



ARK

Smart Contract Audit Report

TABLE OF CONTENTS

| Audited Details

- Audited Project
- Blockchain
- Addresses
- Project Website
- Codebase

| Summary

- Contract Summary
- Audit Findings Summary
- Vulnerabilities Summary

| Conclusion

| Audit Results

| Smart Contract Analysis

- Detected Vulnerabilities

| Disclaimer

| About Us

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
ARK	ARKFI	Binance Smart Chain

Addresses

Contract address	0x111120a4cfacf4c78e0d6729274fd5a5ae2b1111
Contract deployer address	0x810C5704d0b10254d36440c773402180ed973dd0

Project Website

<https://www.arkfi.io/>

Codebase

<https://bscscan.com/address/0x111120a4cfacf4c78e0d6729274fd5a5ae2b1111#code>

SUMMARY

Ark Fi products and smart contracts are built on the BSC layer one blockchain for its' security, decentralization, low gas fees, and compatibility with Ethereum Virtual Machine (EVM). The native token BNB and stable coin BUSD are available on the Binance CEX and several other major exchanges worldwide. The BSC has made a name for itself in the DeFi space as a reliable, low-cost, and secure solution for decentralized applications. By launching on BSC, Ark Fi taps into a large community of users and liquidity that will speed the adoption and early growth of the protocol.

Contract Summary

Documentation Quality

ARK provides a very good documentation with standard of solidity base code.

- The technical description is provided clearly and structured and also dont have any high risk issue.

Code Quality

The Overall quality of the basecode is standard.

- Standard solidity basecode and rules are already followed by ARK with the discovery of several low issues.

Test Coverage

Test coverage of the project is 100% (Through Codebase)

Audit Findings Summary

- SWC-101 | It is recommended to use vetted safe math libraries for arithmetic operations consistently on lines 48, 48, 123, 131, 143, 170, 170, 172, 172, 173, 173, 173, 173, 186, 187, 193, 194, 195, 200, 201, 202 and 264.

CONCLUSION

We have audited the ARK project released on December 2022 to discover issues and identify potential security vulnerabilities in ARK Project. This process is used to find technical issues and security loopholes which might be found in the smart contract.

The security audit report provides satisfactory results with low-risk issues.

The issues found in the ARK smart contract code do not pose a considerable risk. The writing of the contract is close to the standard of writing contracts in general. The low-risk issues found are some arithmetic operation issues. It is recommended to use vetted safe math libraries for arithmetic operations consistently throughout the smart contract system.

AUDIT RESULT

Article	Category	Description	Result
Default Visibility	SWC-100 SWC-108	Functions and state variables visibility should be set explicitly. Visibility levels should be specified consciously.	PASS
Integer Overflow and Underflow	SWC-101	If unchecked math is used, all math operations should be safe from overflows and underflows.	ISSUE FOUND
Outdated Compiler Version	SWC-102	It is recommended to use a recent version of the Solidity compiler.	PASS
Floating Pragma	SWC-103	Contracts should be deployed with the same compiler version and flags that they have been tested thoroughly.	PASS
Unchecked Call Return Value	SWC-104	The return value of a message call should be checked.	PASS
Unprotected Ether Withdrawal	SWC-105	Due to missing or insufficient access controls, malicious parties can withdraw from the contract.	PASS
SELFDESTRUCT Instruction	SWC-106	The contract should not be self-destructible while it has funds belonging to users.	PASS
Reentrancy	SWC-107	Check effect interaction pattern should be followed if the code performs recursive call.	PASS
Uninitialized Storage Pointer	SWC-109	Uninitialized local storage variables can point to unexpected storage locations in the contract.	PASS
Assert Violation	SWC-110 SWC-123	Properly functioning code should never reach a failing assert statement.	PASS
Deprecated Solidity Functions	SWC-111	Deprecated built-in functions should never be used.	PASS
Delegate call to Untrusted Callee	SWC-112	Delegatecalls should only be allowed to trusted addresses.	PASS

DoS (Denial of Service)	SWC-113 SWC-128	Execution of the code should never be blocked by a specific contract state unless required.	PASS
Race Conditions	SWC-114	Race Conditions and Transactions Order Dependency should not be possible.	PASS
Authorization through tx.origin	SWC-115	tx.origin should not be used for authorization.	PASS
Block values as a proxy for time	SWC-116	Block numbers should not be used for time calculations.	PASS
Signature Unique ID	SWC-117 SWC-121 SWC-122	Signed messages should always have a unique id. A transaction hash should not be used as a unique id.	PASS
Incorrect Constructor Name	SWC-118	Constructors are special functions that are called only once during the contract creation.	PASS
Shadowing State Variable	SWC-119	State variables should not be shadowed.	PASS
Weak Sources of Randomness	SWC-120	Random values should never be generated from Chain Attributes or be predictable.	PASS
Write to Arbitrary Storage Location	SWC-124	The contract is responsible for ensuring that only authorized user or contract accounts may write to sensitive storage locations.	PASS
Incorrect Inheritance Order	SWC-125	When inheriting multiple contracts, especially if they have identical functions, a developer should carefully specify inheritance in the correct order. The rule of thumb is to inherit contracts from more /general/ to more /specific/.	PASS
Insufficient Gas Griefing	SWC-126	Insufficient gas grieving attacks can be performed on contracts which accept data and use it in a sub-call on another contract.	PASS
Arbitrary Jump Function	SWC-127	As Solidity doesnt support pointer arithmetics, it is impossible to change such variable to an arbitrary value.	PASS

Typographical Error	SWC-129	A typographical error can occur for example when the intent of a defined operation is to sum a number to a variable.	PASS
Override control character	SWC-130	Malicious actors can use the Right-To-Left-Override unicode character to force RTL text rendering and confuse users as to the real intent of a contract.	PASS
Unused variables	SWC-131 SWC-135	Unused variables are allowed in Solidity and they do not pose a direct security issue.	PASS
Unexpected Ether balance	SWC-132	Contracts can behave erroneously when they strictly assume a specific Ether balance.	PASS
Hash Collisions Variable	SWC-133	Using abi.encodePacked() with multiple variable length arguments can, in certain situations, lead to a hash collision.	PASS
Hardcoded gas amount	SWC-134	The transfer() and send() functions forward a fixed amount of 2300 gas.	PASS
Unencrypted Private Data	SWC-136	It is a common misconception that private type variables cannot be read.	PASS

SMART CONTRACT ANALYSIS

Started	Friday Dec 16 2022 12:26:51 GMT+0000 (Coordinated Universal Time)
Finished	Saturday Dec 17 2022 11:06:40 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	ARK_TOKEN.sol

Detected Issues

ID	Title	Severity	Status
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "**" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-=" DISCOVERED	low	acknowledged

SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 48

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
47  uint8 private constant _decimals = 18;
48  uint256 private _totalSupply = 800_000 * (10**_decimals);
49
50  mapping(address => uint256) private _balances;
51  mapping(address => mapping(address => uint256)) private _allowances;
52
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 48

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
47  uint8 private constant _decimals = 18;
48  uint256 private _totalSupply = 800_000 * (10**_decimals);
49
50  mapping(address => uint256) private _balances;
51  mapping(address => mapping(address => uint256)) private _allowances;
52
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 123

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
122     require(spender != address(0), "Can't use zero address here");
123     _allowances[msg.sender][spender] = allowance(msg.sender, spender) + addedValue;
124     emit Approval(msg.sender, spender, _allowances[msg.sender][spender]);
125     return true;
126 }
127
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 131

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
130   require(allowance(msg.sender, spender) >= subtractedValue, "Can't subtract more
    than current allowance");
131   _allowances[msg.sender][spender] = allowance(msg.sender, spender) -
    subtractedValue;
132   emit Approval(msg.sender, spender, _allowances[msg.sender][spender]);
133   return true;
134   }
135
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 143

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
142     require(_allowances[sender][msg.sender] >= amount, "Insufficient Allowance");
143     _allowances[sender][msg.sender] -= amount;
144     emit Approval(sender, msg.sender, _allowances[sender][msg.sender]);
145 }
146
147
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 170

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
169     function takeTax(address sender, uint256 amount) internal returns (uint256){
170         _transfer(sender, pair, amount * liqTax / 100);
171         IDEXPair(pair).sync();
172         _transfer(sender, vault, amount * vaultTax / 100);
173         return amount * (100 - vaultTax - liqTax) / 100;
174     }
```


SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 170

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
169     function takeTax(address sender, uint256 amount) internal returns (uint256){
170         _transfer(sender, pair, amount * liqTax / 100);
171         IDEXPair(pair).sync();
172         _transfer(sender, vault, amount * vaultTax / 100);
173         return amount * (100 - vaultTax - liqTax) / 100;
174     }
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 172

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
171     IDEXPair(pair).sync();
172     _transfer(sender, vault, amount * vaultTax / 100);
173     return amount * (100 - vaultTax - liqTax) / 100;
174 }
175
176
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 172

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
171     IDEXPair(pair).sync();
172     _transfer(sender, vault, amount * vaultTax / 100);
173     return amount * (100 - vaultTax - liqTax) / 100;
174 }
175
176
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 173

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
172  _transfer(sender, vault, amount * vaultTax / 100);
173  return amount * (100 - vaultTax - liqTax) / 100;
174  }
175
176  function checkSwap(address sender, address recipient) internal view returns (bool)
177  {
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 173

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
172  _transfer(sender, vault, amount * vaultTax / 100);  
173  return amount * (100 - vaultTax - liqTax) / 100;  
174  }  
175  
176  function checkSwap(address sender, address recipient) internal view returns (bool)  
177  {  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 173

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
172  _transfer(sender, vault, amount * vaultTax / 100);
173  return amount * (100 - vaultTax - liqTax) / 100;
174  }
175
176  function checkSwap(address sender, address recipient) internal view returns (bool)
177  {
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 173

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
172  _transfer(sender, vault, amount * vaultTax / 100);
173  return amount * (100 - vaultTax - liqTax) / 100;
174  }
175
176  function checkSwap(address sender, address recipient) internal view returns (bool)
177  {
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 186

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
185     require(amount <= _balances[sender], "Can't transfer more than you own");
186     _balances[sender] -= amount;
187     _balances[recipient] += amount;
188     emit Transfer(sender, recipient, amount);
189     return true;
190
```


SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 187

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
186  _balances[sender] -= amount;
187  _balances[recipient] += amount;
188  emit Transfer(sender, recipient, amount);
189  return true;
190  }
191
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 193

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
192 function _swapTopUp(uint256 amountNeeded) internal {  
193     uint256 howMuchIsMissing = amountNeeded - _balances[swap];  
194     _balances[swap] += howMuchIsMissing;  
195     _totalSupply += howMuchIsMissing;  
196     emit SwapToppedUp(howMuchIsMissing);  
197 }
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 194

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
193     uint256 howMuchIsMissing = amountNeeded - _balances[swap];
194     _balances[swap] += howMuchIsMissing;
195     _totalSupply += howMuchIsMissing;
196     emit SwapToppedUp(howMuchIsMissing);
197 }
198
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 195

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
194  _balances[swap] += howMuchIsMissing;  
195  _totalSupply += howMuchIsMissing;  
196  emit SwapToppedUp(howMuchIsMissing);  
197  }  
198  
199
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 200

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
199  function _vaultTopUp(address vaultAddress, uint256 amountNeeded) internal {
200  uint256 howMuchIsMissing = amountNeeded - _balances[vaultAddress];
201  _balances[vaultAddress] += howMuchIsMissing;
202  _totalSupply += howMuchIsMissing;
203  emit VaultToppedUp(vaultAddress, howMuchIsMissing);
204
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 201

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
200     uint256 howMuchIsMissing = amountNeeded - _balances[vaultAddress];
201     _balances[vaultAddress] += howMuchIsMissing;
202     _totalSupply += howMuchIsMissing;
203     emit VaultToppedUp(vaultAddress, howMuchIsMissing);
204 }
205
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 202

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
201  _balances[vaultAddress] += howMuchIsMissing;
202  _totalSupply += howMuchIsMissing;
203  emit VaultToppedUp(vaultAddress, howMuchIsMissing);
204  }
205
206
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 264

low SEVERITY

This plugin produces issues to support false positive discovery within mythril.

Source File

- ARK_TOKEN.sol

Locations

```
263     vaultTax = newVaultTax;  
264     require(liqTax + vaultTax <= 100, "Taxes can't exceed 100%");  
265     emit TaxesChanged(liqTax, vaultTax);  
266     }  
267  
268
```


DISCLAIMER

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to, or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without Sysfixed's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Sysfixed to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model, or legal compliance.

This is a limited report on our findings based on our analysis, in accordance with good industry practice as of the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing 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.

This report is provided for information purposes only and on a non-reliance basis and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Sysfixed and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers, and other representatives) (Sysfixed) owe no duty of care.

ABOUT US

Sysfixed is a blockchain security certification organization established in 2021 with the objective to provide smart contract security services and verify their correctness in blockchain-based protocols. Sysfixed automatically scans for security vulnerabilities in Ethereum and other EVM-based blockchain smart contracts. Sysfixed a comprehensive range of analysis techniques—including static analysis, dynamic analysis, and symbolic execution—can accurately detect security vulnerabilities to provide an in-depth analysis report. With a vibrant ecosystem of world-class integration partners that amplify developer productivity, Sysfixed can be utilized in all phases of your project's lifecycle. Our team of security experts is dedicated to the research and improvement of our tools and techniques used to fortify your code.