CONTRACT WOLF

Blockchain Security - Smart Contract Audits

Security Assessment

March 6, 2022

Disclaimer 3

Scope of Work & Engagement 3

Project Description 4

Risk Level Classification 5

Methodology 6

Overall Checkup 10

Verify Claim 11

SWC Attacks 11

Code Validation 13

Audit Result 17

Function Comments 18

Disclaimer

ContractWolf.io audits and reports should not be considered as a form of project's "advertisement" and does not cover any interaction and assessment from "project's contract" to "external contracts" such as Pancakeswap or similar.

ContractWolf does not provide any warranty on its released reports.

ContractWolf should not be used as a decision to invest into an audited project and is not affiliated nor partners to its audited contract projects.

ContractWolf provides transparent report to all its "clients" and to its "clients participants" and will not claim any guarantee of bug-free code within its **SMART CONTRACT**.

ContractWolf presence is to analyze, audit and assess the client's smart contract's code.

Each company or projects should be liable to its security flaws and functionalities.

Scope of Work

BAPT SWAP team agreed and provided us with the files that needs to be tested (Github, Bscscan, Etherscan, files, etc.). The scope of the audit is the main contract.

The goal of this engagement was to identify if there is a possibility of security flaws in the implementation of the contract or system.

ContractWolf will be focusing on contract issues and functionalities along with the projects claims from smart contract to their website, whitepaper and repository which has been provided by **BAPT SWAP**.

Description

BAPT SWAP is a pioneering organization focused on providing safe and secure methods for individuals and businesses to develop innovative projects in the cryptocurrency space through facilitating utilities like BAPTSWAP (DEX) and Wolves of Aptos (NFTs)

Risk Level Classification

Risk Level represents the classification or the probability that a certain function or threat that can exploit vulnerability and have an impact within the system or contract.
Risk Level is computed based on CVSS Version 3.0

Level	Value	Vulnerability
Critical	9 - 10	An Exposure that can affect the contract functions in several events that can risk and disrupt the contract
High	7 - 8.9	An Exposure that can affect the outcome when using the contract that can serve as an opening in manipulating the contract in an unwanted manner
Medium	4 - 6.9	An opening that could affect the outcome in executing the contract in a specific situation
Low	0.1 - 3.9	An opening but doesn't have an impact on the functionality of the contract
Informational	0	An opening that consists of information's but will not risk or affect the contract

Auditing Approach

Every line of code along with its functionalities will undergo manual review to check its security issues, quality, and contract scope of inheritance. The manual review will be done by our team that will document any issues that there were discovered.

Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:

    - Review of the specifications, sources, and instructions provided to ContractWolf to make sure we understand the size, scope, and functionality of the smart contract.
    - Manual review of code, our team will have a process of reading the code line-by-line with the intention of identifying potential vulnerabilities and security flaws.

2. Testing and automated analysis that includes:

    - Testing the smart contract functions with common test cases and scenarios, to ensure that it returns the expected results.

3. Best practices review, the team will review the contract with the aim to improve efficiency, effectiveness, clarifications, maintainability, security, and control within the smart contract.

4. Recommendations to help the project take steps to secure the smart contract.

Description

Symbol 1: BaptSwap-APT-BAPT-LP

Symbol 2: BaptSwap-APT-USDC-LP

Capabilities

Resources

- SwapInfo
- Account
- PackageRegistry
- AptosCoin
- PairEventHolder (USDC)
- TokenPairReserve (USDC)
- TokenPairMetadata (USDC)
- PairEventHolder (BaptLabs)
- TokenPairReserve (BaptLabs)
- TokenPairMetadata (BaptLabs)
- TokenPairRewardsPool (BaptLabs)
- LPToken (USDC)
- LPToken (USDC)
- LPToken (BaptLabs)
- LPToken (BaptLabs)

Correct implementation of Token Standard

Tested	Verified
✓	✓

Overall Checkup (Smart Contract Security)

Tested	Verified
✓	✓

Function	Description	Exist	Tested	Verified
TotalSupply	Information about the total coin or token supply	✗	✗	✗
BalanceOf	Details on the account balance from a specified address	✓	✓	✓
Transfer	An action that transfers a specified amount of coin or token to a specified address	✓	✓	✓
TransferFrom	An action that transfers a specified amount of coin or token from a specified address	✓	✓	✓
Approve	Provides permission to withdraw specified number of coin or token from a specified address	✓	✓	✓

Verify Claims

Statement	Exist	Tested	Deployer
Renounce Ownership	–	–	–
Mint	✓	✓	✓
Burn	✓	✓	✓
Block	–	–	–
Pause	–	–	–

Legend

Attribute	Symbol
Verified / Can	✓
Verified / Cannot	✗
Unverified / Not checked	⚑
Not Available	–

Modules

Name:	math
Name:	swap
Name:	u256
Name:	router
Name:	swap_utils

SWC Attacks

ID	Title	Status
SWC-136	Unencrypted Private Data On-Chain	PASSED
SWC-135	Code With No Effects	PASSED
SWC-134	Message call with hardcoded gas amount	PASSED
SWC-133	Hash Collisions with Multiple Variable Length Arguments	PASSED
SWC-132	Unexpected Ether balance	PASSED
SWC-131	Presence of unused variables	PASSED
SWC-130	Right-To Left Override control character (U+202E)	PASSED
SWC-129	Typographical Error	PASSED
SWC-128	DoS With Block Gas Limit	PASSED
SWC-127	Arbitrary Jump with Function Type Variable	PASSED
SWC-126	Insufficient Gas Griefing	PASSED
SWC-125	Incorrect Inheritance Order	PASSED
SWC-124	Write to Arbitrary Storage Location	PASSED
SWC-123	Requirement Violation	PASSED
SWC-122	Lack of Proper Signature Verification	PASSED
SWC-121	Missing Protection against Signature Replay Attacks	PASSED
SWC-120	Weak Sources of Randomness from Chain Attributes	PASSED
SWC-119	Shadowing State Variables	PASSED
SWC-118	Incorrect Constructor Name	PASSED
SWC-117	Signature Malleability	PASSED
SWC-116	Block values as a proxy for time	PASSED
SWC-115	Authorization through tx.origin	PASSED
SWC-114	Transaction Order Dependence	PASSED
SWC-113	DoS with Failed Call	PASSED
SWC-112	Delegate call to Untrusted Callee	PASSED
SWC-111	Use of Deprecated Solidity Functions	PASSED

SWC-110	Assert Violation	**PASSED**
SWC-109	Uninitialized Storage Pointer	**PASSED**
SWC-108	State Variable Default Visibility	**PASSED**
SWC-107	Reentrancy	**PASSED**
SWC-106	Unprotected SELFDESTRUCT Instruction	**PASSED**
SWC-105	Unprotected Ether Withdrawal	**PASSED**
SWC-104	Unchecked Call Return Value	**PASSED**
SWC-103	Floating Pragma	**PASSED**
SWC-102	Outdated Compiler Version	**PASSED**
SWC-101	Integer Overflow and Underflow	**PASSED**
SWC-100	Function Default Visibility	**PASSED**

Code Validation through bytecode

swapvalid

```
0949  06  MOD
094A  72  PUSH19 0x6f7574657204737761700761636f756e7404
095E  63  PUSH4 0x6f646504
0963  63  PUSH4 0x6f696e05
0968  65  PUSH6 0x76656e74066f
096F  70  PUSH17 0x74696f6e107265736f757263655f616363
0981  6F  PUSH16 0x756e74067369676e657206737472696e
0992  67  PUSH8 0x0974696d65737461
099B  6D  PUSH14 0x7009747970655f696e666f046d61
09AA  74  PUSH21 0x680a737761705f7574696c73047532353611416464
09C0  4C  4C
09C1  69  PUSH10 0x71756964697479457665
09CC  6E  PUSH15 0x740e4665654368616e67654576656e
09DC  74  PUSH21 0x074c50546f6b656e1050616972437265617465644 5
09F2  76  PUSH23 0x656e740f506169724576656e74486f6c6465721452656d
0A0A  6F  PUSH16 0x76654c697175696469747945766 56e74
0A1B  13  SGT
0A1C  52  MSTORE
0A1D  65  PUSH6 0x776172647350
0A24  6F  PUSH16 0x6f6c55736572496e666f095377617045
0A35  76  PUSH23 0x656e740853776170496e666f11546f6b656e506169724d
0A4D  65  PUSH6 0x746164617461
0A54  10  LT
0A55  54  SLOAD
0A56  6F  PUSH16 0x6b656e506169725265736572766514 54
0A67  6F  PUSH16 0x6b656e5061697252657761726473506f
0A78  6F  PUSH16 0x6c0d6164645f6c697175696469747904
0A89  43  NUMBER
0A8A  6F  PUSH16 0x696e146164645f6c6971756964697479
0A9B  5F  5F
0A9C  64  PUSH5 0x6972656374
0AA2  0E  0E
0AA3  61  PUSH2 0x6464
0AA6  5F  5F
0AA7  73  PUSH20 0x7761705f6576656e741b6164645f737761705f65
0ABC  76  PUSH23 0x656e745f776974685f6164647265737305616 46d696e04
0AD4  62  PUSH3 0x75726e
0AD8  17  OR
0AD9  63  PUSH4 0x616c5f61
0ADE  63  PUSH4 0x635f746f
0AE3  6B  PUSH12 0x656e5f7065725f7368617265
0AF0  12  SLT
0AF1  63  PUSH4 0x616c5f70
0AF6  65  PUSH6 0x6e64696e675f
0AFD  72  PUSH19 0x65776172641c636865636b5f6f725f72656769
0B11  73  PUSH20 0x7465725f636f696e5f73746f72650d636c61696d
0B26  5F  5F
0B27  72  PUSH19 0x6577617264730b6372656174655f7061697209
0B3B  64  PUSH5 0x65706f7369
0B41  74  PUSH21 0x5f78096465706f7369745f7909657874726163745f
0B57  78  PUSH25 0x09657874726163745f79066665655f746f0b696e69745f6d6f
0B71  64  PUSH5 0x756c651169
0B77  6E  PUSH15 0x69745f726577617264735f706f6f6c
0B87  0F  0F
0B88  69  PUSH10 0x735f706169725f637265
0B93  61  PUSH2 0x7465
0B96  64  PUSH5 0x0f69735f70
0B9C  6F  PUSH16 0x6f6c5f637265617465640a6c705f6261
0BAD  6C  PUSH13 0x616e6365046d696e740e4d696e
0BBB  74  PUSH21 0x43617061626 96c697479076d696e745f6c700a6d69
0BD1  6E  PUSH15 0x745f6c705f746f0b72656769737465
0BE1  72  PUSH19 0x5f6c701072656d6f76655f6c69717569646974
0BF5  79  PUSH26 0x1772656d6f76655f6c6971756964697479 5f6469726563740b72
0C10  65  PUSH6 0x776172645f64
0C17  65  PUSH6 0x627409736574
0C1E  5F  5F
```

Matched

swaputilsvalid

```
00DF    73    PUSH20 0x7761705f7574696c730a636f6d70617261746f72
00F4    06    MOD
00F5    73    PUSH20 0x7472696e6709747970655f696e666f0e636f6d70
010A    61    PUSH2 0x7265
010D    5F    5F
010E    73    PUSH20 0x74727563740d6765745f616d6f756e745f696e0e
0123    67    PUSH8 0x65745f616d6f756e
012C    74    PUSH21 0x5f6f75740e6765745f657175616c5f656e756d1067
0142    65    PUSH6 0x745f67726561
0149    74    PUSH21 0x65725f656e756d106765745f736d616c6c65725f65
015F    6E    PUSH15 0x756d0e6765745f746f6b656e5f696e
016F    66    PUSH7 0x6f0571756f7465
0177    0F    0F
0178    73    PUSH20 0x6f72745f746f6b656e5f7479706506526573756c
018D    74    PUSH21 0x11636f6d706172655f75385f766563746f720f6973
01A3    5F    5F
01A4    67    PUSH8 0x7265617465725f74
01AD    68    PUSH9 0x616e0869735f657175
01B7    61    PUSH2 0x6c06
01BA    53    MSTORE8
01BB    74    PUSH21 0x72696e6709747970655f6e616d650562797465732a
01D1    D8    D8
01D2    F7    F7
01D3    E6    E6
01D4    4C    4C
01D5    7B    PUSH28 0xffcfe94d7dea84c79380942c30e13f1b12c7a89e98df91d0599b0000
01F2    00    *STOP
01F3    00    *STOP
01F4    00    *STOP
01F5    00    *STOP
```

Matched

mathvalid

```
label_0000:
    // Inputs[6]
    // {
    //     @0000  stack[-1]
    //     @0000  stack[-3]
    //     @0000  memory[stack[-1]:stack[-1] + stack[-2]]
    //     @0000  stack[-2]
    //     @0001  stack[-4]
    //     @0001  stack[-5]
    // }
    0000    A1  LOG1
    0001    1C  SHR
    0002    EB  EB
    // Stack delta = -4
    // Outputs[2]
    // {
    //     @0000  log(memory[stack[-1]:stack[-1] + stack[-2]], [stack[-3]]);
    //     @0001  stack[-5] = stack[-5] >> stack[-4]
    // }
    // Block terminates

    0003    0B    SIGNEXTEND
    0004    05    SDIV
    0005    00    *STOP
    0006    00    *STOP
    0007    00    *STOP
    0008    06    MOD
    0009    01    ADD
    000A    00    *STOP
    000B    02    MUL
    000C    03    SUB
    000D    02    MUL
    000E    19    NOT
    000F    05    SDIV
    0010    1B    SHL
    0011    12    SLT
    0012    07    SMOD
    0013    2D    2D
    0014    1E    1E
    0015    08    ADDMOD
    0016    4B    4B
    0017    20    SHA3
    0018    0C    0C
    0019    6B    PUSH12 0xd4020000000100010000202
    0026    03    SUB
    0027    00    *STOP
    0028    00    *STOP
    0029    03    SUB
    002A    00    *STOP
    002B    01    ADD
    002C    00    *STOP
    002D    00    *STOP
    002E    04    DIV
    002F    04    DIV
    0030    01    ADD
    0031    00    *STOP
    0032    00    *STOP
    0033    05    SDIV
    0034    01    ADD
    0035    01    ADD
    0036    00    *STOP
    0037    02    MUL
    0038    04    DIV
    0039    04    DIV
    003A    01    ADD
```

Matched

routervalid

```
02D9    77    PUSH24 0x61700a737761705f7574696c730d6164645f6c6971756964
02F2    69    PUSH10 0x7479176164645f737761
02FD    70    PUSH17 0x5f6576656e745f696e7465726e616c2461
030F    64    PUSH5 0x645f737761
0315    70    PUSH17 0x5f6576656e745f776974685f6164647265
0327    73    PUSH20 0x735f696e7465726e616c17636c61696d5f726577
033C    61    PUSH2 0x7264
033F    73    PUSH20 0x5f66726f6d5f706f6f6c0b6372656174655f7061
0354    69    PUSH10 0x72136372656174655f72
035F    65    PUSH6 0x77617264735f
0366    70    PUSH17 0x6f6f6c0d6765745f616d6f756e745f696e
0378    16    AND
0379    67    PUSH8 0x65745f616d6f756e
0382    74    PUSH21 0x5f696e5f696e7465726e616c04436f696e17676574
0398    5F    5F
0399    69    PUSH10 0x6e7465726d6564696174
03A4    65    PUSH6 0x5f6f75747075
03AB    74    PUSH21 0x246765745f696e7465726d6564696174655f6f7574
03C1    70    PUSH17 0x75745f785f746f5f65786163745f791869
03D3    73    PUSH20 0x5f706169725f6372656174656 45f696e7465726e
03E8    61    PUSH2 0x6c0b
03EB    72    PUSH19 0x656769737465725f6c700e7265676973746572
03FF    5F    5F
0400    74    PUSH21 0x6f6b656e1072656d6f76655f6c6971756964697479
0416    14    EQ
0417    73    PUSH20 0x74616b655f746f6b656e735f696e5f706f6f6c10
042C    73    PUSH20 0x7761705f65786163745f696e7075741173776170
0441    5F    5F
0442    65    PUSH6 0x786163745f6f
0449    75    PUSH22 0x747075742173776170 5f65786163745f785f746f5f79
0460    5F    5F
0461    64    PUSH5 0x6972656374
0467    5F    5F
0468    65    PUSH6 0x787465726e61
046F    6C    PUSH13 0x21737761705f785f746f5f6578
047D    61    PUSH2 0x6374
0480    5F    5F
0481    79    PUSH26 0x5f6469726563745f65787465726e616c1977697468647261775f
049C    74    PUSH21 0x6f6b656e735f66726f6d5f706f6f6c0f69735f7061
04B2    69    PUSH10 0x725f637265617465640f
04BD    73    PUSH20 0x6f72745f746f6b656e5f747970650a6164647265
04D2    73    PUSH20 0x735f6f661b6164645f737761705f6576656e745f
04E7    77    PUSH24 0x6974685f616464726573730f69735f706f6f6c5f63726561
0500    74    PUSH21 0x65640d636c61696d5f726577617264731169 6e6974
0516    5F    5F
0517    72    PUSH19 0x6577617264735f706f6f6c0e746f6b656e5f72
052B    65    PUSH6 0x736572766573
0532    0A    EXP
0533    74    PUSH21 0x6f6b656e5f666565731873776170 5f65786163745f
0549    78    PUSH25 0x5f746f5f795f6469726563740c64657374726f795f7a65726f
0563    18    XOR
0564    73    PUSH20 0x7761705f65786163745f795f746f5f785f646972
0579    65    PUSH6 0x637418737761
0580    70    PUSH17 0x5f785f746f5f65786163745f795f646972
0592    65    PUSH6 0x637418737761
0599    70    PUSH17 0x5f795f746f5f65786163745f785f646972
05AB    65    PUSH6 0x637408726567
05B2    69    PUSH10 0x737465720c7374616b65
05BD    5F    5F
05BE    74    PUSH21 0x6f6b656e7311737761705f65786163745f785f746f
05D4    5F    5F
05D5    79    PUSH26 0x11737761705f65786163745f795f746f5f7811737761705f785f
05F0    74    PUSH21 0x6f5f65786163745f7911737761705f795f746f5f65
0606    78    PUSH25 0x6163745f780576616c756507657874726163740f7769746864
0620    72    PUSH19 0x61775f746f6b656e732ad8f7e64c7bffcfe94d
0634    7D    PUSH30 0xea84c79380942c30e13f1b12c7a89e98df91d0599b000000000000000000
```

Matched

Audit Result

AUDIT PASSED

Critical Issues

No critical issues found

High Issues

No high issues found

Medium Issues

No medium issues found

Low Issues

No low issues found

Informational Issues

No informational issues found

Function Issues

No informational issues found

Function Comments

- Ownership cannot be transferred and renounced
- Contract has indefinite amount of fees
- Contract has a minting function
- Contract has a burning function
- Owner cannot set max transaction
- Contract is not pausable
- Owner cannot block users
- Contract has no antibot
- Owner can withdraw Team Fee
- Owner can set token pair
- Owner can set token fees
- Owner can set a new admin
- Owner can set treasury fees

CONTRACTWOLF

Blockchain Security - Smart Contract Audits