Cyberscope

Audit Report

Gasclick

April 2023

Github	https://github.com/GasClick/CryptoGas
Commit	f43bc59c1a55b86b55a31fd8f8f50ce44fe8c29a
Audited by	© cyberscope

Table of Contents

Table of Contents **1**
Review **3**
Audit Updates 3
Source Files 3
Introduction **4**
Roles 4
GasClickICO Contract 4
Owner 4
User 5
GasClickAntiWhale Contract 6
Owner 6
User 6
Findings Breakdown **7**
Diagnostics **8**
NCF - Non-Guaranteed Crowdsale Funds 9
Description 9
Recommendation 9
Team Update 10
MVM - Missing Variable Modification 11
Description 11
Recommendation 11
Team Update 11
MU - Modifiers Usage 12
Description 12
Recommendation 12
Team Update 12
MC - Missing Check 13
Description 13
Recommendation 13
Team Update 14
SCAR - Stops Claims And Refunds 15
Description 15
Recommendation 15
Team Update 16
DPI - Decimals Precision Inconsistency 17
Description 17
Recommendation 18
Team Update 18
Functions Analysis **19**

Inheritance Graph **22**
Flow Graph **23**
Summary **24**
Disclaimer **25**
About Cyberscope **26**

Review

Repository	https://github.com/GasClick/CryptoGas
Commit	f43bc59c1a55b86b55a31fd8f8f50ce44fe8c29a

Audit Updates

Initial Audit	05 Apr 2023 https://github.com/cyberscope-io/audits/blob/main/cygas/v1/audit.pdf
Corrected Phase 2	13 Apr 2023 https://github.com/cyberscope-io/audits/blob/main/cygas/v2/audit.pdf
Corrected Phase 3	19 Apr 2023

Source Files

Filename	SHA256
GasClickAntiWhale.sol	c391e1a37af3f64732bb74b566f18db9a2e16b381fcac95c2ca16af084972690
GasClickICO.sol	1ab03a4a08243943fb62a774e2a12dc0b46f6e1dbc4fb7f7c73715a5317e1673

Introduction

The GasClick ecosystem comprises of two contracts, the GasClickICO, and the GasClickAntiWhale contract. This ecosystem facilitates users to deposit and retrieve funds for an Initial Coin Offering (ICO). The GasClickICO contract incorporates several features such as setting the crowdsale stage, updating soft and hard cap limits, defining payment tokens, and accessing information about investors and their contributions. The contract includes anti-whale measures to prevent large investments from overpowering the sale. The ICO tokens are sold at a fixed rate of 0.03 USD each with a choice of payment currencies. The USD price of these payment currencies is dynamically obtained from an Oracle per request. However, this Oracle external call can be bypassed and use, instead, the last known price for this payment currency. Furthermore, the contract includes protections against reentrancy attacks.

Roles

GasClickICO Contract

Owner

The owner has authority over the following functions:

- `function setCrowdsaleStage(uint stage_)`
- `function setHardCapuUSD(uint256 hardCap)`
- `function setSoftCapuUSD(uint256 softCap)`
- `function setDynamicPrice(bool dynPrice)`
- `function setPaymentToken(string calldata symbol, address tokenAdd, address priceFeed, uint256 uUSDPerTokens, uint8 decimals)`
- `function deletePaymentToken(string calldata symbol, uint8 index)`
- `function refundAddress(string calldata symbol, address investor)`
- `function claimAddress(address investor)`
- `function setTokenAddress(address payable add)`
- `function withdraw(string calldata symbol, uint8 percentage)`
- `function setTargetWalletAddress(address payable add)`

User

The user can interact with the following functions:

- `function getCrowdsaleStage()`
- `function getTotaluUSDInvested()`
- `function getHardCap()`
- `function getSoftCap()`
- `function getPriceuUSD()`
- `function gettDynamicPrice()`
- `function getPaymentSymbols()`
- `function getPaymentToken(string calldata symbol)`
- `function getUusdPerToken(string calldata symbol)`
- `function getInvestors()`
- `function getInvestorsCount()`
- `function getContribution(address investor, string calldata symbol)`
- `function getuUSDContribution(address investor, string calldata symbol)`
- `function getuUSDToClaim(address investor)`
- `function depositTokens(string calldata symbol, uint256 rawAmountWitDecimals)`
- `function refund(string calldata symbol)`
- `function claim()`
- `function getTokenAddress()`
- `function getTargetWalletAddress()`

GasClickAntiWhale Contract

Owner

The owner has authority over the following functions:

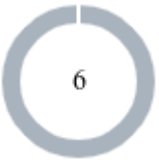
- `function setWhitelistuUSDThreshold(uint256 whitelistuUSDThreshold_)`
- `function whitelistUser(address user)`
- `function unwhitelistUser(address user)`
- `function setUseBlacklist(bool useBlacklist_)`
- `function blacklistUser(address user)`
- `function unblacklistUser(address user)`
- `function setExcludedFromMaxInvestment(address account, bool exclude)`
- `function setMaxuUSDInvestment(uint256 maxuUSDInvestment_)`
- `function setExcludedFromMaxTransfer(address account, bool exclude)`
- `function setMaxuUSDTransfer(uint256 maxuUSDTransfer_)`
- `function setMinuUSDTransfer(uint256 minuUSDTransfer_)`

User

The user can interact with the following functions:

- `function getWhitelistuUSDThreshold()`
- `function getWhitelisted()`
- `function getWhitelistUserCount()`
- `function isWhitelisted(address user)`
- `function getUseBlacklist()`
- `function getBlacklisted()`
- `function getBlacklistUserCount()`
- `function isBlacklisted(address user)`
- `function isExcludedFromMaxInvestment(address acc)`
- `function getMaxUSDInvestment()`
- `function isExcludedFromMaxTransfer(address acc)`
- `function getMaxUSDTransfer()`
- `function getMinUSDTransfer()`

Findings Breakdown



Critical	0
Medium	0
Minor / Informative	6

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	0	0	0	0
Medium	0	0	0	0
Minor / Informative	0	6	0	0

Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
●	NCF	Non-Guaranteed Crowdsale Funds	Acknowledged
●	MVM	Missing Variable Modification	Acknowledged
●	MU	Modifiers Usage	Acknowledged
●	MC	Missing Check	Acknowledged
●	SCAR	Stops Claims And Refunds	Acknowledged
●	DPI	Decimals Precision Inconsistency	Acknowledged

NCF - Non-Guaranteed Crowdsale Funds

Criticality	Minor / Informative
Location	GasClickICO.sol#L352
Status	Acknowledged

Description

The contract does not guarantee the funds the users will have to claim during the ongoing crowdsale stage. The invested amounts are transferred to the contract's balance, but during the claiming period, the funds are transferred to the users from the owner's balance.

```
IERC20(tokenAddress).safeTransferFrom(owner(), investor, claimed);
```

Recommendation

The contract could guarantee the funds will be available during the ongoing crowdsale stage. The owner is responsible for having all the available funds during the claiming period. 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 time-locker mechanism with a reasonable delay.
- Introduce a multi-sign wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.
- Renouncing the ownership will eliminate the threats but it is non-reversible.

Team Update

The team responded with the following statement:

"Having a multi network / multi payment token ICO contract, we cannot allocate the funds to a specific deployment in a network because we do not know the amount that will be purchased on every network. There is no technology today in blockchain to maintain a cross network token supply. We manage the keys of the owner with a 2 step transfer ownership and a multisig wallet. Cannot fix."

MVM - Missing Variable Modification

Criticality	Minor / Informative
Location	GasClickICO.sol#L295,330
Status	Acknowledged

Description

The contract does not decrease all the required variables when a user refunds or claims the investment. The variables `ptuUSDInvested` and `ptAmountInvested` should be decreased by the user's `cAmountInvested` and `cuUSDInvested` variables.

```
function refundInvestor(string calldata symbol, address investor) internal {
    ...
    contributions[investor].uUSDToPay -=
contributions[investor].conts[symbol].cuUSDInvested;
    delete contributions[investor].conts[symbol];
    ...
}
```

Recommendation

The team is advised to subtract the user's invested amounts from these variables.

Team Update

The team responded with the following statement:

"uUSDToPay and cuUSDInvested are only added for audit purposes and they won't interfere in logic. Decreasing cAmountInvested would have the same effect that assigning to zero, but more cost in gas, since the value is read just before. Won't fix."

MU - Modifiers Usage

Criticality	Minor / Informative
Location	GasClickICO.sol#L247,296,331
Status	Acknowledged

Description

The contract is using repetitive statements on some methods to validate some preconditions. In Solidity, the form of preconditions is usually represented by the modifiers. Modifiers allow you to define a piece of code that can be reused across multiple functions within a contract. This can be particularly useful when you have several functions that require the same checks to be performed before executing the logic within the function.

```
require(stage == CrowdsaleStage.Ongoing, "ERRD_MUST_ONG");
require(stage == CrowdsaleStage.Finished, "ERRR_MUST_FIN");
```

Recommendation

The team is advised to use modifiers since it is a useful tool for reducing code duplication and improving the readability of smart contracts. By using modifiers to perform these checks, it reduces the amount of code that is needed to write, which can make the smart contract more efficient and easier to maintain.

Team Update

The team responded with the following statement:

"We think it is more clear to have all the info to understand the function on the function itself than to scroll up and down along the codebase. A matter of coding preference. On the other hand, the modifier includes the 'require' so there will probably be a minimal gas degradation by using the modifier. Won't fix."

MC - Missing Check

Criticality	Minor / Informative
Location	GasClickICO.sol#L79,95GasClickAntiWhale.sol#L153,163
Status	Acknowledged

Description

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

The `softCapuUSD` should be less than the `hardCapuUSD` when set and vice versa. The `minuUSDTransfer` should be less than the `maxuUSDTransfer` when set and vice versa.

```
function setHardCapuUSD(uint256 hardCap) external onlyOwner {
    hardCapuUSD = hardCap;
    emit UpdatedHardCap(hardCap);
}
...
function setSoftCapuUSD(uint256 softCap) external onlyOwner {
    softCapuUSD = softCap;
    emit UpdatedSoftCap(softCap);
}
```

Recommendation

The team is advised to properly check the variables according to the required specifications.

Team Update

The team responded with the following statement:

"Having a multi network / multi payment token ICO contract, we cannot allocate the funds to a specific deployment in a network because we do not know the amount that will be purchased on every network. There is no technology today in blockchain to maintain a cross network token supply. We added the SoftCap and HardCap variables but assumed that probably would need to be manually managed. Won't fix."

SCAR - Stops Claims And Refunds

Criticality	Minor / Informative
Location	GasClickICO.sol#L297,332
Status	Acknowledged

Description

A user can refund or claim an investment by calling the `refund` and `claim` functions respectively. The invested amount is required to be less than the `softCapuUSD` in the case of a refund, and greater than the `softCapuUSD` in the case of a claim. The owner has the authority to set the `softCapuUSD` to any value. As a result, a user may not be able to refund/claim the investment.

```
require(totaluUSDTInvested < softCapuUSD, "ERRR_PASS_SOF");
require(totaluUSDTInvested > softCapuUSD, "ERRC_NPAS_SOF");
```

Recommendation

The contract could embody a check for not allowing setting the softCapuUSD less than and greater than a reasonable amount. 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 time-locker mechanism with a reasonable delay.
- Introduce a multi-sign wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.
- Renouncing the ownership will eliminate the threats but it is non-reversible.

Team Update

The team responded with the following statement:

"Having a multi network / multi payment token ICO contract, we cannot allocate the funds to a specific deployment in a network because we do not know the amount that will be purchased on every network. There is no technology today in blockchain to maintain a cross network token supply. Investors can either refund or claim, depending on the overall invested amount. There is no other choice, to maintain the multi network approach, than making softcap and hardcap flexible for each network. Cannot fix."

DPI - Decimals Precision Inconsistency

Criticality	Minor / Informative
Location	GasClickICO.sol#L336,352
Status	Acknowledged

Description

The decimals field of a contract's ERC20 token can be used to specify the number of decimal places that the token uses. For example, if decimals is set to `8`, it means that the smallest unit of the token is `0.00000001`, and if decimals are set to `18`, it means that the smallest unit of the token is `0.000000000000000001`.

However, there is an inconsistency in the way that the decimals field is handled in some ERC20 contracts. The ERC20 specification does not specify how the decimals field should be implemented, and as a result, some contracts use different precision numbers.

This inconsistency can cause problems when interacting with these contracts, as it is not always clear how the decimals field should be interpreted. For example, if a contract expects the decimals field to be 18 digits, but the contract being interacted with uses 8 digits, the result of the interaction may not be what was expected.

The contract transfers the claimable amount from the owner's balance to the investor's balance. The claimable amount is calculated assuming the `tokenAddress` has 18 decimal places. Since the `tokenAddress` can be changed, the token's decimal places may differ. As a result, the contract will transfer an inaccurate amount.

```
uint claimed = contributions[investor].uUSDToPay * 10**18 /
UUSDT_PER_TOKEN;
...
IERC20(tokenAddress).safeTransferFrom(owner(), investor, claimed);
```

Recommendation

The team is advised to use the token's decimals places as a variable for the calculation, instead of using a hardcoded value. This way, the contract will ensure that the claimable amount is accurate.

Team Update

The team responded with the following statement:

"Reported issue is technically correct but it would add complexity in error handling and some cost in gas. Despite the impact being negligible, the idea is to provide an ERC-20 token with 18 decimals. Won't fix."

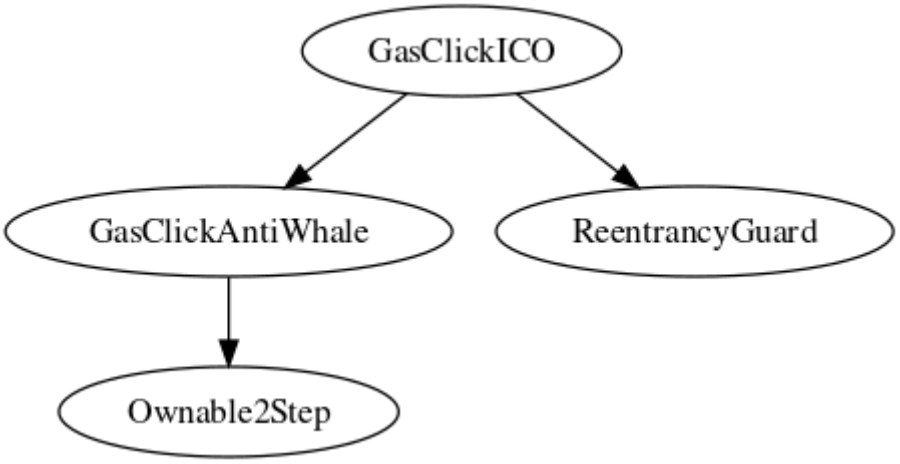
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
GasClickAntiWhale	Implementation	Ownable2Step		
	getWhitelistuUSDThreshold	External		-
	setWhitelistuUSDThreshold	External	✓	onlyOwner
	getWhitelisted	External		-
	getWhitelistUserCount	External		-
	isWhitelisted	External		-
	whitelistUser	External	✓	onlyOwner
	unwhitelistUser	External	✓	onlyOwner
	getUseBlacklist	External		-
	setUseBlacklist	External	✓	onlyOwner
	getBlacklisted	External		-
	getBlacklistUserCount	External		-
	isBlacklisted	External		-
	blacklistUser	External	✓	onlyOwner
	unblacklistUser	External	✓	onlyOwner
	isExcludedFromMaxInvestment	External		-
	setExcludedFromMaxInvestment	External	✓	onlyOwner
	getMaxUSDInvestment	External		-

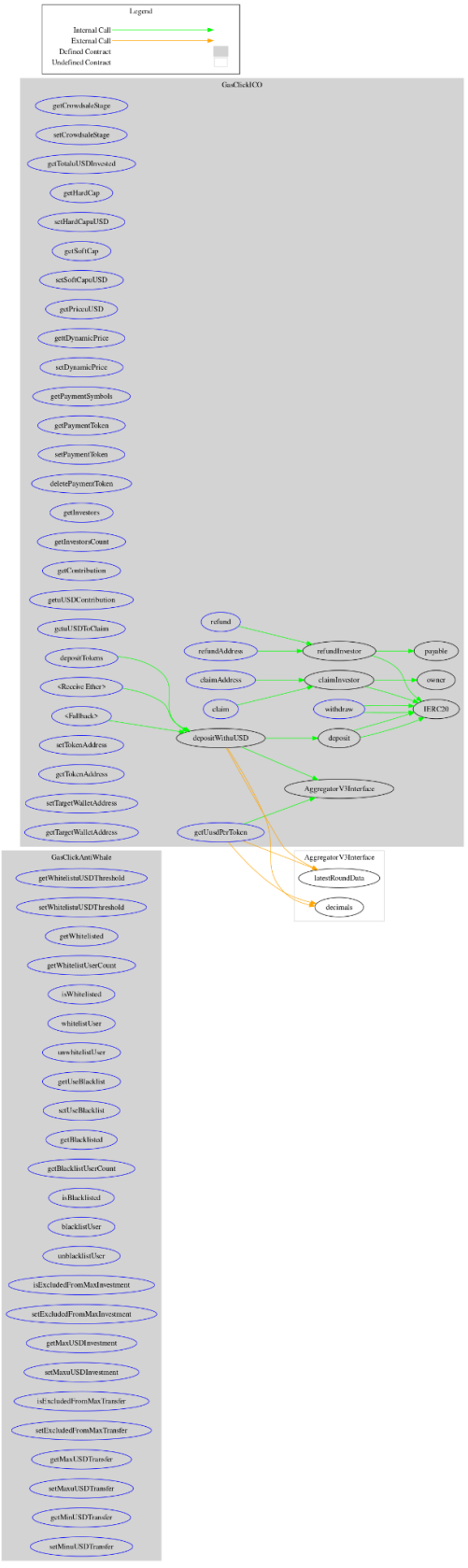
	setMaxuUSDInvestment	External	✓	onlyOwner
	isExcludedFromMaxTransfer	External		-
	setExcludedFromMaxTransfer	External	✓	onlyOwner
	getMaxUSDTransfer	External		-
	setMaxuUSDTransfer	External	✓	onlyOwner
	getMinUSDTransfer	External		-
	setMinuUSDTransfer	External	✓	onlyOwner
GasClickICO	Implementation	GasClickAntiWhale, ReentrancyGuard		
	getCrowdsaleStage	External		-
	setCrowdsaleStage	External	✓	onlyOwner
	getTotaluUSDInvested	External		-
	getHardCap	External		-
	setHardCapuUSD	External	✓	onlyOwner
	getSoftCap	External		-
	setSoftCapuUSD	External	✓	onlyOwner
	getPriceuUSD	External		-
	gettDynamicPrice	External		-
	setDynamicPrice	External	✓	onlyOwner
	getPaymentSymbols	External		-
	getPaymentToken	External		-
	setPaymentToken	External	✓	onlyOwner

	deletePaymentToken	External	✓	onlyOwner
	getUusdPerToken	External		-
	getInvestors	External		-
	getInvestorsCount	External		-
	getContribution	External		-
	getuUSDContribution	External		-
	getuUSDToClaim	External		-
		External	Payable	-
		External	Payable	-
	depositTokens	External	✓	nonReentrant
	depositWithuUSD	Internal	✓	
	deposit	Internal	✓	
	refund	External	✓	nonReentrant
	refundAddress	External	✓	nonReentrant onlyOwner
	refundInvestor	Internal	✓	
	claim	External	✓	nonReentrant
	claimAddress	External	✓	onlyOwner
	claimInvestor	Internal	✓	
	setTokenAddress	External	✓	onlyOwner
	getTokenAddress	External		-
	withdraw	External	✓	nonReentrant onlyOwner
	setTargetWalletAddress	External	✓	onlyOwner
	getTargetWalletAddress	External		-

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
GasClickICO
getCrowdsaleStage
setCrowdsaleStage
getTotalsUSDInvested
getHardCap
setHardCapsUSD
getSoftCap
setSoftCapsUSD
getPricesUSD
getDynamicPrice
setDynamicPrice
getPaymentSymbols
getPaymentToken
setPaymentToken
deletePaymentToken
getInvestors
getInvestorsCount
getContribution
getsUSDContribution
getsUSDToClaim
depositTokens
<Receive Ether>
<Fallback>
setTokenAddress
getTokenAddress
setTargetWalletAddress
getTargetWalletAddress
refund
refundAddress
claimAddress
claim
depositWithsUSD
refundInvestor
claimInvestor
withdraw
deposit
payable
owner
IERC20
AggregatorV3Interface
getUsdPerToken
AggregatorV3Interface
latestRoundData
decimals
GasClickAntiWhale
getWhitelistsUSDThreshold
setWhitelistsUSDThreshold
getWhitelisted
getWhitelistUserCount
isWhitelisted
whitelistUser
unwhitelistUser
getUseBlacklist
setUseBlacklist
getBlacklisted
getBlacklistUserCount
isBlacklisted
blacklistUser
unblacklistUser
isExcludedFromMaxInvestment
setExcludedFromMaxInvestment
getMaxUSDInvestment
setMaxsUSDInvestment
isExcludedFromMaxTransfer
setExcludedFromMaxTransfer
getMaxUSDTransfer
setMaxsUSDTransfer
getMinUSDTransfer
setMinsUSDTransfer

Summary

Gasclick contract implements a financial mechanism. This audit investigates security issues, business logic concerns, and potential improvements.

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