verichains

SECURITY AUDIT OF

ARENA PIXEL

Public Report

Jun 14, 2023

Verichains Lab

info@verichains.io

https://www.verichains.io

Driving Technology > Forward

ABBREVIATIONS

Name	Description
Ethereum	An open source platform based on blockchain technology to create and distribute smart contracts and decentralized applications.
Ether (ETH)	A cryptocurrency whose blockchain is generated by the Ethereum platform. Ether is used for payment of transactions and computing services in the Ethereum network.
Smart contract	A computer protocol intended to digitally facilitate, verify or enforce the negotiation or performance of a contract.
Solidity	A contract-oriented, high-level language for implementing smart contracts for the Ethereum platform.
Solc	A compiler for Solidity.
ERC20	ERC20 (BEP20 in Binance Smart Chain or *x*RP20 in other chains) tokens are blockchain-based assets that have value and can be sent and received. The primary difference with the primary coin is that instead of running on their own blockchain, ERC20 tokens are issued on a network that supports smart contracts such as Ethereum or Binance Smart Chain.

EXECUTIVE SUMMARY

This Security Audit Report was prepared by Verichains Lab on Jun 14, 2023. We would like to thank the Arena Pixel for trusting Verichains Lab in auditing smart contracts. Delivering high-quality audits is always our top priority.

This audit focused on identifying security flaws in code and the design of the Arena Pixel. The scope of the audit is limited to the source code files provided to Verichains. Verichains Lab completed the assessment using manual, static, and dynamic analysis techniques.

During the audit process, the audit team had identified some vulnerable issues in the smart contracts code.

TABLE OF CONTENTS

1. MANAGEMENT SUMMARY

1.1. About Arena Pixel

Arena Pixel is an action arena game with colorful and attractive classic pixel graphics. In this game, players can choose a character and participate in fierce battles with other opponents in the arena. With many unique weapons and skills, players must fight to win and become the champion. The game has many modes and provides players with a simple, rich graphical experience

Arena Pixel is a P2E game project incorporating NFT collectability on the blockchain. Players can own in-game items and experience them in a graphic style from 8-bit games. With a completely new mechanism, we call it Tap to Earn.

1.2. Audit scope

This audit focused on identifying security flaws in code and the design of the Arena Pixel.

The audited contracts from address `0xf89E1F8442850504b595b1F99DEA8fEe3B6FcDf7`. The latest version of the deployed smart contract is listed in Table.

FIELD	VALUE
Contract Name	ArenaPixelToken
Contract Address	0x247B62BAACBeE6BAc0E6b3e48CA86BEA23800E70
Compiler Version	v0.8.18+commit.87f61d96
Optimization Enabled	Yes with 200 runs
Explorer	*https://bscscan.com/address/0x247B62BAACBeE6BAc0E6b3e48CA86BEA23800E70*

Table 1. The latest version of ArenaPixelToken smart contract details

1.3. Audit methodology

Our security audit process for smart contract includes two steps:

- Smart contract codes are scanned/tested for commonly known and more specific vulnerabilities using public and RK87, our in-house smart contract security analysis tool.
- Manual audit of the codes for security issues. The contracts are manually analyzed to look for any potential problems.

Following is the list of commonly known vulnerabilities that were considered during the audit of the smart contract:

- Integer Overflow and Underflow
- Timestamp Dependence
- Race Conditions
- Transaction-Ordering Dependence
- DoS with (Unexpected) revert
- DoS with Block Gas Limit
- Gas Usage, Gas Limit and Loops
- Redundant fallback function
- Unsafe type Inference
- Reentrancy
- Explicit visibility of functions state variables (external, internal, private and public)
- Logic Flaws

For vulnerabilities, we categorize the findings into categories as listed in table below, depending on their severity level:

SEVERITY LEVEL	DESCRIPTION
CRITICAL	A vulnerability that can disrupt the contract functioning; creates a critical risk to the contract; required to be fixed immediately.
HIGH	A vulnerability that could affect the desired outcome of executing the contract with high impact; needs to be fixed with high priority.
MEDIUM	A vulnerability that could affect the desired outcome of executing the contract with medium impact in a specific scenario; needs to be fixed.
LOW	An issue that does not have a significant impact, can be considered as less important.

Table 2. Severity levels

1.4. Disclaimer

Please note that security auditing cannot uncover all existing vulnerabilities, and even an audit in which no vulnerabilities are found is not a guarantee for a 100% secure smart contract. However, auditing allows discovering vulnerabilities that were unobserved, overlooked during development and areas where additional security measures are necessary.

2. AUDIT RESULT

2.1. Overview

Table lists some properties of the audited Arena Pixel (as of the report writing time).

PROPERTY	VALUE
Name	Arena Pixel
Symbol	APX
Decimals	18
Total Supply	80,000,000 ($x10^{18}$) Note: the number of decimals is 18, so the total representation token will be 80,000,000 or 80 million.

Table 3. The Arena Pixel properties

2.2. Contract codes

The Arena Pixel was written in `Solidity` language, with the required version to be `^0.8.16`.

2.2.1. ArenaPixelToken contract

ArenaPixelToken token extends `IERC20`, `Context`, and `Owable` contracts. The smart contract handles the vesting of PLX tokens for developers and ecosystem users and the release of vested tokens to them.

An owner of the contract is `0x2F096f22209Eb6d824203686335b3816762C782d`, a marketing wallet is `0x0649AFa989d32752030a15Aea2256EB1EfE68E9c`, an ecosystem wallet is `0xf116250bC7Ad73AD591F27F675FfDD4D89b10a68`, and a dev wallet is `0x8423529E8778eadc5E64871F9619aeAEE65c1368`. The owner can change the ecosystem wallet, dev wallet, marketing wallet, and vesting period.

On the Binance Smart Chain, the ArenaPixelToken contract has been deployed at address `0x4e7cF0d2A8FfB0A0f6e6eA3cF5Bb9f4B2aAeFbF6`. A marketing's wallet and the contract are initialized with 56,000,000 and 24,000,000 $x10^{18}$ tokens.

At the first generation event, the contract will release 30% of 20,000,000 $x10^{18}$ tokens for ecosystem wallet and 20% of 4,000,000 $x10^{18}$ tokens for the developer wallet. The remaining 50% will be locked into the contract and released after a period of time. Additional, the TGE launch at Saturday, 17 June 2023 09:00:00+0

The contract takes fees from each transaction and distributes them to the marketing wallet. The ecosystem wallet, dev wallet, and marketing wallet are all excluded from the fee. The fee is 1% of the transaction amount, and the owner can change the fee amount and add or remove the excluded addresses.

The contract has a blacklist mechanism to prevent the blacklisted addresses from transferring tokens. The owner can add or remove addresses from the blacklist.

Anyone can call the burn function to burn their own tokens, and the total supply of tokens will be reduced.

The owner of the contract can pause the contract to prevent it from transferring tokens or stop delivering tokens via vesting. The owner can also unpause the contract to resume it.

UPDATES

Jun 14, 2023, the Arena Pixel fixed issues are reported by Verichains and deployed a patched version on `0x247B62BAACBeE6BAc0E6b3e48CA86BEA23800E70`. The patched version has an owner is `0xB6062C85d74590bec7c8b13df734fBb17515e5c9`.

2.3. Findings

During the audit process, the audit team found some vulnerability issues in the given version of Arena Pixel.

2.3.1. A blacklist mechanism of transferFrom function can be bypassed - MEDIUM

A `transferFrom` function is used for transfer from one user to another user. The function will check the blacklist before transfer. However, if a blacklisted user uses another wallet to call with a `sender` parameter is a blacklist, the transfer will be successful.

```
function transferFrom(address sender, address recipient, uint256 amount) public override
returns (bool) {
        require(msg.sender != recipient, "Cannot transfer to self");
        require(isBlacklist[msg.sender] == false, "Blacklisted");

        // ...
}
```

RECOMMENDATION

We recommended adding a blacklist check for the `sender` parameter.

UPDATES

- *Jun 14, 2023*: The team has been acknowledged and fixed by the Arena Pixel team.

2.3.2. Lack of input validation when set start time for eco stages details and dev stage details - INFORMATIVE

`setEcoStage` and `setDevStage` are methods that allow the owner to set the start time for vesting in eco stages and dev stages. The `ecoStageDetails` and `devStageDetails` storage variables are initialized starting from 1, not zero.

Currently, there is no validation in place to check the validity of the `_stages` values containing zero when the owner sets the start time for the eco stage and dev stage.

```
constructor(...){
    // ignore index 0
    for(uint256 index = 0; index < _ecoStageRelease.length; index++){
        EcoStageDetail memory stage;
        stage.id = index+1;
        stage.stageRelease = _ecoStageRelease[index];

        ecoStageDetails[index+1] = stage;
    }

    // ignore index 0
    for(uint256 index = 0; index < _devStageRelease.length; index++){
        DevStageDetail memory stage;
        stage.id = index+1;
        stage.stageRelease = _devStageRelease[index];

        devStageDetails[index+1] = stage;
    }
}

function setEcoStage (uint256[] calldata _stages, uint256[] calldata _startTime) public
onlyOwner {
    require(_stages.length == _startTime.length, "Array length not match");
    // missing check stages include 0

    for(uint256 index = 0; index < _stages.length; index++){
        // ...
    }
}

function setDevStage (uint256[] memory _stages,uint256[] memory _startTime) public
onlyOwner {
    require(_stages.length == _startTime.length, "Array length not match");
    // missing check stages include 0

    for(uint256 index = 0; index < _stages.length; index++){
        // ...
    }
}
```

RECOMMENDATION

We recommend adding a validation check to ensure that the stages values are greater than zero.

UPDATES

- *Jun 14, 2023*: The team has been acknowledged and fixed by the Arena Pixel team.

2.3.3. The lack of event emission when change configurations - INFORMATIVE

Certain methods, including setSellTax, setBuyTax, enableTax, addExcludeFee, removeExcludeFee, setMktWallet, setEcosystemWallet, setDevWallet, and setPauseContract, suffer from the lack of event emission for global variable adjustments.

These effects include worse user experience, limited automation, lower transparency, and insufficient audit trails.

RECOMMENDATION

We recommended implementing event emissions, standardizing event structures, conducting thorough testing and auditing, and promoting documentation and best practices.

UPDATES

- *Jun 14, 2023*: The team has been acknowledged and fixed by the Arena Pixel team.

2.3.4. Missing zero address validation - INFORMATIVE

The missing zero address validation in certain methods, namely `setMktWallet`, `setEcosystemWallet`, and `setDevWallet`, has negative implications. When the owner sets the zero address as the wallet address, it prevents the contract from transferring tokens to other users. Consequently, the owner loses control of the tokens.

RECOMMENDATION

We recommended adding a validation check to ensure that the input values are not the zero address.

UPDATES

- *Jun 14, 2023*: The team has been acknowledged and fixed by the Arena Pixel team.

2.3.5. Approve for a pancake pair allowed to spend all tokens that is unnecessary - INFORMATIVE

The Pancake pair is able to use all tokens when using the "approve" procedure. Since the Pancake pair only needs to expend the tokens required for the swap, this is not necessary.

```
constructor(...){
    _approve(address(this), address(uniswapV2Router), ~uint256(0));
}
```

RECOMMENDATION

We recommended removing this line of code.

UPDATES

- *Jun 14, 2023*: The team has been acknowledged and fixed by the Arena Pixel team.

2.3.6. Centralized mechanism INFORMATIVE

Currently, the contract uses a centralized mechanism to allow the `owner` to interact with the functions of the contract, such as: setPauseContract, addBlacklist, removeBlacklist, setSellTax, setBuyTax, enableTax, addExcludeFee, removeExcludeFee, removeExcludeFee, setEcosystemWallet, setDevWallet, setTgeTime, setEcoStage, setDevStage.

Any compromise to the `owner` account may allow the hacker to take advantage of this.

RECOMMENDATION

We strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices.

UPDATES

- *Jun 14, 2023*: The team has been acknowledged.

APPENDIX

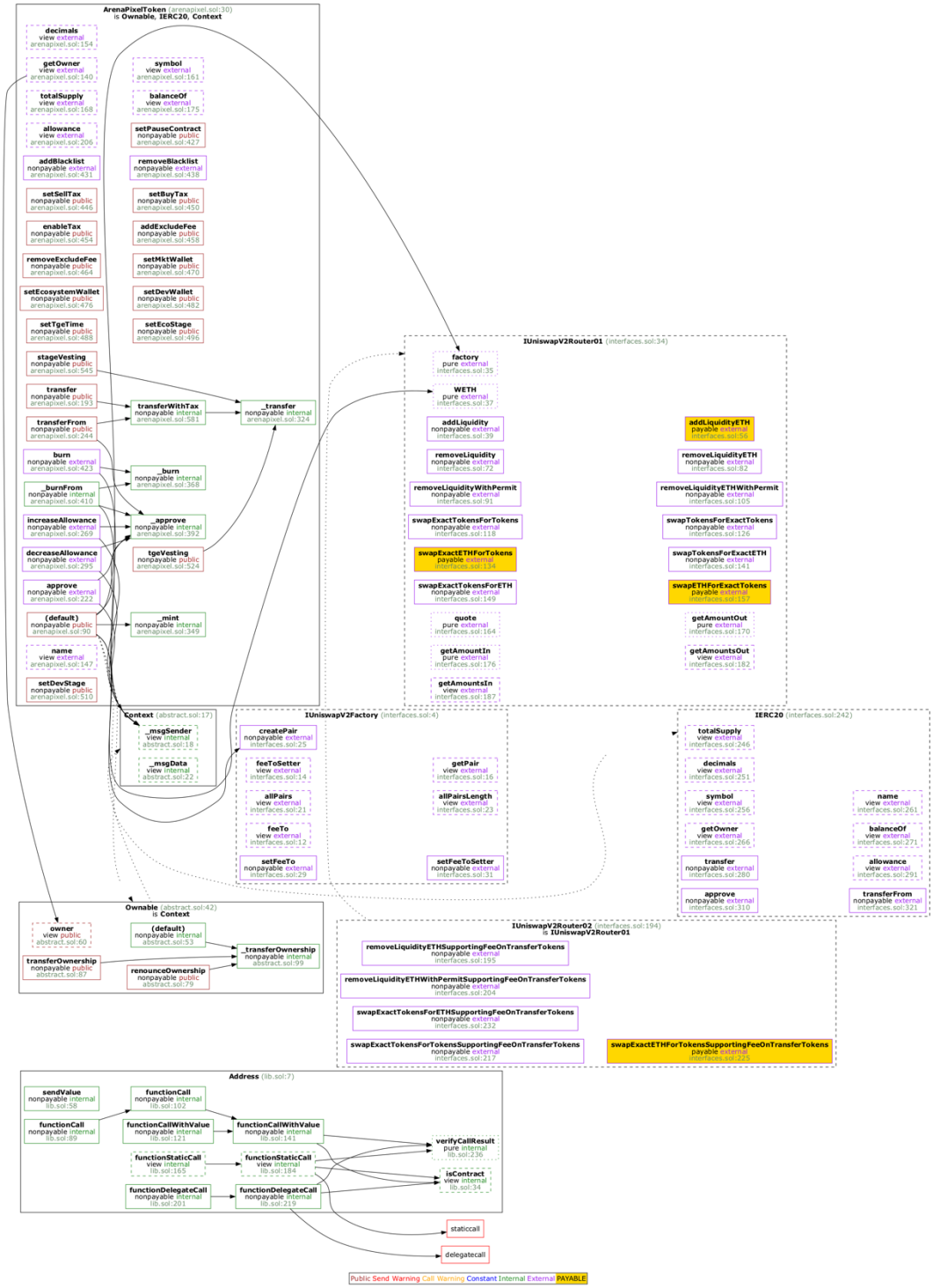

Image 1. Arena Pixel call graph

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Jun 13, 2023*	Public Report	Verichains Lab
1.1	*Jun 14, 2023*	Public Report	Verichains Lab

Table 4. Report versions history