



Cyberscope

Audit Report

Stealth Pad XYZ

August 2023

Network ETH

Address 0xb18f98822c22492bd6b77d19cae9367f3d60fcbf

Audited by © cyberscope

Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	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
●	PVC	Price Volatility Concern	Unresolved
●	RSML	Redundant SafeMath Library	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L06	Missing Events Access Control	Unresolved
●	L07	Missing Events Arithmetic	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved
●	L16	Validate Variable Setters	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	5
PVC - Price Volatility Concern	6
Description	6
Recommendation	6
RSML - Redundant SafeMath Library	7
Description	7
Recommendation	7
L02 - State Variables could be Declared Constant	8
Description	8
Recommendation	8
L04 - Conformance to Solidity Naming Conventions	9
Description	9
Recommendation	10
L06 - Missing Events Access Control	11
Description	11
Recommendation	11
L07 - Missing Events Arithmetic	12
Description	12
Recommendation	12
L14 - Uninitialized Variables in Local Scope	13
Description	13
Recommendation	13
L16 - Validate Variable Setters	14
Description	14
Recommendation	14
L19 - Stable Compiler Version	15
Description	15
Recommendation	15
Functions Analysis	16
Inheritance Graph	19
Flow Graph	20
Summary	21
Disclaimer	22

Review

Explorer	https://etherscan.io/address/0xb18f98822c22492bd6b77d19cae9367f3d60fcfb
Symbol	STEALTH
Decimals	8
Total Supply	1,000,000,000

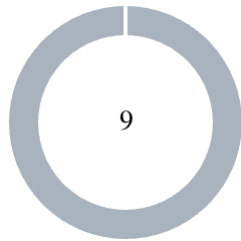
Audit Updates

Initial Audit	26 Aug 2023
---------------	-------------

Source Files

Filename	SHA256
StealthPad.sol	e48720dbfff3489b3ad076fcacc88d3c3b7e98856e091360ce058f824647dcb0

Findings Breakdown



Critical	0
Medium	0
Minor / Informative	9

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	0	0	0	0
Medium	0	0	0	0
Minor / Informative	9	0	0	0

PVC - Price Volatility Concern

Criticality	Minor / Informative
Location	StealthPad.sol#L414
Status	Unresolved

Description

The contract accumulates tokens from the taxes to swap them for ETH. The variable `swapThreshold` sets a threshold where the contract will trigger the swap functionality. If the variable is set to a big number, then the contract will swap a huge amount of tokens for ETH.

It is important to note that the price of the token representing it, can be highly volatile. This means that the value of a price volatility swap involving Ether could fluctuate significantly at the triggered point, potentially leading to significant price volatility for the parties involved.

```
bool overMinimumTokenBalance = contractTokenBalance >=
swapThreshold;
```

Recommendation

The contract could ensure that it will not sell more than a reasonable amount of tokens in a single transaction. A suggested implementation could check that the maximum amount should be less than a fixed percentage of the total supply. Hence, the contract will guarantee that it cannot accumulate a huge amount of tokens in order to sell them.

RSML - Redundant SafeMath Library

Criticality	Minor / Informative
Location	StealthPad.sol
Status	Unresolved

Description

SafeMath is a popular Solidity library that provides a set of functions for performing common arithmetic operations in a way that is resistant to integer overflows and underflows.

Starting with Solidity versions that are greater than or equal to 0.8.0, the arithmetic operations revert to underflow and overflow. As a result, the native functionality of the Solidity operations replaces the SafeMath library. Hence, the usage of the SafeMath library adds complexity, overhead and increases gas consumption unnecessarily.

```
library SafeMath {...}
```

Recommendation

The team is advised to remove the SafeMath library. Since the version of the contract is greater than `0.8.0` then the pure Solidity arithmetic operations produce the same result.

If the previous functionality is required, then the contract could exploit the `unchecked { ... }` statement.

Read more about the breaking change on

<https://docs.soliditylang.org/en/v0.8.16/080-breaking-changes.html#solidity-v0-8-0-breaking-changes>.

L02 - State Variables could be Declared Constant

Criticality	Minor / Informative
Location	StealthPad.sol#L221,222,223,243
Status	Unresolved

Description

State variables can be declared as constant using the constant keyword. This means that the value of the state variable cannot be changed after it has been set. Additionally, the constant variables decrease gas consumption of the corresponding transaction.

```
string private _name = "StealthPad"  
string private _symbol = "STEALTH"  
uint8 private _decimals = 8  
uint256 feedenominator = 100
```

Recommendation

Constant state variables can be useful when the contract wants to ensure that the value of a state variable cannot be changed by any function in the contract. This can be useful for storing values that are important to the contract's behavior, such as the contract's address or the maximum number of times a certain function can be called. The team is advised to add the constant keyword to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	StealthPad.sol#L162,163,179,198,228,229,230,232,233,234,236,237,245,255,256,267,539,619,624,632,644,648,652,656,660,672,676,680,684,691,700,704
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.

```
function DOMAIN_SEPARATOR() external view returns (bytes32);
function PERMIT_TYPEHASH() external pure returns (bytes32);
function MINIMUM_LIQUIDITY() external pure returns (uint);
function WETH() external pure returns (address);
uint _buydevelopmentTax = 2
uint _buyLpTax = 1
uint _buyRewardTax = 1
uint _selldevelopmentTax = 2
uint _sellLpTax = 1
uint _sellRewardTax = 1
uint256 public _totalbuyFee =
_buydevelopmentTax.add(_buyLpTax).add(_buyRewardTax)
uint256 public _totalSellFee =
_selldevelopmentTax.add(_sellLpTax).add(_sellRewardTax)
mapping (address => uint256) _balances
uint256 public _maxTxAmount = _totalSupply.mul(1).div(100)

...
```

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>.

L06 - Missing Events Access Control

Criticality	Minor / Informative
Location	StealthPad.sol#L677
Status	Unresolved

Description

Events are a way to record and log information about changes or actions that occur within a contract. They are often used to notify external parties or clients about events that have occurred within the contract, such as the transfer of tokens or the completion of a task. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
lpReceiverWallet = _newWallet
```

Recommendation

To avoid this issue, it's important to carefully design and implement the events in a contract, and to ensure that all required events are included. It's also a good idea to test the contract to ensure that all events are being properly triggered and logged.

By including all required events in the contract and thoroughly testing the contract's functionality, the contract ensures that it performs as intended and does not have any missing events that could cause issues.

L07 - Missing Events Arithmetic

Criticality	Minor / Informative
Location	StealthPad.sol#L625,633,665,669,697
Status	Unresolved

Description

Events are a way to record and log information about changes or actions that occur within a contract. They are often used to notify external parties or clients about events that have occurred within the contract, such as the transfer of tokens or the completion of a task.

It's important to carefully design and implement the events in a contract, and to ensure that all required events are included. It's also a good idea to test the contract to ensure that all events are being properly triggered and logged.

```
_buydevelopmentTax = _developmentFee  
_selldevelopmentTax = _developmentFee  
_walletMax = newLimit  
_maxTxAmount = newLimit  
swapThreshold = _threshold
```

Recommendation

By including all required events in the contract and thoroughly testing the contract's functionality, the contract ensures that it performs as intended and does not have any missing events that could cause issues with its arithmetic.

L14 - Uninitialized Variables in Local Scope

Criticality	Minor / Informative
Location	StealthPad.sol#L465
Status	Unresolved

Description

Using an uninitialized local variable can lead to unpredictable behavior and potentially cause errors in the contract. It's important to always initialize local variables with appropriate values before using them.

```
uint feeAmount
```

Recommendation

By initializing local variables before using them, the contract ensures that the functions behave as expected and avoid potential issues.

L16 - Validate Variable Setters

Criticality	Minor / Informative
Location	StealthPad.sol#L620,673,677,681,705
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.

```
(bool success, ) =  
address(_token).call(abi.encodeWithSignature('transfer(address,uint256)',  
recipient, _amount))  
developmentWallet = _newWallet  
lpReceiverWallet = _newWallet  
rewardWallet = _newWallet  
dexPair = _pair
```

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.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	StealthPad.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 ^0.8.10;
```

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.

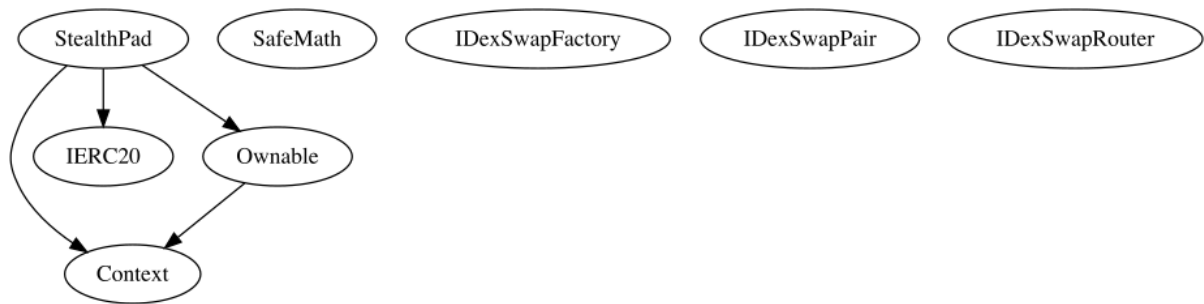
Functions Analysis

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
StealthPad	Implementation	Context, IERC20, Ownable		
		Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	allowance	Public		-
	getCirculatingSupply	Public		-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	approve	Public	✓	-
	_approve	Private	✓	
		External	Payable	-
	transfer	Public	✓	-
	transferFrom	Public	✓	-
	_transfer	Private	✓	
	_basicTransfer	Internal	✓	

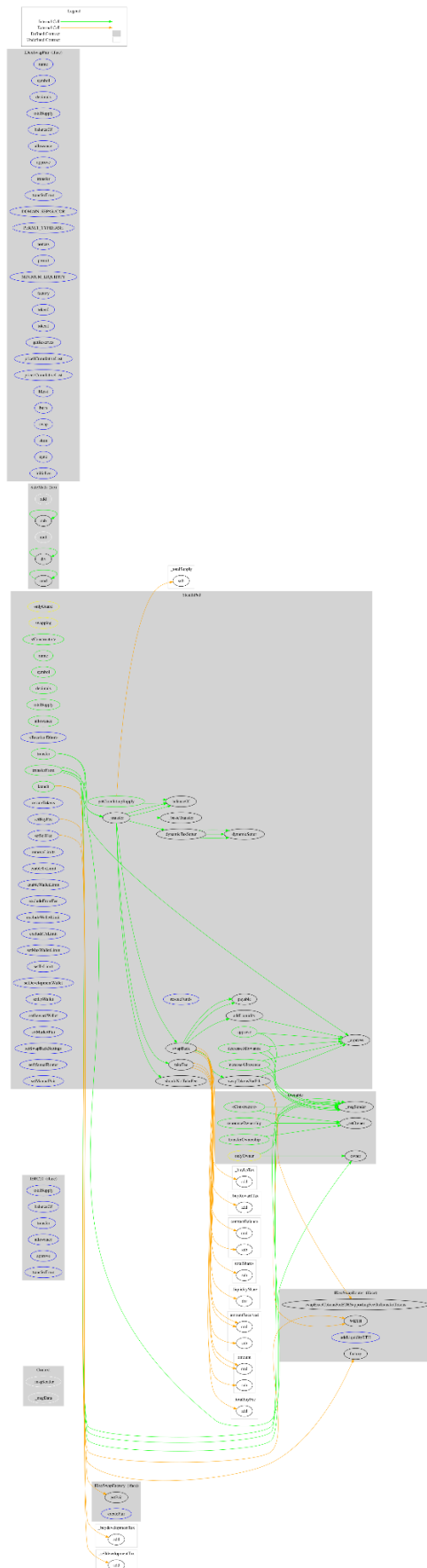
	shouldNotTakeFee	Internal		
	takeFee	Internal	✓	
	launch	Public	Payable	onlyOwner
	dynamicTaxSetter	Internal	✓	
	dynamicSetter	Internal	✓	
	swapBack	Internal	✓	swapping
	addLiquidity	Private	✓	
	swapTokensForEth	Private	✓	
	rescueFunds	External	✓	onlyGuard
	rescueTokens	External	✓	onlyGuard
	setBuyFee	External	✓	onlyOwner
	setSellFee	External	✓	onlyOwner
	removeLimits	External	✓	onlyGuard
	enableTxLimit	External	✓	onlyOwner
	enableWalletLimit	External	✓	onlyOwner
	excludeFromFee	External	✓	onlyOwner
	excludeWalletLimit	External	✓	onlyOwner
	excludeTxLimit	External	✓	onlyOwner
	setMaxWalletLimit	External	✓	onlyOwner
	setTxLimit	External	✓	onlyOwner
	setDevelopmentWallet	External	✓	onlyOwner
	setLpWallet	External	✓	onlyOwner
	setRewardWallet	External	✓	onlyOwner

	setMarketPair	External	✓	onlyOwner
	setSwapBackSettings	External	✓	onlyGuard
	setManualRouter	External	✓	onlyOwner
	setManualPair	External	✓	onlyOwner

Inheritance Graph



Flow Graph



Summary

stealthpadxyz contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. stealthpadxyz is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error or critical issues. The Guard role can access some admin functions that can not be used in a malicious way to disturb the users' transactions. The fees are fixed to 4%.

The contract ownership has been renounced:

<https://etherscan.io/tx/0xe049f633b5726c8198070e9d17432a98789117168a129a77cb77b881bec95be3>

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>