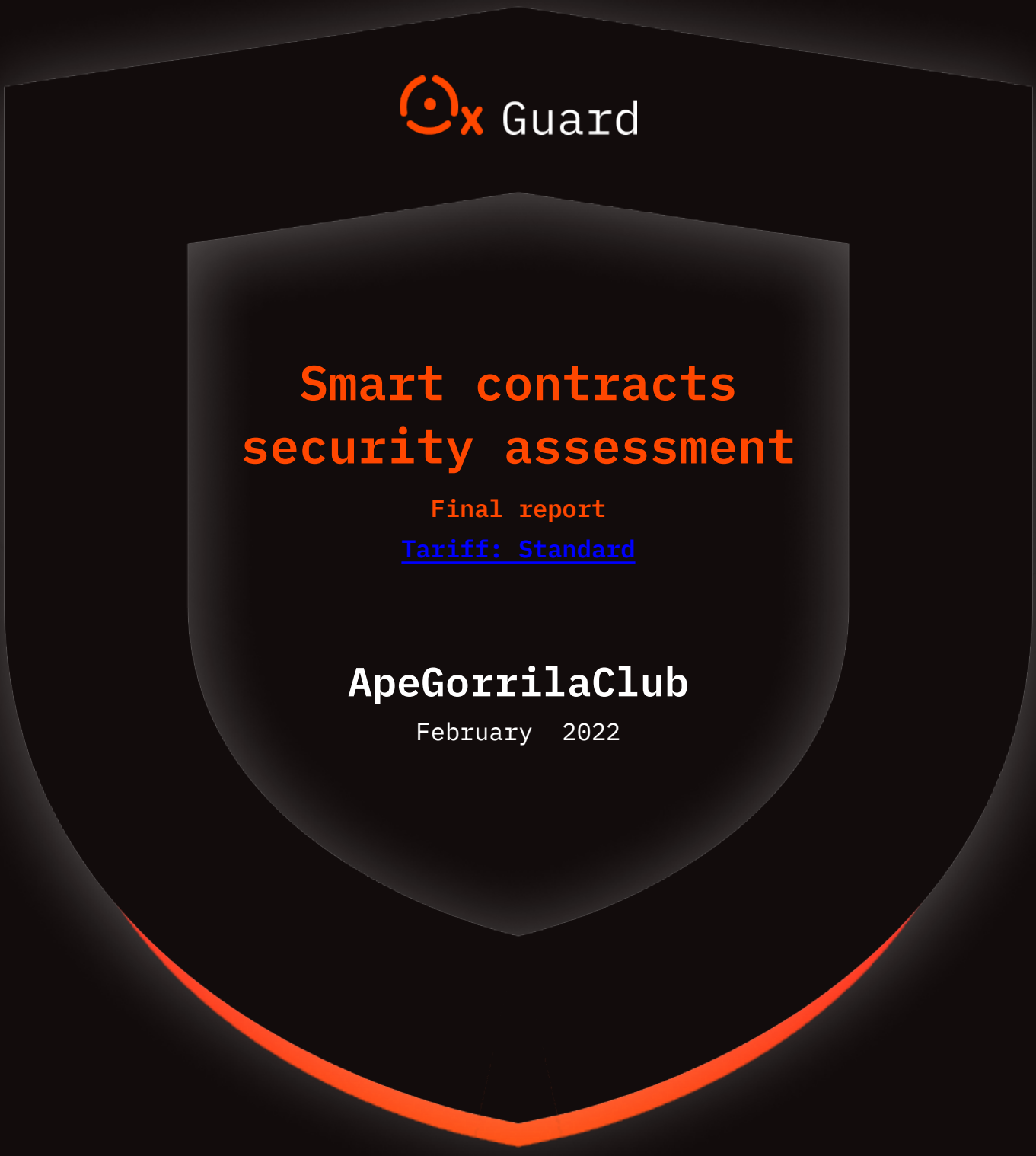

Guard

Smart contracts security assessment

Final report

Tariff: Standard

ApeGorrilaClub

February 2022

0xguard.com

hello@0xguard.com

Contents

Introduction

Standard ERC721 token contract with an added author tip feature. ERC721 interface is realized with the use of OpenZeppelin libraries, which is considered the best practice.

The contract can be considered an NFT service as it has a minting limit of 11337 tokens, with a max balance of 22 tokens per customer.

The contract allows buying NFT (mints tokens with unique id) on presale (or 'platinum' presale) or buying NFT (mints tokens with unique id) on public sale for different prices. The presales have limits: 1337 tokens for presale and from 20 up to 2000 for platinum presale.

The md5 sum of the file `apegorillaclub.sol` under investigation is `30cef7b098547209f62c3509759323bd`. The md5 sum of the file `paymentaplitter.sol` under investigation is `b54af5d36e01f9add273ee62d9eaa260`.

Update: md5 sum of the file with updated code `apegorillaclub_FINAL.sol` is `05c1190a7861eb8d76c9e8dae2644d57`.

Update 2: md5 sum of the file with updated code `apegorillaclub_NFT.sol` is `c6fb3e5caa188e028a857375ee125850`.

Name	ApeGorrilaClub
Audit date	2022-01-16 - 2022-02-04
Language	Solidity
Platform	Ethereum

Contracts checked

Name	Address
nft	

royalties

Procedure

We perform our audit according to the following procedure:

Automated analysis

- Scanning the project's smart contracts with several publicly available automated Solidity analysis tools
- Manual verification (reject or confirm) all the issues found by the tools

Manual audit

- Manually analyse smart contracts for security vulnerabilities
- Smart contracts' logic check

Known vulnerabilities checked

Title	Check result
Unencrypted Private Data On-Chain	not passed
Code With No Effects	passed
Message call with hardcoded gas amount	passed
Typographical Error	not passed
DoS With Block Gas Limit	passed
Presence of unused variables	not passed
Incorrect Inheritance Order	passed
Requirement Violation	passed
Weak Sources of Randomness from Chain Attributes	passed

Shadowing State Variables	passed
Incorrect Constructor Name	passed
Block values as a proxy for time	passed
Authorization through tx.origin	passed
DoS with Failed Call	passed
Delegatecall to Untrusted Callee	passed
Use of Deprecated Solidity Functions	passed
Assert Violation	passed
State Variable Default Visibility	passed
Reentrancy	passed
Unprotected SELFDESTRUCT Instruction	passed
Unprotected Ether Withdrawal	passed
Unchecked Call Return Value	passed
Floating Pragma	passed
Outdated Compiler Version	passed
Integer Overflow and Underflow	passed
Function Default Visibility	passed

Classification of issue severity

High severity	High severity issues can cause a significant or full loss of funds, change of contract ownership, major interference with contract logic. Such issues require immediate attention.
Medium severity	Medium severity issues do not pose an immediate risk, but can be detrimental to the client's reputation if exploited. Medium severity issues may lead to a contract failure and can be fixed by modifying the contract state or redeployment. Such issues require attention.

Low severity	Low severity issues do not cause significant destruction to the contract's functionality. Such issues are recommended to be taken into consideration.

Issues

High severity issues

1. Missing msg.value check - FIXED (nft)

The functions `platinumSaleMint()`, `presaleMint()` are `payable`, but there are no checks for the amount of `msg.value`. Any user, having peeped the `v, r, s` arguments in the previous transactions will be able to call these functions to mint tokens on presales without paying ether. Also, the call of the function`sendToCommunityWallet()` at L1356 will not transfer any ether to the `communityWallet`.

Moreover, the variable `mintCostPresale` is never checked.

Recommendation: Add a check for `msg.value` into the functions `platinumSaleMint()`, `presaleMint()`. Check the contract for desired logic.

Medium severity issues

1. Royalty not implemented (nft)

The website https://apegorilla.com indicates that royalties of 10% are expected when reselling tokens. But there is no royalty implementation in the code.

Recommendation: Check the desired logic.

2. Withdraw to only one account - FIXED (nft)

The owner has the ability to withdraw almost all funds to the '`communityWallet`', bypassing the shareholders. He needs to wait until all the NFTs are distributed, and then execute the

`transferToCommunityWallet()` function several times. In such a case, only 80% of the balance, which is less than 700 ether, will be distributed to the shareholders.

Recommendation: Since the team of 0xGuard hasn't been provided with the detailed documentation, we recommend limiting the number of calls to the '`transferToCommunityWallet()`' function

3. No support for ERC20 tokens (royalties)

If ERC20 tokens are sent to the contract address, they will be locked.

Recommendation: We recommend using OpenZeppelin implementation of the PaymentSplitter without modifications.

Low severity issues

1. Gas optimization (nft)

The variables `unrevealedURI`, `maxReservedMints`, `maxMints` could be declared as immutable to save gas.

2. Wrong code comment - FIXED (nft)

1) The comment at L1272 (amount 100) does not match the variable `maxReservedMints` value at L1273 (amount 50).

2) The comment message at L1201 should be changed: 'presale' to 'platinum presale'.

3. Costly loop (nft)

The `reservedMint()` function L1443 has expensive for() loop, especially when it is called with an argument close to '50'.

4. Account takeover risk (nft)

Note, that any of the hardcoded addresses would receive the accumulated payments in native tokens

even after losing control over that account.

Recommendation: We recommend avoiding using the EOA as hardcoded receivers.

5. Unused variables (nft)

The variable '`id_to_URI`' L641 is never used. It can be removed to save gas.

6. Using of outdated libraries and contracts (nft)

The used library '`Address`' and the contract '`Ownable`' from OpenZeppelin are outdated. We recommend using the latest libraries and contracts.

7. Reusing function return (nft)

The function `saleState()` is executed twice inside the functions `platinumSaleMint()` L1335, `presaleMint()` L1364, `publicSaleMint()` L1387. A local variable should be declared (e.g. 'State currentState = saleState()') to save gas inside the require checks.

8. Redundant code - FIXED (nft)

The function `fallback()` is redundant.

9. Variable visibility (nft)

Note, that every private variable of the contract (e.g. `maxTokensPresale`, `maxTokensPlatinumSale`, `_currentBaseURI`, `unrevealedURI`, `mintCostPresale`, `mintCostPublicSale`, `_reservedMints`, `maxReservedMints`, `dummy`, `dummy2`, `presaleLaunchTime`) can be read, even though it is private.

10. Redundant comparison - FIXED (nft)

The values `tokenMintedInPresale[tokenId_]` at L1415 and reveal at L1420 do not need to be compared to `true` (or `false`).

11. Random receipt (nft)

Even though the functions `platinumSaleMint()`, `presaleMint()`, `publicSaleMint()` are

payable, there is the possibility of a random receipt of user funds in the contract via `receive()` function.

12. Redundant modifier - FIXED (nft)

The modifier `onlyOwner` of the view functions `accountBalance()` L1475, `reservedMintsLeft()` L1485 is redundant and can be removed.

13. ERC721 modifications (nft)

The original ERC721 implementation was modified with:

1) '`maxBalance`' equal to 22 for each customer. It also affected the `_transfer()` function;

2) adding the unused variable '`id_to_URI`';

3) overridden `tokenURI()` function;

3) `setApprovalForAll()` function and deleted `_afterTokenTransfer()` function;

4) deleted `_afterTokenTransfer()` hook;

14. Constant variables - FIXED (nft)

The variables `shareholder1`, `shareholder2`, `shareholder3`, `shareholder4`, `shareholder5`, `shareholder6`, `communityWallet`, `dummy2` can be declared as constant.

Moreover the variables `shareholder2`, `shareholder3`, `shareholder4`, `shareholder5`, `shareholder6` are the same, so to save gas upon the execution of the function `withdrawAll()` only one variable can be used instead.

Note, unlike the `dummy` variable, the variable `dummy2` hasn't got a setter function.

15. Functions visibility - FIXED (nft)

Visibility of the functions `changeBaseURI()`, `platinumSaleMint()`, `presaleMint()`, `publicSaleMint()`, `reservedMint()`, `switchToPresale()`, `switchToPublicSale()`,

`burnUnmintedTokens()`, `changeDummy()`, `reservedMintsLeft()`, `withdrawAll()`, `transferToCommunityWallet()`, `maxMintsPerAddress()`, `publicSalelaunch()`, `presalelaunch()`, `toggleReveal()`, `addTokensPlatinumSale()`, `tokensLeftPlatinumSale()`, `tokensLeftPresale()` should be declared as external to save gas.

16. Incorrect payment config (royalties)

The contract initializes the constructor with the same payees, which is not supported. The contract will fail on deployment.

```
    //addresses to split payment with
    address[] private _team = [
        0x318cBF186eB13C74533943b054959867eE44eFFE, //project wallet
        0x318CBF186Eb13C74533943b054959867eE44EfF1, //wallet 2
        0x318cbF186Eb13c74533943b054959867Ee44eff2, //wallet 3
        0x318CbF186EB13C74533943B054959867ee44eFF3  //wallet 4
    ];

    constructor () PaymentSplitter(_team, _teamSharesSplit) payable {}
```

Recommendation: Set correct distribution config with different payees.

Conclusion

ApeGorrilaClub nft and royalties contracts were audited.

In audited contracts, 1 high and 3 medium Severity issues were found. 1 high, 1 medium, and several low severity issues were fixed with the update. The contracts highly depend on the owner's account.

Users must check that they are interacting with the same contract as was audited.

Update. The high severity, 1 medium severity, and 6 low severity issues were fixed in the update.

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 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 without 0xGuard prior written consent.

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 0xGuard 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 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.

Slither output

```
nft.mintCost() (apegorillaclub.sol#1552-1560) performs a multiplication on the result
of a division:
        -mintCostPublicSale + (((block.timestamp - publicSaleLaunchTime) / 604800) *
50000000000000000) (apegorillaclub.sol#1558)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-
multiply

nft.saleState() (apegorillaclub.sol#1539-1549) uses a dangerous strict equality:
        - presaleLaunchTime == 0 (apegorillaclub.sol#1540)
nft.saleState() (apegorillaclub.sol#1539-1549) uses a dangerous strict equality:
        - publicSaleLaunchTime == 0 (apegorillaclub.sol#1543)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-
strict-equalities

Reentrancy in nft.platinumSaleMint(uint256,uint8,bytes32,bytes32)
(apegorillaclub.sol#1335-1358):
        External calls:
        - _safeMint(msg.sender,tid) (apegorillaclub.sol#1350)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data)
(apegorillaclub.sol#983-993)
        State variables written after the call(s):
        - mintsPerAddress[msg.sender] += 1 (apegorillaclub.sol#1351)
        - numberOfTokensPlatinumSale += 1 (apegorillaclub.sol#1352)
Reentrancy in nft.presaleMint(uint256,uint8,bytes32,bytes32)
(apegorillaclub.sol#1364-1385):
        External calls:
        - _safeMint(msg.sender,tid) (apegorillaclub.sol#1379)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data)
(apegorillaclub.sol#983-993)
        State variables written after the call(s):
        - mintsPerAddress[msg.sender] += 1 (apegorillaclub.sol#1380)
        - numberOfTokensPresale += 1 (apegorillaclub.sol#1381)
Reentrancy in nft.publicSaleMint(uint256) (apegorillaclub.sol#1387-1407):
        External calls:
        - _safeMint(msg.sender,tid) (apegorillaclub.sol#1403)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data)
(apegorillaclub.sol#983-993)
```

```
        State variables written after the call(s):
        - mintsPerAddress[msg.sender] += 1 (apegorillaclub.sol#1404)
Reentrancy in nft.reservedMint(uint256) (apegorillaclub.sol#1434-1447):
        External calls:
        - _safeMint(msg.sender,tid) (apegorillaclub.sol#1440)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data)
(apegorillaclub.sol#983-993)
        State variables written after the call(s):
        - _reservedMints += 1 (apegorillaclub.sol#1442)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-
vulnerabilities-1

ERC721._checkOnERC721Received(address,address,uint256,bytes)
(apegorillaclub.sol#976-997) ignores return value by
IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data)
(apegorillaclub.sol#983-993)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

nft.addTokensPlatinumSale(uint256) (apegorillaclub.sol#1569-1572) should emit an event
for:
        - maxTokensPlatinumSale += number (apegorillaclub.sol#1571)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-
arithmetic

nft.changeDummy(address)._dummy (apegorillaclub.sol#1480) lacks a zero-check on :
                - dummy = _dummy (apegorillaclub.sol#1481)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-
address-validation

ERC721._checkOnERC721Received(address,address,uint256,bytes)
(apegorillaclub.sol#976-997) has external calls inside a loop:
IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data)
(apegorillaclub.sol#983-993)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-
a-loop

Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval
(apegorillaclub.sol#983)' in
ERC721._checkOnERC721Received(address,address,uint256,bytes)
(apegorillaclub.sol#976-997) potentially used before declaration: retval ==
IERC721Receiver.onERC721Received.selector (apegorillaclub.sol#984)
```

```
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason
(apegorillaclub.sol#985)' in
ERC721._checkOnERC721Received(address,address,uint256,bytes)
(apegorillaclub.sol#976-997) potentially used before declaration: reason.length == 0
(apegorillaclub.sol#986)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason
(apegorillaclub.sol#985)' in
ERC721._checkOnERC721Received(address,address,uint256,bytes)
(apegorillaclub.sol#976-997) potentially used before declaration:
revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (apegorillaclub.sol#990)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-
declaration-usage-of-local-variables

Reentrancy in nft.platinumSaleMint(uint256,uint8,bytes32,bytes32)
(apegorillaclub.sol#1335-1358):
        External calls:
        - _safeMint(msg.sender,tid) (apegorillaclub.sol#1350)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data)
(apegorillaclub.sol#983-993)
        State variables written after the call(s):
        - tokenMintedInPresale[tid] = true (apegorillaclub.sol#1353)
Reentrancy in nft.presaleMint(uint256,uint8,bytes32,bytes32)
(apegorillaclub.sol#1364-1385):
        External calls:
        - _safeMint(msg.sender,tid) (apegorillaclub.sol#1379)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data)
(apegorillaclub.sol#983-993)
        State variables written after the call(s):
        - tokenMintedInPresale[tid] = true (apegorillaclub.sol#1382)
Reentrancy in nft.reservedMint(uint256) (apegorillaclub.sol#1434-1447):
        External calls:
        - _safeMint(msg.sender,tid) (apegorillaclub.sol#1440)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data)
(apegorillaclub.sol#983-993)
        State variables written after the call(s):
        - mintsPerAddress[msg.sender] += 1 (apegorillaclub.sol#1441)
        - tokenMintedInPresale[tid] = true (apegorillaclub.sol#1444)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-
vulnerabilities-2

nft.publicSaleMint(uint256) (apegorillaclub.sol#1387-1407) uses timestamp for
comparisons
```

```
        Dangerous comparisons:
        - require(bool,string)(msg.value >= number * mintCost(),Insufficient Funds to
mint this number of Tokens!) (apegorillaclub.sol#1392)
nft.publicSalelaunch() (apegorillaclub.sol#1527-1530) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(publicSaleLaunchTime != 0,Public Sale has not opened
yet!) (apegorillaclub.sol#1528)
nft.presalelaunch() (apegorillaclub.sol#1533-1536) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(presaleLaunchTime != 0,Presale has not opened yet!)
(apegorillaclub.sol#1534)
nft.saleState() (apegorillaclub.sol#1539-1549) uses timestamp for comparisons
        Dangerous comparisons:
        - presaleLaunchTime == 0 (apegorillaclub.sol#1540)
        - publicSaleLaunchTime == 0 (apegorillaclub.sol#1543)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-
timestamp

Address.isContract(address) (apegorillaclub.sol#187-197) uses assembly
        - INLINE ASM (apegorillaclub.sol#193-195)
Address.verifyCallResult(bool,bytes,string) (apegorillaclub.sol#356-376) uses assembly
        - INLINE ASM (apegorillaclub.sol#368-371)
ERC721._checkOnERC721Received(address,address,uint256,bytes)
(apegorillaclub.sol#976-997) uses assembly
        - INLINE ASM (apegorillaclub.sol#989-991)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

nft.tokenURI(uint256) (apegorillaclub.sol#1412-1430) compares to a boolean constant:
        -reveal == false && totalSupply() < maxTotalTokens (apegorillaclub.sol#1420)
nft.tokenURI(uint256) (apegorillaclub.sol#1412-1430) compares to a boolean constant:
        -tokenMintedInPresale[tokenId_] == true (apegorillaclub.sol#1415)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-
equality

ERC721Enumerable._removeTokenFromAllTokensEnumeration(uint256)
(apegorillaclub.sol#1167-1185) has costly operations inside a loop:
        - delete _allTokensIndex[tokenId] (apegorillaclub.sol#1183)
ERC721Enumerable._removeTokenFromAllTokensEnumeration(uint256)
(apegorillaclub.sol#1167-1185) has costly operations inside a loop:
        - _allTokens.pop() (apegorillaclub.sol#1184)
ERC721Enumerable._removeTokenFromOwnerEnumeration(address,uint256)
```

```
(apegorillaclub.sol#1142-1160) has costly operations inside a loop:
        - delete _ownedTokensIndex[tokenId] (apegorillaclub.sol#1158)
nft.platinumSaleMint(uint256,uint8,bytes32,bytes32) (apegorillaclub.sol#1335-1358) has
costly operations inside a loop:
        - numberOfTokensPlatinumSale += 1 (apegorillaclub.sol#1352)
nft.presaleMint(uint256,uint8,bytes32,bytes32) (apegorillaclub.sol#1364-1385) has
costly operations inside a loop:
        - numberOfTokensPresale += 1 (apegorillaclub.sol#1381)
nft.reservedMint(uint256) (apegorillaclub.sol#1434-1447) has costly operations inside a
loop:
        - _reservedMints += 1 (apegorillaclub.sol#1442)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-
operations-inside-a-loop

Address.functionCall(address,bytes) (apegorillaclub.sol#240-242) is never used and
should be removed
Address.functionCall(address,bytes,string) (apegorillaclub.sol#250-256) is never used
and should be removed
Address.functionCallWithValue(address,bytes,uint256) (apegorillaclub.sol#269-275) is
never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string)
(apegorillaclub.sol#283-294) is never used and should be removed
Address.functionDelegateCall(address,bytes) (apegorillaclub.sol#329-331) is never used
and should be removed
Address.functionDelegateCall(address,bytes,string) (apegorillaclub.sol#339-348) is
never used and should be removed
Address.functionStaticCall(address,bytes) (apegorillaclub.sol#302-304) is never used
and should be removed
Address.functionStaticCall(address,bytes,string) (apegorillaclub.sol#312-321) is never
used and should be removed
Address.sendValue(address,uint256) (apegorillaclub.sol#215-220) is never used and
should be removed
Address.verifyCallResult(bool,bytes,string) (apegorillaclub.sol#356-376) is never used
and should be removed
Context._msgData() (apegorillaclub.sol#88-90) is never used and should be removed
ERC721._baseURI() (apegorillaclub.sol#709-711) is never used and should be removed
ERC721._burn(uint256) (apegorillaclub.sol#909-921) is never used and should be removed
Strings.toHexString(uint256) (apegorillaclub.sol#40-51) is never used and should be
removed
Strings.toHexString(uint256,uint256) (apegorillaclub.sol#56-66) is never used and
should be removed
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (apegorillaclub.sol#4) allows old versions
solc-0.8.11 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-
versions-of-solidity

Low level call in Address.sendValue(address,uint256) (apegorillaclub.sol#215-220):
        - (success) = recipient.call{value: amount}() (apegorillaclub.sol#218)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string)
(apegorillaclub.sol#283-294):
        - (success,returndata) = target.call{value: value}(data)
(apegorillaclub.sol#292)
Low level call in Address.functionStaticCall(address,bytes,string)
(apegorillaclub.sol#312-321):
        - (success,returndata) = target.staticcall(data) (apegorillaclub.sol#319)
Low level call in Address.functionDelegateCall(address,bytes,string)
(apegorillaclub.sol#339-348):
        - (success,returndata) = target.delegatecall(data) (apegorillaclub.sol#346)
Low level call in nft.withdraw(address,uint256) (apegorillaclub.sol#1503-1506):
        - (sent) = _address.call{value: amount}() (apegorillaclub.sol#1504)
Low level call in nft.sendToCommunityWallet() (apegorillaclub.sol#1516-1519):
        - (sent) = address(communityWallet).call{value: msg.value}()
(apegorillaclub.sol#1517)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-
calls

Parameter ERC721.safeTransferFrom(address,address,uint256,bytes)._data
(apegorillaclub.sol#786) is not in mixedCase
Variable ERC721._owners (apegorillaclub.sol#629) is not in mixedCase
Variable ERC721.id_to_URI (apegorillaclub.sol#641) is not in mixedCase
Contract nft (apegorillaclub.sol#1188-1585) is not in CapWords
Parameter nft.isValidAccessMessage(address,uint8,bytes32,bytes32,uint256)._add
(apegorillaclub.sol#1318) is not in mixedCase
Parameter nft.isValidAccessMessage(address,uint8,bytes32,bytes32,uint256)._v
(apegorillaclub.sol#1318) is not in mixedCase
Parameter nft.isValidAccessMessage(address,uint8,bytes32,bytes32,uint256)._r
(apegorillaclub.sol#1318) is not in mixedCase
Parameter nft.isValidAccessMessage(address,uint8,bytes32,bytes32,uint256)._s
(apegorillaclub.sol#1318) is not in mixedCase
Parameter nft.platinumSaleMint(uint256,uint8,bytes32,bytes32)._v
```

```
(apegorillaclub.sol#1335) is not in mixedCase
Parameter nft.platinumSaleMint(uint256,uint8,bytes32,bytes32)._r
(apegorillaclub.sol#1335) is not in mixedCase
Parameter nft.platinumSaleMint(uint256,uint8,bytes32,bytes32)._s
(apegorillaclub.sol#1335) is not in mixedCase
Parameter nft.presaleMint(uint256,uint8,bytes32,bytes32)._v (apegorillaclub.sol#1364)
is not in mixedCase
Parameter nft.presaleMint(uint256,uint8,bytes32,bytes32)._r (apegorillaclub.sol#1364)
is not in mixedCase
Parameter nft.presaleMint(uint256,uint8,bytes32,bytes32)._s (apegorillaclub.sol#1364)
is not in mixedCase
Parameter nft.changeDummy(address)._dummy (apegorillaclub.sol#1480) is not in mixedCase
Parameter nft.withdraw(address,uint256)._address (apegorillaclub.sol#1503) is not in
mixedCase
Constant nft.mintCostPresale (apegorillaclub.sol#1210) is not in
UPPER_CASE_WITH_UNDERSCORES
Constant nft.mintCostPublicSale (apegorillaclub.sol#1211) is not in
UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-
solidity-naming-conventions

Variable nft.shareholder1 (apegorillaclub.sol#1222) is too similar to nft.shareholder2
(apegorillaclub.sol#1223)
Variable nft.shareholder1 (apegorillaclub.sol#1222) is too similar to nft.shareholder3
(apegorillaclub.sol#1224)
Variable nft.shareholder1 (apegorillaclub.sol#1222) is too similar to nft.shareholder4
(apegorillaclub.sol#1225)
Variable nft.shareholder1 (apegorillaclub.sol#1222) is too similar to nft.shareholder5
(apegorillaclub.sol#1226)
Variable nft.shareholder1 (apegorillaclub.sol#1222) is too similar to nft.shareholder6
(apegorillaclub.sol#1227)
Variable nft.shareholder2 (apegorillaclub.sol#1223) is too similar to nft.shareholder3
(apegorillaclub.sol#1224)
Variable nft.shareholder2 (apegorillaclub.sol#1223) is too similar to nft.shareholder4
(apegorillaclub.sol#1225)
Variable nft.shareholder2 (apegorillaclub.sol#1223) is too similar to nft.shareholder5
(apegorillaclub.sol#1226)
Variable nft.shareholder2 (apegorillaclub.sol#1223) is too similar to nft.shareholder6
(apegorillaclub.sol#1227)
Variable nft.shareholder3 (apegorillaclub.sol#1224) is too similar to nft.shareholder4
(apegorillaclub.sol#1225)
```

```
Variable nft.shareholder3 (apegorillaclub.sol#1224) is too similar to nft.shareholder5
(apegorillaclub.sol#1226)
Variable nft.shareholder3 (apegorillaclub.sol#1224) is too similar to nft.shareholder6
(apegorillaclub.sol#1227)
Variable nft.shareholder4 (apegorillaclub.sol#1225) is too similar to nft.shareholder5
(apegorillaclub.sol#1226)
Variable nft.shareholder4 (apegorillaclub.sol#1225) is too similar to nft.shareholder6
(apegorillaclub.sol#1227)
Variable nft.shareholder5 (apegorillaclub.sol#1226) is too similar to nft.shareholder6
(apegorillaclub.sol#1227)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-
are-too-similar

ERC721.id_to_URI (apegorillaclub.sol#641) is never used in nft
(apegorillaclub.sol#1188-1585)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-
variable

nft.communityWallet (apegorillaclub.sol#1228) should be constant
nft.dummy2 (apegorillaclub.sol#1232) should be constant
nft.shareholder1 (apegorillaclub.sol#1222) should be constant
nft.shareholder2 (apegorillaclub.sol#1223) should be constant
nft.shareholder3 (apegorillaclub.sol#1224) should be constant
nft.shareholder4 (apegorillaclub.sol#1225) should be constant
nft.shareholder5 (apegorillaclub.sol#1226) should be constant
nft.shareholder6 (apegorillaclub.sol#1227) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-
variables-that-could-be-declared-constant

renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (apegorillaclub.sol#143-145)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (apegorillaclub.sol#151-154)
name() should be declared external:
        - ERC721.name() (apegorillaclub.sol#688-690)
symbol() should be declared external:
        - ERC721.symbol() (apegorillaclub.sol#695-697)
tokenURI(uint256) should be declared external:
        - ERC721.tokenURI(uint256) (apegorillaclub.sol#702)
        - nft.tokenURI(uint256) (apegorillaclub.sol#1412-1430)
approve(address,uint256) should be declared external:
```

```
        - ERC721.approve(address,uint256) (apegorillaclub.sol#716-726)
setApprovalForAll(address,bool) should be declared external:
        - ERC721.setApprovalForAll(address,bool) (apegorillaclub.sol#740-745)
transferFrom(address,address,uint256) should be declared external:
        - ERC721.transferFrom(address,address,uint256) (apegorillaclub.sol#757-766)
safeTransferFrom(address,address,uint256) should be declared external:
        - ERC721.safeTransferFrom(address,address,uint256) (apegorillaclub.sol#771-777)
tokenOfOwnerByIndex(address,uint256) should be declared external:
        - ERC721Enumerable.tokenOfOwnerByIndex(address,uint256)
(apegorillaclub.sol#1062-1065)
tokenByIndex(uint256) should be declared external:
        - ERC721Enumerable.tokenByIndex(uint256) (apegorillaclub.sol#1077-1080)
changeBaseURI(string) should be declared external:
        - nft.changeBaseURI(string) (apegorillaclub.sol#1295-1297)
platinumSaleMint(uint256,uint8,bytes32,bytes32) should be declared external:
        - nft.platinumSaleMint(uint256,uint8,bytes32,bytes32)
(apegorillaclub.sol#1335-1358)
presaleMint(uint256,uint8,bytes32,bytes32) should be declared external:
        - nft.presaleMint(uint256,uint8,bytes32,bytes32) (apegorillaclub.sol#1364-1385)
publicSaleMint(uint256) should be declared external:
        - nft.publicSaleMint(uint256) (apegorillaclub.sol#1387-1407)
reservedMint(uint256) should be declared external:
        - nft.reservedMint(uint256) (apegorillaclub.sol#1434-1447)
switchToPresale() should be declared external:
        - nft.switchToPresale() (apegorillaclub.sol#1450-1453)
switchToPublicSale() should be declared external:
        - nft.switchToPublicSale() (apegorillaclub.sol#1456-1460)
burnUnmintedTokens() should be declared external:
        - nft.burnUnmintedTokens() (apegorillaclub.sol#1463-1472)
changeDummy(address) should be declared external:
        - nft.changeDummy(address) (apegorillaclub.sol#1480-1482)
reservedMintsLeft() should be declared external:
        - nft.reservedMintsLeft() (apegorillaclub.sol#1485-1487)
withdrawAll() should be declared external:
        - nft.withdrawAll() (apegorillaclub.sol#1490-1501)
transferToCommunityWallet() should be declared external:
        - nft.transferToCommunityWallet() (apegorillaclub.sol#1509-1514)
maxMintsPerAddress() should be declared external:
        - nft.maxMintsPerAddress() (apegorillaclub.sol#1522-1524)
publicSalelaunch() should be declared external:
        - nft.publicSalelaunch() (apegorillaclub.sol#1527-1530)
```

```
presalelaunch() should be declared external:
        - nft.presalelaunch() (apegorillaclub.sol#1533-1536)
toggleReveal() should be declared external:
        - nft.toggleReveal() (apegorillaclub.sol#1563-1566)
addTokensPlatinumSale(uint256) should be declared external:
        - nft.addTokensPlatinumSale(uint256) (apegorillaclub.sol#1569-1572)
tokensLeftPlatinumSale() should be declared external:
        - nft.tokensLeftPlatinumSale() (apegorillaclub.sol#1575-1577)
tokensLeftPresale() should be declared external:
        - nft.tokensLeftPresale() (apegorillaclub.sol#1580-1582)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-
function-that-could-be-declared-external
```

0x Guard