3.0.

Level	Value	Vulnerability	Risk (Required Action)
<b>Critical</b>	9 - 10	A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.	Immediate action to reduce risk level.
<b>High</b>	7 - 8.9	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.	Implementation of corrective actions as soon as possible.
<b>Medium</b>	4 - 6.9	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.	Implementation of corrective actions in a certain period.
<b>Low</b>	2 - 3.9	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.	Implementation of certain corrective actions or accepting the risk.
<b>Informational</b>	0 - 1.9	A vulnerability that have informational character but is not effecting any of the code.	An observation that does not determine a level of risk

# Auditing Strategy and Techniques Applied

Throughout the review process, care was taken to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices. To do so, reviewed line-by-line by our team of expert pentesters and smart contract developers, documenting any issues as there were discovered.

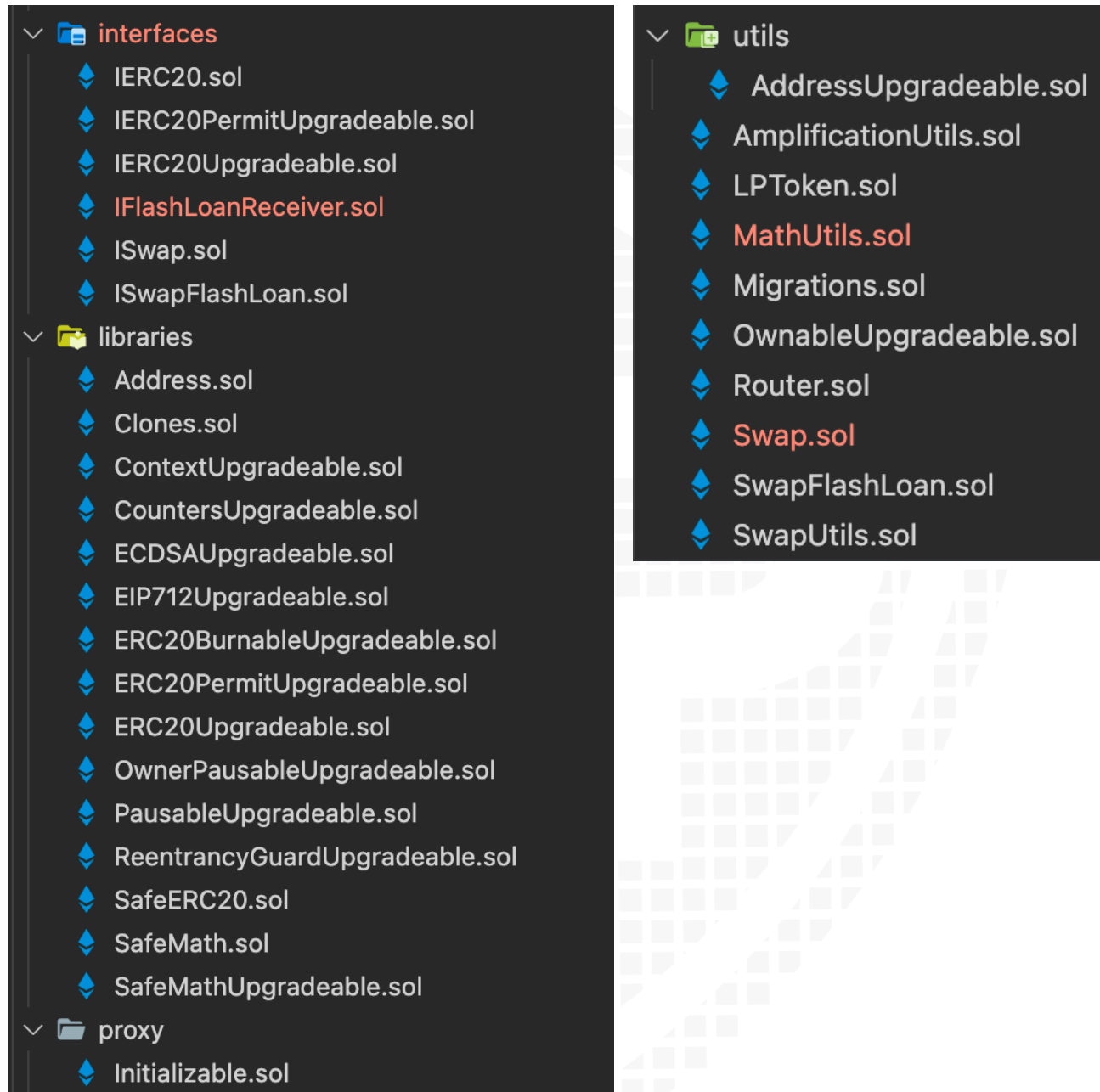
## Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:
  - i) Review of the specifications, sources, and instructions provided to SolidProof to make sure we understand the size, scope, and functionality of the smart contract.
  - ii) Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
  - iii) Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to SolidProof describe.
2. Testing and automated analysis that includes the following:
  - i) Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
  - ii) Symbolic execution, which is analysing a program to determine what inputs causes each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, actionable recommendations to help you take steps to secure your smart contracts.

## Used Code from other Frameworks/Smart Contracts (direct imports)

Imported packages:





# Tested Contract Files

This audit covered the following files listed below with a SHA-1 Hash.

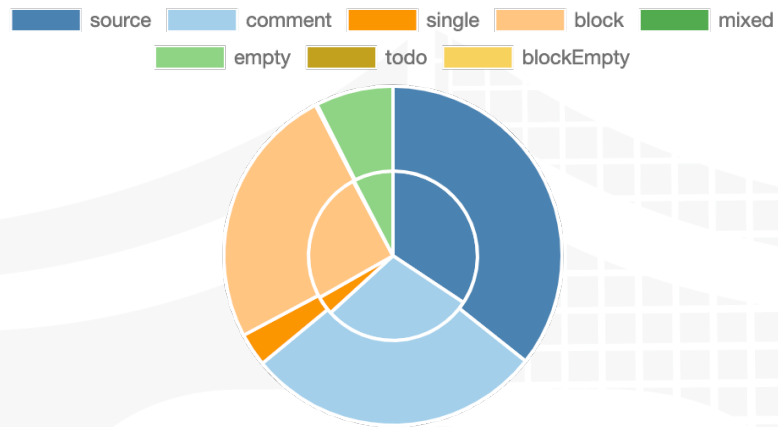
*A file with a different Hash has been modified, intentionally or otherwise, after the security review. A different Hash could be (but not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of this review.*

## v1.0

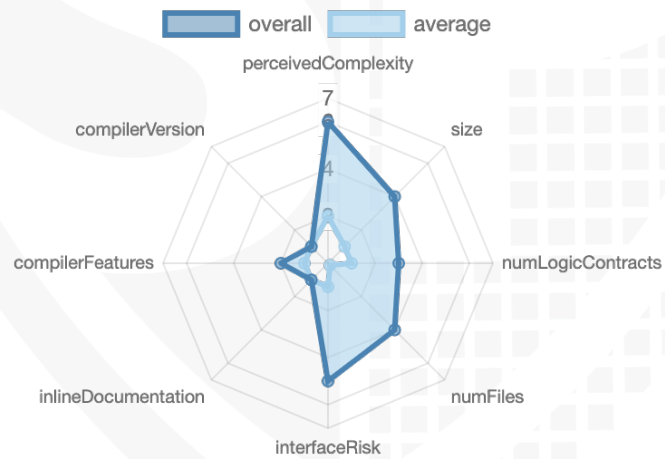
File Name	SHA-1 Hash
contracts/SwapUtils.sol	c3cfed5e082d250cc6fa16911050c5200960933c
contracts/interfaces/IERC20Upgradeable.sol	b310903de7a32c0cd3631971a9113188ac411347
contracts/interfaces/IERC20PermitUpgradeable.sol	cc21709cb288cc8bb1c80c680bc3c11760fb1e15
contracts/interfaces/ISwapFlashLoan.sol	1c27e6a929372be200a3ac4639d5c7fd2ff819b
contracts/interfaces/ISwap.sol	ccc0080c08d4cded28f12f16d9246f5bb8b86715
contracts/interfaces/IERC20.sol	2d6eb8a102a8a92dcfef4d19c977a062e891c7cf
contracts/interfaces/IFlashLoanReceiver.sol	bb153792ac1068f3d6897adafc3356156741dff5
contracts/helpers/GenericERC20.sol	77cbe7441448d8e8fca3f59e66396e71d002faf8
contracts/helpers/FlashLoanBorrowerExample.sol	a3cb61710f0cc97ffb5c05986cc6a03b97b2d982
contracts/Swap.sol	eef3bccda300188ce74bcf39da99ff2fb35e1784
contracts/Router.sol	3a35eba91a0c60a89cb97746c98e7e18688518a6
contracts/LPToken.sol	73b9a6294622f963788e10e6e445d3fcb7f9e089
contracts/Utils/AddressUpgradeable.sol	522674d9d63da6c735286f3a85f9bbd4d8c8af02
contracts/libraries/ECDsaUpgradeable.sol	2209ef889686657878c1b694971f730e4ef3050
contracts/libraries/CountersUpgradeable.sol	f3e7cee73ddb9ff685c70832e271b823e62b7f60
contracts/libraries/SafeMathUpgradeable.sol	c3fc02887779f0de3ab4c368a5bd8c6196b1926
contracts/libraries/ERC20PermitUpgradeable.sol	698875230444a1d6f5755c6bae5dc9a6af1d5817
contracts/libraries/ERC20BurnableUpgradeable.sol	340a37f7e751cb73639652a5e0f01deca66357e5
contracts/libraries/ReentrancyGuardUpgradeable.sol	4294648e9b18b56b640790229feedbf9c9817464
contracts/libraries/ContextUpgradeable.sol	7c8676f62f5224a79b5f8d9cc0639a6914c27c42
contracts/libraries/PausableUpgradeable.sol	bf27015d9bc6025dd9117e056d121a927cc97355
contracts/libraries/Address.sol	1aea000b5e51774d55f9db58553f5d25caeb0b
contracts/libraries/EIP712Upgradeable.sol	9d010c203bc15504e902e7cc99937c2e74c94d93
contracts/libraries/OwnerPausableUpgradeable.sol	2de27f6db029b91bc968f11c805436be9123dc38
contracts/libraries/SafeMath.sol	252b3caeb72fa4bde1cf723d04677a593bd82d36
contracts/libraries/Clones.sol	2b50bb3d43f711b4ec12fc68120f65c497795bf
contracts/libraries/SafeERC20.sol	2e2dfbf4f2ac98115ef92774ad58b6bcccc80d19
contracts/libraries/ERC20Upgradeable.sol	92455b72bde577f3dcd8870538fcb57ece55085e
contracts/OwnableUpgradeable.sol	cf71edff7ed1d323f1969cb1edf00b76cb496ea
contracts/MathUtils.sol	f644331c463b88117e279d2e6c64aaae25275db
contracts/proxy/Initializable.sol	b4ea9be81cf5755124ca4c530b62d35721708bdd
contracts/AmplificationUtils.sol	91bc6dcbafe34f931cd4e02d9ec1265d499ab9f
contracts/SwapFlashLoan.sol	f7171b20368ae37ce4d3b4cfbbe51a86ec9acb0d

# Metrics

## Source Lines v1.0



## Risk Level v1.0



## Capabilities

### Components

Version	Contracts	Libraries	Interfaces	Abstract
1.0	7	11	6	9

### Exposed Functions

*This section lists functions that are explicitly declared public or payable. Please note that getter methods for public stateVars are not included.*

Version	Public	Payable
1.0	123	0

Version	External	Internal	Private	Pure	View
1.0	98	218	6	40	83

## State Variables

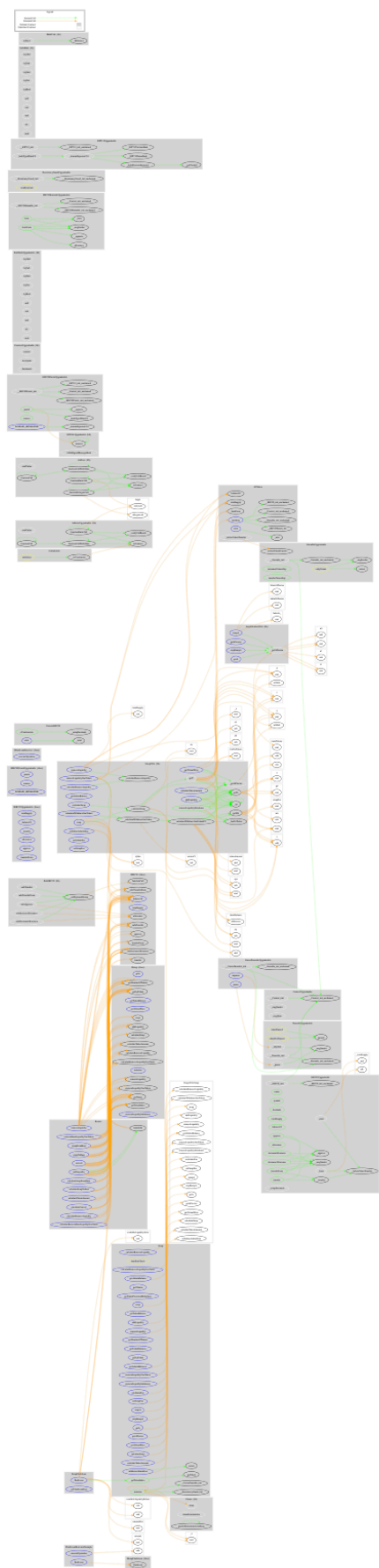
Version	Total	Public
1.0	40	9

## Capabilities

Version	Solidity Versions observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
1.0	0.6.12 >=0.6.0 <0.8.0 0.8.4 >=0.6.2 <0.8.0 >=0.6.5 <0.8.0 >=0.4.24 <0.8.0			yes (9 asm blocks)	

Version	Transfers ETH	Low-Level Calls	DelegateCall	Uses Hash Functions	EC Recover	New/Create/Create2
---------	---------------	-----------------	--------------	---------------------	------------	--------------------





## Scope of Work/Verify Claims

The above token Team provided us with the files that needs to be tested (Github, Bscscan, Etherscan, files, etc.). The scope of the audit is the main contract (usual the same name as team appended with .sol).

We will verify the following claims:

1. Overall checkup (Smart Contract Security)



## Write functions of contract v1.0

Swap/SwapFlashLoan

```
initialize  
flashLoan  
setFlashLoanFees  
initialize  
swap  
addLiquidity  
removeLiquidity  
removeLiquidityOneToken  
removeLiquidityImbalance  
withdrawAdminFees  
setAdminFee  
setSwapFee  
rampA  
stopRampA  
pause  
unpause  
renounceOwnership  
transferOwnership
```

Router

```
convert  
addLiquidity  
removeLiquidity  
removeBaseLiquidityOneToken  
swapFromBase  
swapToBase
```

LPToken

```
initialize  
mint  
burn  
burnFrom  
transfer  
approve  
transferFrom  
increaseAllowance  
decreaseAllowance  
permit
```



## Deployer cannot mint any new tokens

Name	Exist	Tested	Status
Deployer cannot mint	✓	✓	✗

Comments:

### v1.0

- LPToken
  - Owner can mint new tokens
- GenericERC20
  - Owner can mint new tokens

## Deployer cannot burn or lock user funds

Name	Exist	Tested	Status
Deployer cannot lock	✓	✓	✓
Deployer cannot burn	✓	✓	✓

Comments:

**v1.0**

- LPToken
  - Everybody can burn own tokens

## Deployer cannot pause the contract

Name	Exist	Tested	Status
Deployer cannot pause	✓	✓	✗



## Overall checkup (Smart Contract Security)

Tested	Verified
✓	✓

### Legend

Attribute	Symbol
Verified / Checked	✓
Partly Verified	⚠
Unverified / Not checked	✗
Not available	—

# Modifiers and public functions

v1.0

```

  ✓ 🔹 initialize
    |   Ⓜ initializer
  ✓ 🔹 flashLoan
    |   Ⓜ nonReentrant
  ✓ 🔹 setFlashLoanFees
    |   Ⓜ onlyOwner

```

```

  ✓ 🔹 pause
    |   Ⓜ onlyOwner
  ✓ 🔹 unpause
    |   Ⓜ onlyOwner

```

```

  ✓ 🔹 initialize
    |   Ⓜ initializer
  ✓ 🔹 swap
    |   Ⓜ nonReentrant
    |   Ⓜ whenNotPaused
    |   Ⓜ deadlineCheck
  ✓ 🔹 addLiquidity
    |   Ⓜ nonReentrant
    |   Ⓜ whenNotPaused
    |   Ⓜ deadlineCheck
  ✓ 🔹 removeLiquidity
    |   Ⓜ nonReentrant
    |   Ⓜ deadlineCheck
  ✓ 🔹 removeLiquidityOneToken
    |   Ⓜ nonReentrant
    |   Ⓜ whenNotPaused
    |   Ⓜ deadlineCheck
  ✓ 🔹 removeLiquidityImbalance
    |   Ⓜ nonReentrant
    |   Ⓜ whenNotPaused
    |   Ⓜ deadlineCheck
  ✓ 🔹 withdrawAdminFees
    |   Ⓜ onlyOwner
  ✓ 🔹 setAdminFee
    |   Ⓜ onlyOwner
  ✓ 🔹 setSwapFee
    |   Ⓜ onlyOwner
  ✓ 🔹 rampA
    |   Ⓜ onlyOwner
  ✓ 🔹 stopRampA
    |   Ⓜ onlyOwner

```

```

  🔹 convert
  🔹 addLiquidity
  🔹 removeLiquidity
  🔹 removeBaseLiquidityOneToken
  🔹 swapFromBase
  🔹 swapToBase

```

```

  ✓ 🔹 initialize
    |   Ⓜ initializer
  ✓ 🔹 mint
    |   Ⓜ onlyOwner

```

```

  🔹 permit
















































```

**Please check if an OnlyOwner or similar restrictive modifier has been forgotten.**



# Source Units in Scope

## v1.0

Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
	contracts/SwapUtils.sol	1	————	1063	965	585	265	483	————
	contracts/interfaces/IERC20Upgradeable.sol	————	1	77	26	17	57	13	
	contracts/interfaces/IERC20PermitUpgradeable.sol	————	1	51	43	10	41	7	
	contracts/interfaces/ISwapFlashLoan.sol	————	1	14	8	4	1	5	————
	contracts/interfaces/ISwap.sol	————	1	89	9	4	4	35	————
	contracts/interfaces/IERC20.sol	————	1	77	26	17	57	13	
	contracts/interfaces/FlashLoanReceiver.sol	————	1	20	13	3	8	3	————
	contracts/helpers/GenericERC20.sol	1	————	38	38	16	18	14	————
	contracts/helpers/FlashLoanBorrowerExample.sol	1	————	66	55	40	8	62	
	contracts/Swap.sol	1	————	552	454	212	195	177	————
	contracts/Router.sol	1	————	319	241	209	4	315	
	contracts/LPToken.sol	1	————	61	53	24	24	22	————
	contracts/utlis/AddressUpgradeable.sol	1	————	165	149	67	100	42	
	contracts/libraries/ECDSAUpgradeable.sol	1	————	86	86	27	50	34	
	contracts/libraries/CountersUpgradeable.sol	1	————	40	40	17	17	2	
	contracts/libraries/SafeMathUpgradeable.sol	1	————	214	214	61	139	16	
	contracts/libraries/ERC20PermitUpgradeable.sol	1	————	87	87	45	28	35	
	contracts/libraries/ERC20BurnableUpgradeable.sol	1	————	51	51	22	22	24	
	contracts/libraries/ReentrancyGuardUpgradeable.sol	1	————	68	68	20	38	11	
	contracts/libraries/ContextUpgradeable.sol	1	————	32	32	17	12	7	
	contracts/libraries/PausableUpgradeable.sol	1	————	97	97	35	50	23	
	contracts/libraries/Address.sol	1	————	189	169	78	113	47	
	contracts/libraries/EIP712Upgradeable.sol	1	————	121	121	46	66	30	
	contracts/libraries/OwnerPausableUpgradeable.sol	1	————	37	37	19	13	17	————
	contracts/libraries/SafeMath.sol	1	————	214	214	61	139	16	
	contracts/libraries/Clones.sol	1	————	78	78	38	35	123	
	contracts/libraries/SafeERC20.sol	1	————	75	74	33	32	25	
	contracts/libraries/ERC20Upgradeable.sol	1	————	313	313	95	185	86	
	contracts/OwnableUpgradeable.sol	1	————	75	75	33	33	31	
	contracts/MathUtils.sol	1	————	39	39	14	20	3	————
	contracts/proxy/Initializable.sol	1	————	55	55	21	24	9	
	contracts/AmplificationUtils.sol	1	————	160	148	90	44	46	————
	contracts/SwapFlashLoan.sol	1	————	169	152	73	65	40	————
	<b>Totals</b>	<b>27</b>	<b>6</b>	<b>4792</b>	<b>4230</b>	<b>2053</b>	<b>1907</b>	<b>1816</b>	

## Legend

Attribute	Description
Lines	total lines of the source unit
nLines	normalized lines of the source unit (e.g. normalizes functions spanning multiple lines)
nSLOC	normalized source lines of code (only source-code lines; no comments, no blank lines)
Comment Lines	lines containing single or block comments

Complexity Score	a custom complexity score derived from code statements that are known to introduce code complexity (branches, loops, calls, external interfaces, ...)
------------------	---





# Audit Results

# AUDIT PASSED

## Critical issues

**No critical issues**

## High issues

**No high issues**

## Medium issues

**No medium issues**

## Low issues

Issue	File	Type	Line	Description
#1	Main	Contract doesn't import npm packages from source (like OpenZeppelin etc.)	-	We recommend to import all packages from npm directly without flatten the contract. Functions could be modified or can be susceptible to vulnerabilities

#2	Files	A floating pragma is set	At the top of source file	<p>The current pragma Solidity directive is „<code>&gt;=0.6.0 &lt;0.8.0</code>“.</p> <ul style="list-style-type: none"> <li>- OwnableUpgradeable</li> <li>- IERC20</li> <li>- IERC20PermitUpgradeable</li> <li>- IERC20Upgradeable</li> <li>- ISwap</li> <li>- Address</li> <li>- Clones</li> <li>- ContextUpgradeable</li> <li>- CountersUpgradeable</li> <li>- ECDSAUpgradeable</li> <li>- EIP712Upgradeable</li> <li>- ERC20BurnableUpgradeable</li> <li>- ERC20PermitUpgradeable</li> <li>- ERC20Upgradeable</li> <li>- PausableUpgradeable</li> <li>- ReentrancyGuardUpgradeable</li> <li>- SafeERC20</li> <li>- SafeMath</li> <li>- SafeMathUpgradeable</li> <li>- Initializable</li> <li>- AddressUpgradeable</li> </ul>
#3	LPToken	State variables shadowing	26	Rename the state variables that shadow another component
#4	SwapFlashLoan	Missing Events Arithmetic	166-167	Emit an event for critical parameter changes

#5	Files	Layout ordering	See description	<p>According to solidity documentation, the correct way to order is the following way:</p> <ul style="list-style-type: none"> <li>- types</li> <li>- receive</li> <li>- fallback</li> <li>- external</li> <li>- public</li> <li>- Internal</li> <li>- Private</li> </ul> <p>Keep it in mind, that the view and pure are grouping the external etc.</p> <p>Following files can be reordered by this layout:</p> <ul style="list-style-type: none"> <li>- SwapUtils</li> <li>- Swap</li> <li>- AmplificationUtils</li> </ul>
----	-------	-----------------	-----------------	--

## Informational issues

Issue	File	Type	Line	Description
#1	Router	Unused return values	96, 33, 43, 29, 156, 159, 185, 212,	Ensure that all the return values of the function calls are used and handle both success and failure cases if needed by the business logic
#2	SwapUtils	Misspelling	See description	<p>Change following words:</p> <ul style="list-style-type: none"> <li>- stableswap L205</li> </ul> <p>Make sure to change it everywhere else as well.</p>

#3	Main	NatSpec documentation missing	-	<p>If you started to comment your code, also comment all other functions, variables etc.</p> <p>Some NatSpec format is missing in following files:</p> <ul style="list-style-type: none"> <li>- Address</li> <li>- AddressUpgradeable</li> <li>- ContextUpgradeable</li> <li>- CounterUpgradeable</li> <li>- EIP712Upgradeable</li> <li>- SafeERC20</li> <li>- Router</li> <li>- Swap</li> </ul>
#4	ERC20PermitUpgradeable	Unused parameter	41	<p>Remove unused parameter. Only just remove the variable "name" and leave the rest</p>
#5	SwapUtils  Router  AddressUpgradeable	Inconsistent coding style in source files	Source files	<p>Some internal/private functions are leading with underscore and the others not like the following:</p> <ul style="list-style-type: none"> <li>- _calculateWithdrawOneToken L155</li> <li>- calculateWithdrawOneToken L191</li> <li>- .....</li> </ul> <p>And so on.</p> <p>If you are going to change every internal function leading with an underscore make sure to change it everywhere else too.</p>

#6	SwapUtils	Unclear variable names	-	<p>Variables were set with unclear names. This would be problematically if your are going to let other devs work on the contract because the unclear variables are not understandable.</p> <p>Examples:</p> <ul style="list-style-type: none"> <li>- _xp</li> <li>- getD</li> <li>- getY</li> <li>- .....</li> </ul> <p>And so on.</p>
----	-----------	------------------------	---	--

## Commented Code exist

There are some instances of code being commented out in the following files that should be removed:

File	Line	Comment
AmplificationUtils	73	// a0 + (a1 - a0) * (block.timestamp - t0) / (t1 - t0)
	79	// a0 - (a0 - a1) * (block.timestamp - t0) / (t1 - t0)
Swap	177-178	// swapStorage.initialATime = 0; // swapStorage.futureATime = 0;
SwapUtils	226-228	// if i == tokenIndex, dxExpected = xp[i] * d1 / d0 - newY // else dxExpected = xp[i] - (xp[i] * d1 / d0) // xpReduced[i] -= dxExpected * fee / FEE_DENOMINATOR

## Recommendation

Remove the commented code, or address them properly.

## Audit Comments

We recommend you to use the special form of comments (NatSpec Format, Follow link for more information <https://docs.soliditylang.org/en/v0.5.10/natspec-format.html>) for your contracts to provide rich documentation for functions, return variables and more. This helps investors to make clear what that variables, functions etc. do.

## 20. April 2022:

- Read whole report for more information

## SWC Attacks

ID	Title	Relationships	Status
<a href="#">SW C-1 36</a>	Unencrypted Private Data On-Chain	<a href="#">CWE-767: Access to Critical Private Variable via Public Method</a>	PASSED
<a href="#">SW C-1 35</a>	Code With No Effects	<a href="#">CWE-1164: Irrelevant Code</a>	PASSED
<a href="#">SW C-1 34</a>	Message call with hardcoded gas amount	<a href="#">CWE-655: Improper Initialization</a>	PASSED
<a href="#">SW C-1 33</a>	Hash Collisions With Multiple Variable Length Arguments	<a href="#">CWE-294: Authentication Bypass by Capture-replay</a>	PASSED
<a href="#">SW C-1 32</a>	Unexpected Ether balance	<a href="#">CWE-667: Improper Locking</a>	PASSED
<a href="#">SW C-1 31</a>	Presence of unused variables	<a href="#">CWE-1164: Irrelevant Code</a>	PASSED
<a href="#">SW C-1 30</a>	Right-To-Left-Override control character (U+202E)	<a href="#">CWE-451: User Interface (UI) Misrepresentation of Critical Information</a>	PASSED
<a href="#">SW C-1 29</a>	Typographical Error	<a href="#">CWE-480: Use of Incorrect Operator</a>	PASSED
<a href="#">SW C-1 28</a>	DoS With Block Gas Limit	<a href="#">CWE-400: Uncontrolled Resource Consumption</a>	PASSED

<a href="#">SW C-1 27</a>	Arbitrary Jump with Function Type Variable	<a href="#">CWE-695: Use of Low-Level Functionality</a>	<b>PASSED</b>
<a href="#">SW C-1 25</a>	Incorrect Inheritance Order	<a href="#">CWE-696: Incorrect Behavior Order</a>	<b>PASSED</b>
<a href="#">SW C-1 24</a>	Write to Arbitrary Storage Location	<a href="#">CWE-123: Write-what-where Condition</a>	<b>PASSED</b>
<a href="#">SW C-1 23</a>	Requirement Violation	<a href="#">CWE-573: Improper Following of Specification by Caller</a>	<b>PASSED</b>
<a href="#">SW C-1 22</a>	Lack of Proper Signature Verification	<a href="#">CWE-345: Insufficient Verification of Data Authenticity</a>	<b>PASSED</b>
<a href="#">SW C-1 21</a>	Missing Protection against Signature Replay Attacks	<a href="#">CWE-347: Improper Verification of Cryptographic Signature</a>	<b>PASSED</b>
<a href="#">SW C-1 20</a>	Weak Sources of Randomness from Chain Attributes	<a href="#">CWE-330: Use of Insufficiently Random Values</a>	<b>PASSED</b>
<a href="#">SW C-11 9</a>	Shadowing State Variables	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	<b>NOT PASSED</b>
<a href="#">SW C-11 8</a>	Incorrect Constructor Name	<a href="#">CWE-665: Improper Initialization</a>	<b>PASSED</b>
<a href="#">SW C-11 7</a>	Signature Malleability	<a href="#">CWE-347: Improper Verification of Cryptographic Signature</a>	<b>PASSED</b>

<a href="#">SW C-11 6</a>	Timestamp Dependence	<a href="#">CWE-829: Inclusion of Functionality from Untrusted Control Sphere</a>	<b>PASSED</b>
<a href="#">SW C-11 5</a>	Authorization through tx.origin	<a href="#">CWE-477: Use of Obsolete Function</a>	<b>PASSED</b>
<a href="#">SW C-11 4</a>	Transaction Order Dependence	<a href="#">CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')</a>	<b>PASSED</b>
<a href="#">SW C-11 3</a>	DoS with Failed Call	<a href="#">CWE-703: Improper Check or Handling of Exceptional Conditions</a>	<b>PASSED</b>
<a href="#">SW C-11 2</a>	Delegatecall to Untrusted Callee	<a href="#">CWE-829: Inclusion of Functionality from Untrusted Control Sphere</a>	<b>PASSED</b>
<a href="#">SW C-11 1</a>	Use of Deprecated Solidity Functions	<a href="#">CWE-477: Use of Obsolete Function</a>	<b>PASSED</b>
<a href="#">SW C-11 0</a>	Assert Violation	<a href="#">CWE-670: Always-Incorrect Control Flow Implementation</a>	<b>PASSED</b>
<a href="#">SW C-1 09</a>	Uninitialized Storage Pointer	<a href="#">CWE-824: Access of Uninitialized Pointer</a>	<b>PASSED</b>
<a href="#">SW C-1 08</a>	State Variable Default Visibility	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	<b>PASSED</b>
<a href="#">SW C-1 07</a>	Reentrancy	<a href="#">CWE-841: Improper Enforcement of Behavioral Workflow</a>	<b>PASSED</b>
<a href="#">SW C-1 06</a>	Unprotected SELFDESTRUCT Instruction	<a href="#">CWE-284: Improper Access Control</a>	<b>PASSED</b>



<a href="#">SW</a> <a href="#">C-1</a> <a href="#">05</a>	Unprotected Ether Withdrawal	<a href="#">CWE-284: Improper Access Control</a>	<b>PASSED</b>
<a href="#">SW</a> <a href="#">C-1</a> <a href="#">04</a>	Unchecked Call Return Value	<a href="#">CWE-252: Unchecked Return Value</a>	<b>PASSED</b>
<a href="#">SW</a> <a href="#">C-1</a> <a href="#">03</a>	Floating Pragma	<a href="#">CWE-664: Improper Control of a Resource Through its Lifetime</a>	<b>NOT PASSED</b>
<a href="#">SW</a> <a href="#">C-1</a> <a href="#">02</a>	Outdated Compiler Version	<a href="#">CWE-937: Using Components with Known Vulnerabilities</a>	<b>PASSED</b>
<a href="#">SW</a> <a href="#">C-1</a> <a href="#">01</a>	Integer Overflow and Underflow	<a href="#">CWE-682: Incorrect Calculation</a>	<b>PASSED</b>
<a href="#">SW</a> <a href="#">C-1</a> <a href="#">00</a>	Function Default Visibility	<a href="#">CWE-710: Improper Adherence to Coding Standards</a>	<b>PASSED</b>

The logo features the word "SolidProofed" in a white, handwritten-style script. The "P" is particularly large and stylized, with a long horizontal stroke that extends to the left. The background is a solid blue color with a faint, large shield emblem. The shield has a grid-like pattern on its right side and a solid blue area on its left side.

SolidProofed

**Blockchain Security | Smart Contract Audits | KYC**

A small horizontal bar representing the German flag, with black, red, and gold stripes.

MADE IN GERMANY