Cyberscope

Audit Report

PAW

December 2023

Network	ETH
Address	0x419777D3E39AA9b00405724EaCE5ea57620c9062
Minimal Proxy	0x3669c9E2467fef198FB7963A2dF887D0a005216C
Audited by	© cyberscope

Analysis

🔴 Critical 🟡 Medium ⚪ Minor / Informative 🔵 Pass

Severity	Code	Description	Status
⚪	ST	Stops Transactions	Renounced
🔵	OTUT	Transfers User's Tokens	Passed
🔵	ELFM	Exceeds Fees Limit	Passed
⚪	MT	Mints Tokens	Renounced
🔵	BT	Burns Tokens	Passed
🔵	BC	Blacklists Addresses	Passed

Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L19	Stable Compiler Version	Unresolved

Table of Contents

Analysis **1**

Diagnostics **2**

Table of Contents **3**

Review **4**

Audit Updates 4

Source Files 4

Findings Breakdown **6**

ST - Stops Transactions 7

Description 7

Recommendation 7

Team Update 8

MT - Mints Tokens 9

Description 9

Recommendation 9

Team Update 9

L04 - Conformance to Solidity Naming Conventions 11

Description 11

Recommendation 11

L19 - Stable Compiler Version 13

Description 13

Recommendation 13

Functions Analysis **14**

Inheritance Graph **15**

Flow Graph **16**

Summary **17**

Initial Audit, 3 Dec 2023 17

Disclaimer **18**

About Cyberscope **19**

Review

Contract Name	UltimateTokenOwnable
Compiler Version	v0.8.19+commit.7dd6d404
Optimization	100 runs
Explorer	https://etherscan.io/address/0x419777d3e39aa9b00405724eace5ea57620c9062
Address	0x419777d3e39aa9b00405724eace5ea57620c9062
Network	ETH
Symbol	PAW
Decimals	18
Total Supply	1,000,000,000,000,000

Audit Updates

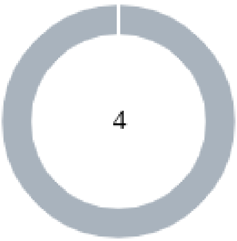
Initial Audit	02 Dec 2023

Source Files

Filename	SHA256
contracts/UltimateTokenOwnable.sol	0bccf54aa13263dc72655a286971ed2e8e04eaae2437814fb0ccf53ef3eec75e
contracts/core/Pausable.sol	9b4a40df3813cd5ee52d489642db6cde409fe81d600651a2978ba2756bd5997f

contracts/core/Ownable.sol	6f29566565861b97b58f90a779a352904dfe1d899494020122c0de3ea350ef63
contracts/core/Initializable.sol	55ef499aa3f25d27eb9f441aebb9ebe8f2ee14b9a84337abdaf643c6f9fc1cad
contracts/core/ERC20.sol	a3458afa93dc123b5271219f6baaa07490419e99d7cdb539e3238178aef0466b
contracts/core/libraries/Address.sol	15525c114f8958b4623707971f3ba8affb5f05ce20c8e81bf36d4f5e31c62b77
contracts/core/interfaces/IERC20Metadata.sol	88a52a0dc7d80a519b1db3440f3dc559600c98f78dbc29229af0615dee47848d
contracts/core/interfaces/IERC20.sol	d1a34b15495eb9acd4a12406629ad31007bb5bb81c38f043f1433748865f3c82

Findings Breakdown



● Critical		0
● Medium		0
● Minor / Informative		4

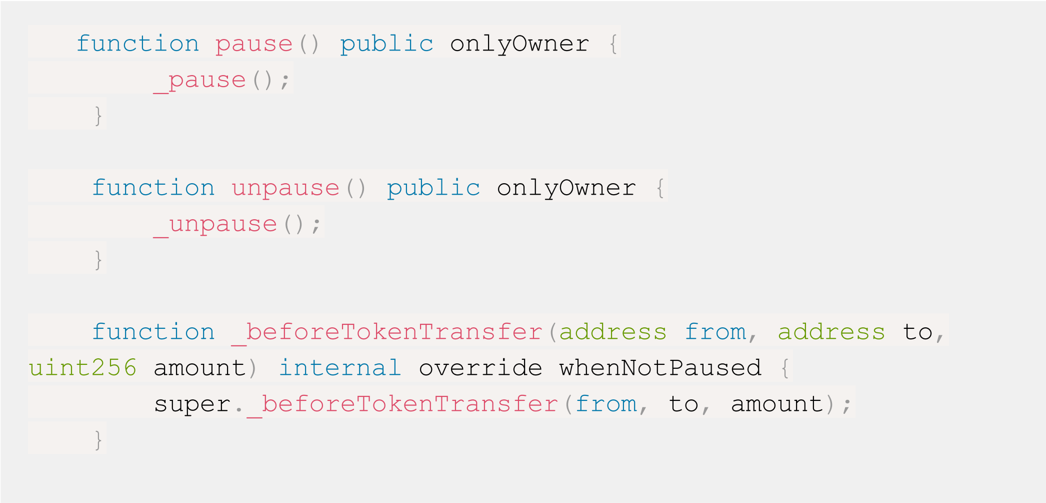
	Severity	Unresolved	Acknowledged	Resolved	Other
●	Critical	0	0	0	0
●	Medium	0	0	0	0
●	Minor / Informative	2	0	2	0

ST - Stops Transactions

Criticality	Minor / Informative
Location	contracts/UltimateTokenOwnable.sol#L35
Status	Renounced

Description

The contract owner has the authority to pause and unpause the transfers for all users including the owner. The owner may pause the transfers by calling the `pause` function and unpause them through the `unpaused` function.



```
    function pause() public onlyOwner {
        _pause();
    }

    function unpause() public onlyOwner {
        _unpause();
    }

    function _beforeTokenTransfer(address from, address to,
uint256 amount) internal override whenNotPaused {
        super._beforeTokenTransfer(from, to, amount);
    }
```

Recommendation

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions.

Temporary Solutions:

These measurements do not decrease the severity of the finding

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-signature wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

Permanent Solution:

- Renouncing the ownership, which will eliminate the threats but it is non-reversible.

Team Update

The contract's ownership has been renounced. The information regarding the transaction can be accessed through the following link: https://etherscan.io/tx/0xff3c88516a3019a0737ef3e36aac5b6bd9a243d8ad352f305dc240a7de061ff3.

MT - Mints Tokens

Criticality	Minor / Informative
Location	contracts/UltimateTokenOwnable.sol#L43
Status	Renounced

Description

The contract owner has the authority to mint tokens. The owner may take advantage of it by calling the `mint` function. As a result, the contract tokens will be highly inflated.

```
    function mint(address to, uint256 amount) public onlyOwner {
        _mint(to, amount);
    }
```

Recommendation

The team should carefully manage the private keys of the owner’s account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions.

Temporary Solutions:

These measurements do not decrease the severity of the finding

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-signature wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

Permanent Solution:

- Renouncing the ownership, which will eliminate the threats but it is non-reversible.

Team Update

The contract's ownership has been renounced. The information regarding the transaction can be accessed through the following link:

https://etherscan.io/tx/0xff3c88516a3019a0737ef3e36aac5b6bd9a243d8ad352f305dc240a7de061ff3.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	contracts/UltimateTokenOwnable.sol#L16,17,18,19,20,21
Status	Unresolved

Description

The Solidity style guide is a set of guidelines for writing clean and consistent Solidity code. Adhering to a style guide can help improve the readability and maintainability of the Solidity code, making it easier for others to understand and work with.

The followings are a few key points from the Solidity style guide:

1. Use camelCase for function and variable names, with the first letter in lowercase (e.g., myVariable, updateCounter).
2. Use PascalCase for contract, struct, and enum names, with the first letter in uppercase (e.g., MyContract, UserStruct, ErrorEnum).
3. Use uppercase for constant variables and enums (e.g., MAX_VALUE, ERROR_CODE).
4. Use indentation to improve readability and structure.
5. Use spaces between operators and after commas.
6. Use comments to explain the purpose and behavior of the code.
7. Keep lines short (around 120 characters) to improve readability.

```
address _owner
string memory _name
string memory _symbol
uint8 _decimals
uint256 _initialSupply
uint256 _maxSupply
```

Recommendation

By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.

Find more information on the Solidity documentation https://docs.soliditylang.org/en/v0.8.17/style-guide.html#naming-convention.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	contracts/UltimateTokenOwnable.sol#L3
Status	Unresolved

Description

The `^` symbol indicates that any version of Solidity that is compatible with the specified version (i.e., any version that is a higher minor or patch version) can be used to compile the contract. The version lock is a mechanism that allows the author to specify a minimum version of the Solidity compiler that must be used to compile the contract code. This is useful because it ensures that the contract will be compiled using a version of the compiler that is known to be compatible with the code.

```
pragma solidity ^0.8.19;
```

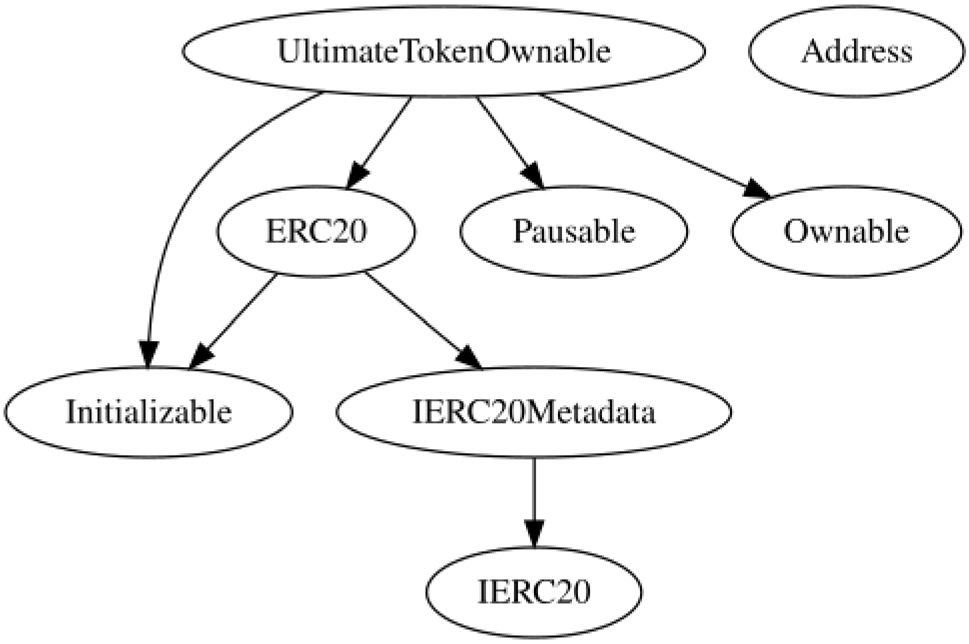
Recommendation

The team is advised to lock the pragma to ensure the stability of the codebase. The locked pragma version ensures that the contract will not be deployed with an unexpected version. An unexpected version may produce vulnerabilities and undiscovered bugs. The compiler should be configured to the lowest version that provides all the required functionality for the codebase. As a result, the project will be compiled in a well-tested LTS (Long Term Support) environment.

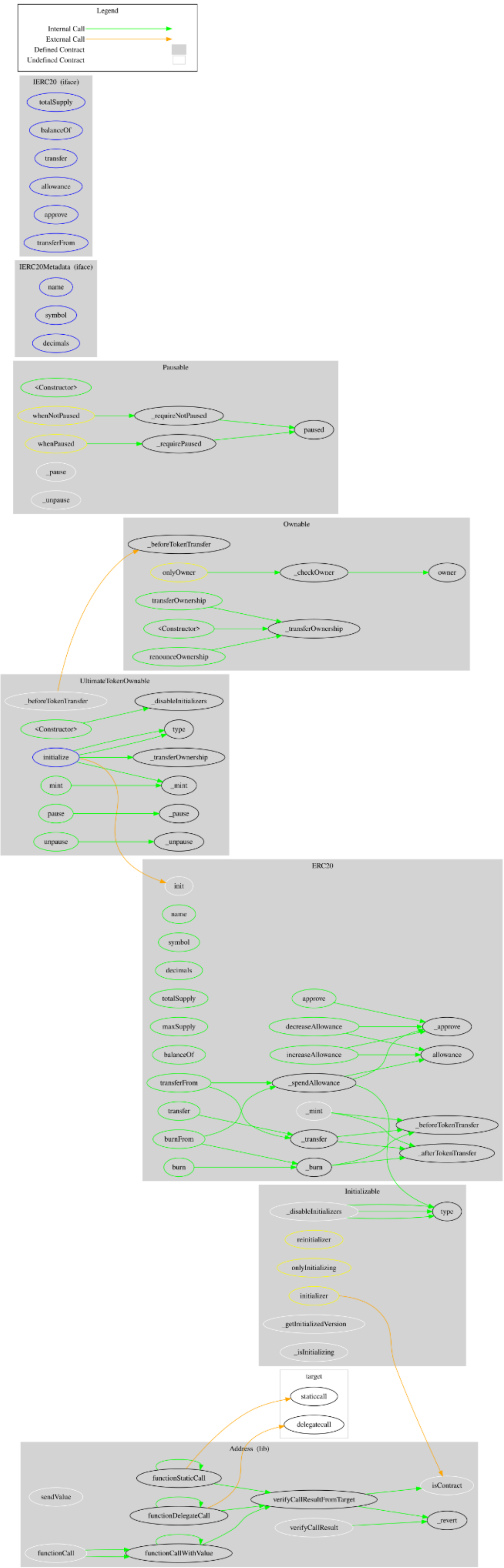
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
UltimateTokenOwnable	Implementation	Initializable, ERC20, Pausable, Ownable		
		Public	✓	-
	initialize	External	✓	initializer
	pause	Public	✓	onlyOwner
	unpause	Public	✓	onlyOwner
	mint	Public	✓	onlyOwner
	_beforeTokenTransfer	Internal	✓	whenNotPaused

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
IERC20Metadata (iface)
name
symbol
decimals
Pausable
<Constructor>
whenNotPaused
_requireNotPaused
paused
whenPaused
_requirePaused
_pause
_unpause
Ownable
_beforeTokenTransfer
onlyOwner
_checkOwner
owner
transferOwnership
<Constructor>
_transferOwnership
renounceOwnership
UltimateTokenOwnable
_beforeTokenTransfer
_disableInitializers
<Constructor>
type
initialize
_transferOwnership
mint
_mint
pause
_pause
unpause
_unpause
ERC20
init
name
symbol
decimals
totalSupply
approve
maxSupply
decreaseAllowance
_approve
balanceOf
increaseAllowance
allowance
transferFrom
_spendAllowance
transfer
_mint
_beforeTokenTransfer
burnFrom
_transfer
_afterTokenTransfer
burn
_burn
Initializable
_disableInitializers
type
reinitializer
onlyInitializing
initializer
getInitializedVersion
_isInitializing
target
staticcall
delegatecall
Address (lib)
functionStaticCall
isContract
sendValue
verifyCallResultFromTarget
functionDelegateCall
_revert
verifyCallResult
functionCall
functionCallWithValue

Summary

PAW contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. PAW is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions.

The contract's ownership has been renounced. The information regarding the transaction can be accessed through the following link:

https://etherscan.io/tx/0xff3c88516a3019a0737ef3e36aac5b6bd9a243d8ad352f305dc240a7de061ff3

Initial Audit, 3 Dec 2023

At the time of the audit report, the contract with address 0x419777D3E39AA9b00405724EaCE5ea57620c9062 is the minimal proxy contract for the following address: 0x3669c9E2467fef198FB7963A2dF887D0a005216C.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. 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 Cyberscope’s 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 regarded as an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Cyberscope to perform a security assessment. This document 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 represents an extensive assessment 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 Cyberscope’s position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io