Cyberscope

Audit Report

PawFury

November 2023

Network	ETH
Address	0x27f465E99725071c671c346E08BD99eF93567b8C
Audited by	© cyberscope

Analysis

Critical Medium Minor / Informative Pass

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

Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
	UPC	Unrestricted Pair Configuration	Unresolved
	L04	Conformance to Solidity Naming Conventions	Unresolved
	L16	Validate Variable Setters	Unresolved

Table of Contents

Review

Contract Name	PawFuryToken
Compiler Version	v0.8.21+commit.d9974bed
Optimization	200 runs
Explorer	https://etherscan.io/address/0x27f465e99725071c671c346e08bd99ef93567b8c
Address	0x27f465e99725071c671c346e08bd99ef93567b8c
Network	ETH
Symbol	PAW
Decimals	18
Total Supply	2,000,000,000

Audit Updates

Initial Audit	02 Nov 2023

Source Files

Filename	SHA256
contracts/PawfuryToken.sol	f67f6298def83346756d85fbefd17f57aebcdb0c7800a224546ca2ceecc77bd1
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a
@openzeppelin/contracts/token/ERC20/IERC20.sol	94f23e4af51a18c2269b355b8c7cf4db8003d075c9c541019eb8dcf4122864d5
@openzeppelin/contracts/token/ERC20/ERC20.sol	bce14c3fd3b1a668529e375f6b70ffdf9cef8c4e410ae99608be5964d98fa701
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	af5c8a77965cc82c33b7ff844deb9826166689e55dc037a7f2f790d057811990
@openzeppelin/contracts/token/ERC20/extensions/ERC20Burnable.sol	0344809a1044e11ece2401b4f7288f414ea41fa9d1dad24143c84b737c9fc02e
@openzeppelin/contracts/access/Ownable.sol	9353af89436556f7ba8abb3f37a6677249aa4df6024fbfaa94f79ab2f44f3231

Findings Breakdown

Critical	1
Medium	0
Minor / Informative	3

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	1	0	0	0
Medium	0	0	0	0
Minor / Informative	3	0	0	0

ST - Stops Transactions

Criticality	Critical
Location	contracts/PawfuryToken.sol#L59
Status	Unresolved

Description

The transactions are initially disabled for all users excluding the authorized addresses. The owner can enable the transactions for all users. Once the transactions are enabled the owner will not be able to disable them again.

```
if (!tradingEnabled) {
    if (_pairs[from] || _pairs[to]) {
        _pairs[from]
            ? require(
                to == owner() || to == _launcher,
                "MK: trading disabled"
            )
            : require(
                from == owner() || from == _launcher,
                "MK: trading disabled"
            );
    }
}
```

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. Some suggestions are:

- Introduce a multi-sign wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

UPC - Unrestricted Pair Configuration

Criticality	Minor / Informative
Location	contracts/PawfuryToken.sol#L39
Status	Unresolved

Description

The `setPairs` function allows the contract owner to set multiple pair addresses to any arbitrary value without validation. This lack of validation can lead to unintended behavior, including the potential disruption of the contract's transfer flow as long as the transactions are disabled.

```
function setPairs(
    address[] calldata pairs,
    bool[] calldata status
) external onlyOwner {
    require(!tradingEnabled, "MK: trading already enabled");
    require(pairs.length == status.length, "MK: invalid parameters");
    for (uint256 i = 0; i < pairs.length; i++) {
        _pairs[pairs[i]] = status[i];
    }
    emit PairsUpdated();
}
```

Recommendation

To enhance security and prevent the contract owner from setting arbitrary values as pair addresses, the team is advised to implement proper validation checks in the `setPairs` function. These checks should include verifying that the provided addresses are valid before allowing it to be set. This will help ensure the integrity of the contract's operation.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	contracts/PawfuryToken.sol#L9
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 private constant _router =
        0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D
```

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.

L16 - Validate Variable Setters

Criticality	Minor / Informative
Location	contracts/PawfuryToken.sol#L35
Status	Unresolved

Description

The contract performs operations on variables that have been configured on user-supplied input. These variables are missing of proper check for the case where a value is zero. This can lead to problems when the contract is executed, as certain actions may not be properly handled when the value is zero.

```
_launcher = launcher
```

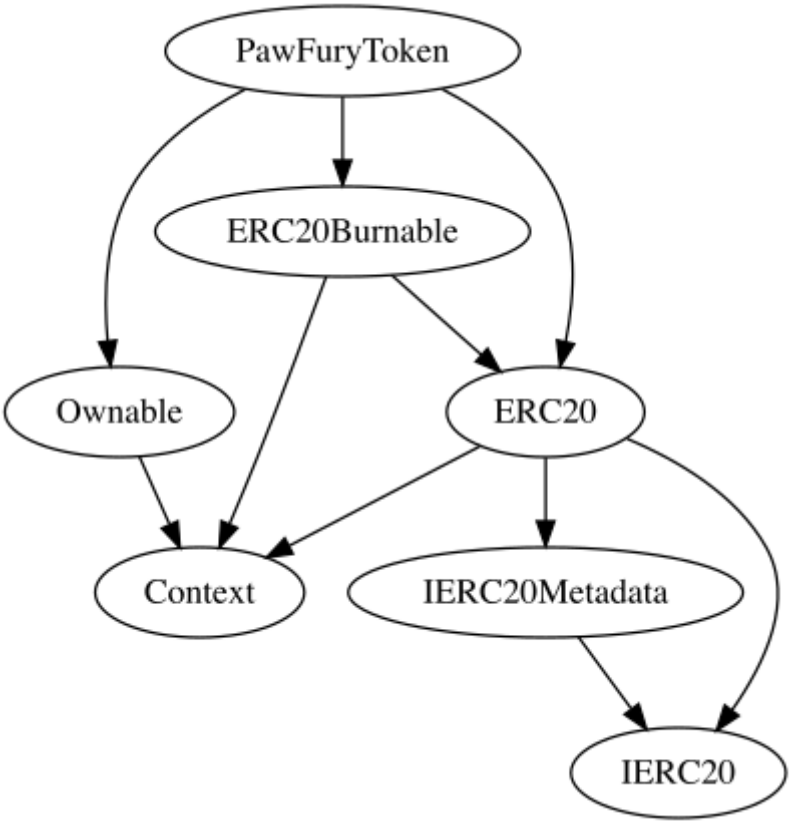
Recommendation

By adding the proper check, the contract will not allow the variables to be configured with zero value. This will ensure that the contract can handle all possible input values and avoid unexpected behavior or errors. Hence, it can help to prevent the contract from being exploited or operating unexpectedly.

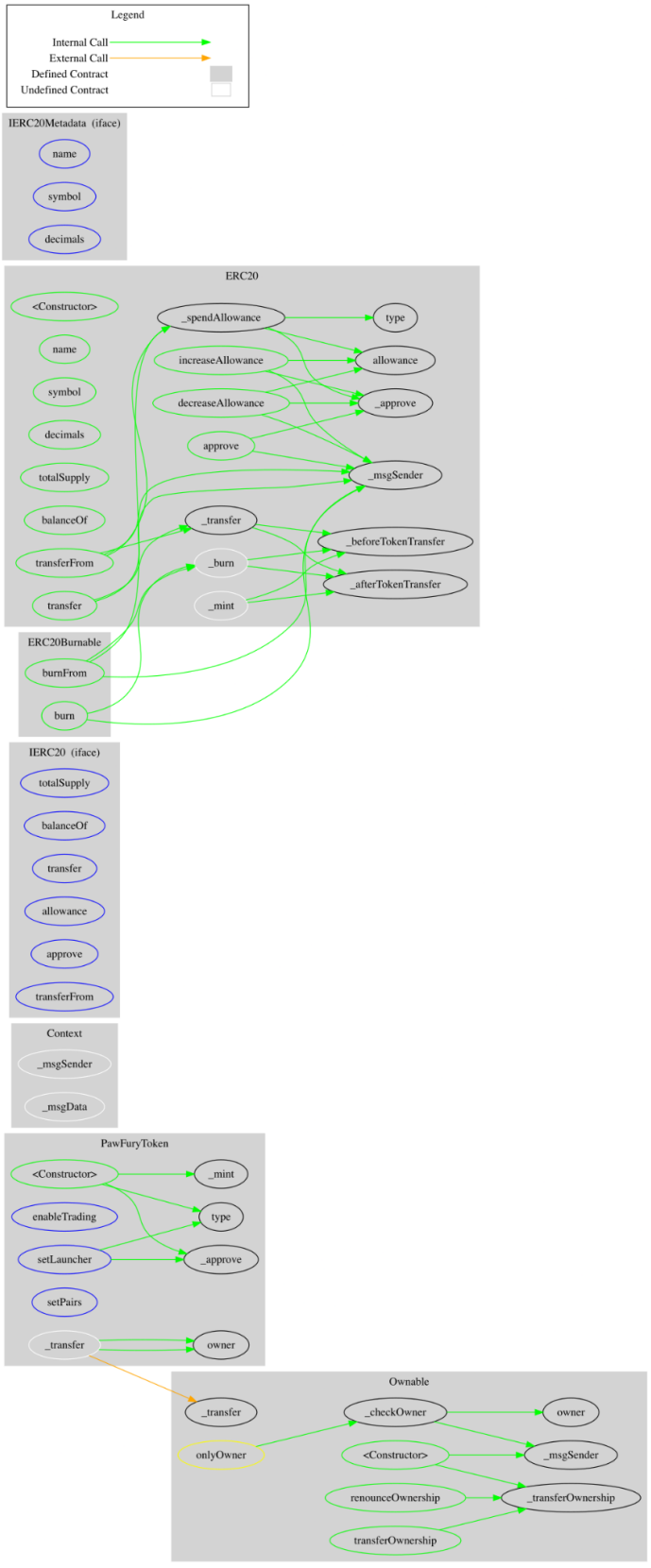
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
PawFuryToken	Implementation	ERC20, ERC20Burnable, Ownable		
		Public	✓	ERC20
	enableTrading	External	✓	onlyOwner
	setLauncher	External	✓	onlyOwner
	setPairs	External	✓	onlyOwner
	_transfer	Internal	✓	

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
IERC20Metadata (iface)
name
symbol
decimals
ERC20
<Constructor>
name
symbol
decimals
totalSupply
balanceOf
transferFrom
transfer
_spendAllowance
increaseAllowance
decreaseAllowance
approve
_transfer
_burn
_mint
type
allowance
_approve
_msgSender
_beforeTokenTransfer
_afterTokenTransfer
ERC20Burnable
burnFrom
burn
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
Context
_msgSender
_msgData
PawFuryToken
<Constructor>
enableTrading
setLauncher
setPairs
_transfer
_mint
type
_approve
owner
Ownable
_transfer
onlyOwner
_checkOwner
<Constructor>
renounceOwnership
transferOwnership
owner
_msgSender
_transferOwnership

Summary

PawFury contract implements a token mechanism. This audit investigates security issues, business logic concerns, and potential improvements. There are some functions that can be abused by the owner like stopping transactions. A multi-wallet signing pattern will provide security against potential hacks.

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