Cyberscope

Audit Report

X Capital

May 2022

Github	https://github.com/XCapital-0510/XCapital/blob/main/XCAP.sol
Commit	3aff166e38cd6aaa0d8f874bc804f62e0e648607
Audited by	© cyberscope

Table of Contents

Table of Contents 1
Contract Review 3
Source Files 3
Audit Updates 3
Contract Analysis 4
ELFM - Exceed Limit Fees Manipulation 5
Description 5
Recommendation 6
Contract Diagnostics 7
MEE - Misleading Event Emission 8
Description 8
Recommendation 8
STC - Succeeded Transfer Check 9
Description 9
Recommendation 9
MC - Missing Check (1/2) 10
Description 10
Recommendation 10
MC - Missing Check (2/2) 11
Description 11
Recommendation 11
L01 - Public Function could be Declared External 12
Description 12
Recommendation 12
L02 - State Variables could be Declared Constant 13
Description 13

Recommendation 13
L04 - Conformance to Solidity Naming Conventions 14
Description 14
Recommendation 14
L09 - Dead Code Elimination 15
Description 15
Recommendation 15
L11 - Unnecessary Boolean equality 16
Description 16
Recommendation 16
L14 - Uninitialized Variables in Local Scope 17
Description 17
Recommendation 17
Contract Functions 18
Contract Flow 24
Domain Info 25
Summary 26
Disclaimer 27
About Cyberscope 28

Contract Review

Contract Name	XToken
Compiler Version	v0.6.8+commit.0bbfe453
Optimization	200 runs
Licence	
Testing Deploy	https://bscscan.com/token/0x50203342f8cD125791d27b2902e3922F67B50d8C
Symbol	XCAP
Decimals	9
Total Supply	10,000,000,000
Domain	xcap.finance
Github	https://github.com/XCapital-0510/XCapital/blob/main/XCAP.sol
Commit	3aff166e38cd6aaa0d8f874bc804f62e0e648607

Source Files

Filename	**SHA256**
contract.sol	3f69e4d5a49572c930a9fd4a4cc07ce4d560cd83b311e408a56e6475bc03af15

Audit Updates

Initial Audit	31st May 2022
Corrected	

Contract Analysis

Critical Medium Minor Pass

Severity	Code	Description
●	ST	Contract Owner is not able to stop or pause transactions
●	OCTD	Contract Owner is not able to transfer tokens from specific address
●	OTUT	Owner Transfer User's Tokens
●	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
●	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
●	MT	Contract Owner is not able to mint new tokens
●	BT	Contract Owner is not able to burn tokens from specific wallet
●	BC	Contract Owner is not able to blacklist wallets from selling

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#L1000, 1010, 1020, 1029

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by calling the `setBuyXFee` , `setSellXFee` , `setBuyNotXFee` and `setSellNotXFee` functions with a high percentage value.

```
function setBuyXFee(uint256 X, uint256 Y, uint256 Z)
        public
    {
        require(tx.origin == owner());
        require(X+Y+Z < 100);
        X0 = X;
        Y0 = Y;
        Z0 = Z;
    }
```

```
function setSellXFee(uint256 X, uint256 Y, uint256 Z)
        public
    {
        require(tx.origin == owner());
        require(X+Y+Z < 100);
        X1 = X;
        Y1 = Y;
        Z1 = Z;
    }
```

```
function setBuyNotXFee(uint256 X, uint256 Y)
        public
    {
        require(tx.origin == owner());
        require(X+Y < 100);
        X2 = X;
        Y2 = Y;
    }
```

```
function setSellNotXFee(uint256 X, uint256 Y)
        public
    {
        require(tx.origin == owner());
        require(X+Y < 100);
        X3 = X;
        Y3 = Y;
    }
```

Recommendation

The contract could embody a check for the maximum acceptable value to be less than or equal to 25%.

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. That risk can be prevented by temporarily locking the contract or renouncing ownership.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	MEE	Misleading Event Emission
	STC	Succeeded Transfer Check
	MC	Missing Check (1/2)
	MC	Missing Check (2/2)
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L09	Dead Code Elimination
	L11	Unnecessary Boolean equality
	L14	Uninitialized Variables in Local Scope

MEE - Misleading Event Emission

Criticality	minor
Location	contract.sol#L790

Description

The construtor emits a token transfer event to the contract deployer. This event is not happening. The tokens are moved to the contract owner once in the `setInitialToken` method.

```
emit Transfer(address(0), _msgSender(), _tTotal);
```

Recommendation

The event emission should be moved to the `setInitialToken` method.

STC - Succeeded Transfer Check

Criticality	minor
Location	contract.sol

Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
IBEP20(usdt).transfer(address(rewardPool), X);
```

Recommendation

The contract should check if the result of the transfer methods is successful.

MC - Missing Check (1/2)

Criticality	minor
Location	contract.sol#L910

Description

The contract is processing variables that have not properly sanitized and checked that they form the proper shape. These variables may produce vulnerability issues.

The `nftAddr` could be any address. The setter method `setMintNft()` should implement some typical checks like zero value equality.

```
OriginNFT(nftAddr).mint(accountPair[to], XId1);
```

Recommendation

The contract should properly check the variables according to the required specifications

MC - Missing Check (2/2)

Criticality	medium
Location	contract.sol#L1048

Description

The contract is processing variables that have not properly sanitized and checked that they form the proper shape. These variables may produce vulnerability issues.

The swapTokensForX enables the swapUnlock but it does not disable. Additionally, the swap does not check if it is already in swap. The `swapTokensForX` calls the `initialAccount` that calls the `OriginNFT(nftAddr).mint(accountPair[to], XId1);` if the `external OriginNFT(nftAddr).mint(...)` calls the swap, then a re-enter attack may caused.

```
function swapTokensForX(uint256 _amount, address tokenAddress)
    public
    payable
{
    swapUnlock = true;
    initialAccount(baseAccount, msg.sender);
```

Recommendation

The contract should prevent the swaps if the swap flag is enabled. The contract should also keep the swap flag updated.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L463,472,478,483,491,793,797,801,805,809,813,818,822,827,833,838,941,976,990,1000,1010,1020,1029,1048,1107,1165,1173,1181,1189,1208,1214,1237

Description

Public functions that are never called by the contract should be declared external to save gas.

```
getUSDT
removeWhiteList
addWhiteList
getSellNotXFee
getBuyNotXFee
getSellXFee
getBuyXFee
swapXForToken
swapTokensForX
...
```

Recommendation

Use the external attribute for functions never called from the contract.

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L737,735,736,733

Description

Constant state variables should be declared constant to save gas.

```
_tTotal
_symbol
_name
_decimals
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L131,133,530,531,548,568,920,931,942,943,944,945,946,977,978,979,1000,1010,1020,1029,1038,1048,760,761,762,763,764,765,767,768,769,770,773,774

Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
XId1
XId0
Y3
X3
Y2
X2
Z1
Y1
X1
...
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L401,361,371,386,396,308,335,704

Description

Functions that are not used in the contract, and make the code's size bigger.

```
random
sendValue
isContract
functionCallWithValue
functionCall
_functionCallWithValue
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L853

Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
whiteListActive[from] == true
whiteListActive[to] == true
```

Recommendation

Remove the equality to the boolean constant.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L1216

Description

The are variables that are defined in the local scope and are not initialized.

```
i
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IBEP20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC165	Interface			
	supportsInterface	External		-
IERC721	Interface	IERC165		
	balanceOf	External		-
	ownerOf	External		-
	safeTransferFrom	External	✓	-
	transferFrom	External	✓	-
	approve	External	✓	-
	getApproved	External		-
	setApprovalForAll	External	✓	-
	isApprovedForAll	External		-
	safeTransferFrom	External	✓	-
IERC721Metadata	Interface	IERC721		
	name	External		-
	symbol	External		-
	tokenURI	External		-

IERC721Enumerable	Interface	IERC721		
	totalSupply	External		-
	tokenOfOwnerByIndex	External		-
	tokenByIndex	External		-
IERC721Receiver	Interface			
	onERC721Received	External	✓	-
OriginNFT	Interface			
	XIds	External		-
	mint	External	✓	-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	_functionCallWithValue	Private	✓	

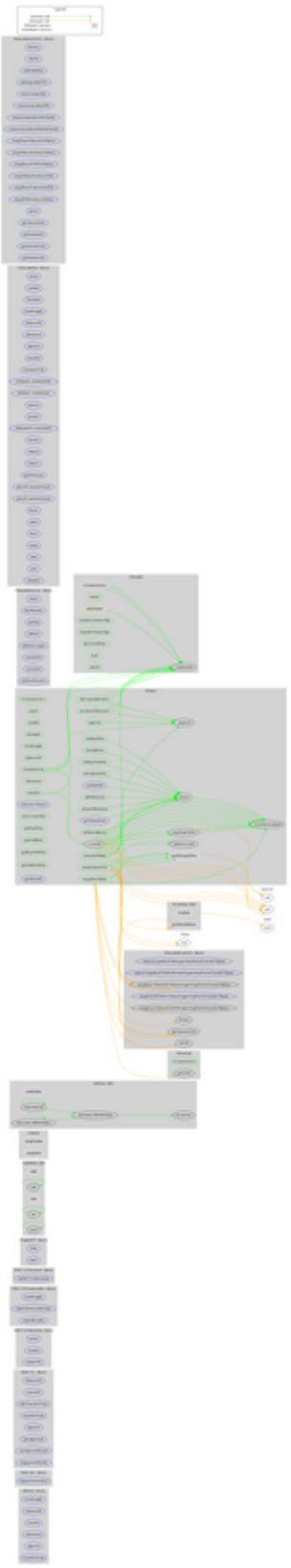
Ownable	Implementation	Context		
	<Constructor>	Internal	✓	
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	geUnlockTime	Public		-
	lock	Public	✓	onlyOwner
	unlock	Public	✓	-
IPancakeFactory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IPancakePair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-

	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IPancakeRouter01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-

	getAmountsIn	External		-
IPancakeRouter02	Interface	IPancakeRouter01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
XCommon	Library			
	random	Internal		
	getPairAddress	Internal		
XToken	Implementation	Context, IBEP20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	<Receive Ether>	External	Payable	-
	_approve	Private	✓	
	_transfer	Private	✓	
	initialAccount	Internal	✓	

	_transferStandard	Private	✓	
	payXSwapTxFee	Internal	✓	
	paySwapTxFee	Internal	✓	
	setInitialToken	Public	✓	-
	getTokenBack	External	✓	-
	setKeyAddress	Public	✓	-
	setAccountPair	Public	✓	-
	setBuyXFee	Public	✓	-
	setSellXFee	Public	✓	-
	setBuyNotXFee	Public	✓	-
	setSellNotXFee	Public	✓	-
	setMintNft	External	✓	-
	swapTokensForX	Public	Payable	-
	swapXForToken	Public	✓	-
	getBuyXFee	Public		-
	getSellXFee	Public		-
	getBuyNotXFee	Public		-
	getSellNotXFee	Public		-
	getMintNft	External		-
	addWhiteList	Public	✓	-
	removeWhiteList	Public	✓	-
XInternal	Implementation			
	<Constructor>	Public	✓	-
	getUSDT	Public	✓	-

Contract Flow



Domain Info

Domain Name	xcap.finance
Registry Domain ID	63f6bd3e84e64fe1b6d13524713136c0-DONUTS
Creation Date	2022-03-18T02:02:15Z
Updated Date	2022-04-15T22:58:35Z
Registry Expiry Date	2024-03-18T02:02:15Z
Registrar WHOIS Server	whois.godaddy.com/
Registrar URL	http://www.godaddy.com/domains/search.aspx?ci=8990
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain has been created 2 months before the creation of the audit. It will expire in almost 2 years.

There is no public billing information, the creator is protected by the privacy settings.

Summary

The contract owner has the authority to manipulate the fees. The maximum fee percentage that can be set is 99%. The fees manipulation varies between sales and buys. If the 99% fees are configured only to the sales, then the contract will behave similar to a **honeypot**. Additionally, the contract contains some security issues. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io