Cyberscope

Audit Report

Metaburst

September 2022

Type	BEP20
Network	BSC
Address	0xC3eFc6E1cFeCb4903570eD86142c7f9A8A760BF6
Audited by	© cyberscope

Table of Contents

Table of Contents 1
Contract Review 3
Source Files 3
Audit Updates 4
Contract Analysis 5
Contract Diagnostics 6
STC - Succeeded Transfer Check 7
Description 7
Recommendation 7
L01 - Public Function could be Declared External 8
Description 8
Recommendation 8
L04 - Conformance to Solidity Naming Conventions 9
Description 9
Recommendation 9
L07 - Missing Events Arithmetic 10
Description 10
Recommendation 10
L12 - Using Variables before Declaration 11
Description 11
Recommendation 11
L14 - Uninitialized Variables in Local Scope 12
Description 12
Recommendation 12
Contract Functions 13
Contract Flow 16

Domain Info 17
Summary 18
Disclaimer 19
About Cyberscope 20

Contract Review

Contract Name	Metaburst
Compiler Version	v0.8.17+commit.8df45f5f
Optimization	5000 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xC3eFc6E1cFeCb4903570eD86142c7f9A8A760BF6
Symbol	MEBU
Decimals	18
Total Supply	100,000,000
Domain	https://metaburst.io

Source Files

Filename	**SHA256**
contract.sol	1d535f92e17baa5cdc2771fccd1c49ecc0893433caa08a82005af6f6d48d6a65

Audit Updates

Initial Audit	9th September 2022 https://github.com/cyberscope-io/audits/blob/main/mebu/v1/audit.pdf
Corrected Phase 1	21st September 2022 https://github.com/cyberscope-io/audits/blob/main/mebu/v2/audit.pdf
Corrected Phase 2	22nd September 2022

Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OCTD	Transfers Contract's Tokens	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	ULTW	Transfers Liquidity to Team Wallet	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Contract Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
●	STC	Succeeded Transfer Check	Unresolved
●	L01	Public Function could be Declared External	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L07	Missing Events Arithmetic	Unresolved
●	L12	Using Variables before Declaration	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved

STC - Succeeded Transfer Check

Criticality	minor / informative
Location	contract.sol#L523
Status	Unresolved

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.

```
function contractSwap(uint256 contractTokenBalance) internal inSwapFlag {
        if (_allowances[address(this)][address(dexRouter)] != type(uint256).max) {
            _allowances[address(this)][address(dexRouter)] = type(uint256).max;
        }

        address[] memory path = new address[](2);
        path[0] = address(this);
        path[1] = dexRouter.WETH();

        try dexRouter.swapExactTokensForETHSupportingFeeOnTransferTokens(
            contractTokenBalance,
            0,
            path,
            address(this),
            block.timestamp
        ) {} catch {
            return;
        }

        bool success;
        (success,) = marketingWallet.call{value: address(this).balance, gas: 35000}("");
    }
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L536,271,352
Status	Unresolved

Description

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

```
enableTrading
transfer
getCirculatingSupply
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L116,33,117,360,114,133,152,136,135,115,118,126,132,134
Status	Unresolved

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.

```
_symbol
WETH
_decimals
_antiBlock
startingSupply
maxSellTaxes
_hasLiqBeenAdded
masterTaxDivisor
maxRoundtripTax
...
```

Recommendation

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

L07 - Missing Events Arithmetic

Criticality	minor / informative
Location	contract.sol#L407,416
Status	Unresolved

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
swapThreshold = (_tTotal * thresholdPercent) / thresholdDivisor
piSwapPercent = priceImpactSwapPercent
```

Recommendation

Emit an event for critical parameter changes.

L12 - Using Variables before Declaration

Criticality	minor / informative
Location	contract.sol#L565
Status	Unresolved

Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
check
```

Recommendation

The variables should be declared before any usage of them.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	contract.sol#L565,564
Status	Unresolved

Description

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

```
check
checked
```

Recommendation

All the local scoped variables should be initialized.

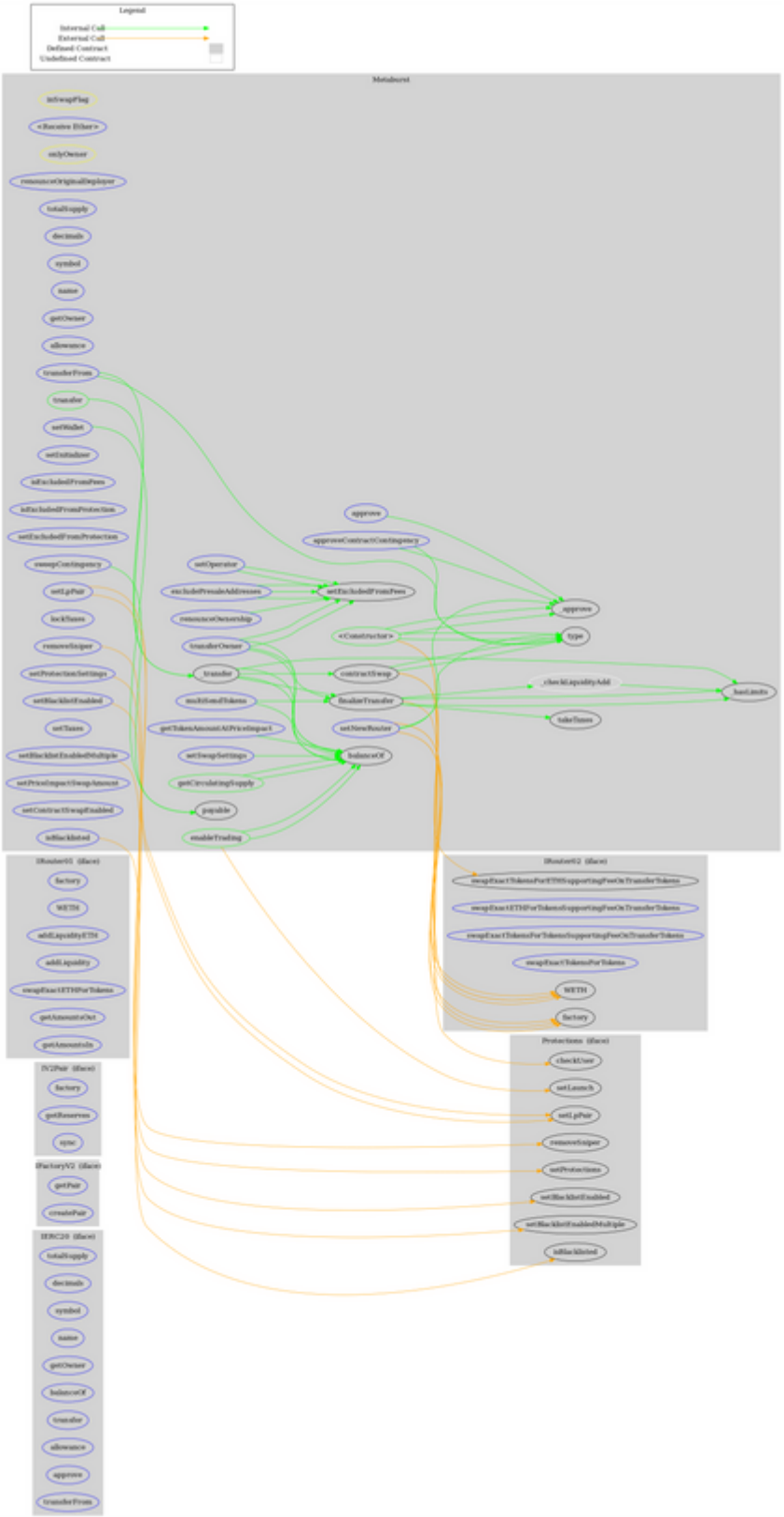
Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IFactoryV2	Interface			
	getPair	External		-
	createPair	External	✓	-
IV2Pair	Interface			
	factory	External		-
	getReserves	External		-
	sync	External	✓	-
IRouter01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	addLiquidity	External	✓	-
	swapExactETHForTokens	External	Payable	-
	getAmountsOut	External		-

	getAmountsIn	External		-
IRouter02	Interface	IRouter01		
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokens	External	✓	-
Protections	Interface			
	checkUser	External	✓	-
	setLaunch	External	✓	-
	setLpPair	External	✓	-
	setProtections	External	✓	-
	removeSniper	External	✓	-
	isBlacklisted	External		-
	setBlacklistEnabled	External	✓	-
	setBlacklistEnabledMultiple	External	✓	-
Metaburst	Implementation	IERC20		
	<Constructor>	Public	Payable	-
	<Receive Ether>	External	Payable	-
	transferOwner	External	✓	onlyOwner
	renounceOwnership	External	✓	onlyOwner
	setOperator	External	✓	-
	renounceOriginalDeployer	External	✓	-
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	allowance	External		-
	balanceOf	Public		-
	transfer	Public	✓	-

	approve	External	✓	-
	_approve	Internal	✓	
	approveContractContingency	External	✓	onlyOwner
	transferFrom	External	✓	-
	setNewRouter	External	✓	onlyOwner
	setLpPair	External	✓	onlyOwner
	setInitializer	External	✓	onlyOwner
	isExcludedFromFees	External		-
	isExcludedFromProtection	External		-
	setExcludedFromFees	Public	✓	onlyOwner
	setExcludedFromProtection	External	✓	onlyOwner
	getCirculatingSupply	Public		-
	removeSniper	External	✓	onlyOwner
	setProtectionSettings	External	✓	onlyOwner
	lockTaxes	External	✓	onlyOwner
	setBlacklistEnabled	External	✓	-
	setBlacklistEnabledMultiple	External	✓	-
	isBlacklisted	External		-
	setTaxes	External	✓	onlyOwner
	getTokenAmountAtPriceImpact	External		-
	setWallet	External	✓	onlyOwner
	setSwapSettings	External	✓	onlyOwner
	setPriceImpactSwapAmount	External	✓	onlyOwner
	setContractSwapEnabled	External	✓	onlyOwner
	excludePresaleAddresses	External	✓	onlyOwner
	_hasLimits	Internal		
	_transfer	Internal	✓	
	contractSwap	Internal	✓	inSwapFlag
	_checkLiquidityAdd	Internal	✓	
	enableTrading	Public	✓	onlyOwner
	sweepContingency	External	✓	onlyOwner
	multiSendTokens	External	✓	onlyOwner
	finalizeTransfer	Internal	✓	
	takeTaxes	Internal	✓	

Contract Flow



Domain Info

Domain Name	metaburst.io
Registry Domain ID	ed752fba07a84638ab935a2a219c69ba-DONUTS
Creation Date	2021-12-15T08:23:13Z
Updated Date	2022-08-01T09:03:35Z
Registry Expiry Date	2022-12-15T08:23:13Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain was created 9 months before the creation of the audit. It will expire in 3 months.

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

Summary

Metaburst Token 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. There is also a limit of max 15% fees.

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

Cyberscope

Audit Report

Metaburst

September 2022

Type	BEP20
Network	BSC
Address	0xd945d69d165505d1d96259B1902f143d63b89bc3
Audited by	© cyberscope

Table of Contents

Table of Contents 1
Contract Review 3
Source Files 3
Audit Updates 3
Contract Analysis 4
OCTD - Transfers Contract's Tokens 5
Description 5
Recommendation 5
ULTW - Transfers Liquidity to Team Wallet 6
Description 6
Recommendation 7
Contract Diagnostics 8
FSA - Fixed Swap Address 9
Description 9
Recommendation 9
STC - Succeeded Transfer Check 10
Description 10
Recommendation 10
L01 - Public Function could be Declared External 11
Description 11
Recommendation 11
L02 - State Variables could be Declared Constant 12
Description 12
Recommendation 12
L04 - Conformance to Solidity Naming Conventions 13
Description 13

Recommendation 13
L09 - Dead Code Elimination 14
Description 14
Recommendation 14
L14 - Uninitialized Variables in Local Scope 15
Description 15
Recommendation 15
L15 - Local Scope Variable Shadowing 16
Description 16
Recommendation 16
Contract Functions 17
Contract Flow 22
Domain Info 23
Summary 24
Disclaimer 25
About Cyberscope 26

Contract Review

Contract Name	Token
Compiler Version	v0.8.12+commit.f00d7308
Optimization	20 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xd945d69d165505d1d96259B1902f143d63b89bc3
Symbol	MEBU
Decimals	18
Total Supply	100,000,000
Domain	https://metaburst.io

Source Files

Filename	**SHA256**
contract.sol	1afc9d62425705de89014ee3cda4a91fe6c7291afb4971456372a4a74048b16a

Audit Updates

Initial Audit	9th September 2022
Corrected	

Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OCTD	Transfers Contract's Tokens	Unresolved
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	ULTW	Transfers Liquidity to Team Wallet	Unresolved
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

OCTD - Transfers Contract's Tokens

Criticality	minor / informative
Location	contract.sol#L1544
Status	Unresolved

Description

The contract owner has the authority to claim all the balance of the contract. The owner may take advantage of it by calling the withDraw function.

```
function withDraw() external onlyRole(PROJECT_CONTROL_ROLE) {
        // just in-case swap not work
        uint256 contractTokenBalance = balanceOf(address(this));
        _transfer(address(this), projectWallet, contractTokenBalance);
    }
```

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

ULTW - Transfers Liquidity to Team Wallet

Criticality	minor / informative
Location	contract.sol#L1550,1529
Status	Unresolved

Description

The contract owner has the authority to transfer funds without limit to the team wallet. These funds have been accumulated from fees collected from the contract. The owner may take advantage of it by calling the `claimStuckTokens` and `swapAndSendProject` methods.

```
function claimStuckTokens(address token)
        external
        onlyRole(PROJECT_CONTROL_ROLE)
    {
        require(token != address(this), "Owner cannot claim native tokens");
        if (token == address(0x0)) {
            payable(msg.sender).transfer(address(this).balance);
            return;
        }
        IERC20 ERC20token = IERC20(token);
        uint256 balance = ERC20token.balanceOf(address(this));
        ERC20token.transfer(msg.sender, balance);
    }

function swapAndSendProject(uint256 tokenAmount)
        external
        onlyRole(SWEEPER_ROLE)
    {
        uint256 contractTokenBalance = balanceOf(address(this));
        if (tokenAmount > contractTokenBalance) {
            tokenAmount = contractTokenBalance;
        }
        if (tokenAmount >= tokenForProject) {
            _swapTokensForEth(tokenForProject);
        } else {
            _swapTokensForEth(tokenAmount);
        }
    }
```

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

Contract Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
	FSA	Fixed Swap Address	Unresolved
	STC	Succeeded Transfer Check	Unresolved
	L01	Public Function could be Declared External	Unresolved
	L02	State Variables could be Declared Constant	Unresolved
	L04	Conformance to Solidity Naming Conventions	Unresolved
	L09	Dead Code Elimination	Unresolved
	L14	Uninitialized Variables in Local Scope	Unresolved
	L15	Local Scope Variable Shadowing	Unresolved

FSA - Fixed Swap Address

Criticality	minor / informative
Location	contract.sol#L1523
Status	Unresolved

Description

The swap address is assigned once in the constructor and it can not be changed. The decentralized swaps sometimes create a new swap version or abandon the current. A contract that cannot change the swap address may not be able to catch-up the upgrade.

```
constructor(
      string memory name,
      string memory symbol,
      address newRouter,
      address multiSigWallet
  ) ERC20(name, symbol) {
      _mint(_msgSender(), maxSupply);
      IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(
          newRouter // address uniswap
      );
      uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory())
          .createPair(address(this), _uniswapV2Router.WETH());
      uniswapV2Router = _uniswapV2Router;

      _setupRole(DEFAULT_ADMIN_ROLE, multiSigWallet);
  }
```

Recommendation

It could be better to allow the swap address mutation in case of future swap updates.

STC - Succeeded Transfer Check

Criticality	minor / informative
Location	contract.sol#L1561
Status	Unresolved

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.

```
function claimStuckTokens(address token)
        external
        onlyRole(PROJECT_CONTROL_ROLE)
    {
        require(token != address(this), "Owner cannot claim native tokens");
        if (token == address(0x0)) {
            payable(msg.sender).transfer(address(this).balance);
            return;
        }
        IERC20 ERC20token = IERC20(token);
        uint256 balance = ERC20token.balanceOf(address(this));
        ERC20token.transfer(msg.sender, balance);
    }
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L529,458,572,441,484,552,433,1126,1141,465,507,1161
Status	Unresolved

Description

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

```
transferFrom
decimals
decreaseAllowance
symbol
transfer
increaseAllowance
name
grantRole
revokeRole
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor / informative
Location	contract.sol#L1506
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
maxSupply
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L1469,1429,1455,1237,1442,1419
Status	Unresolved

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.

```
_buyFeeRate
_sellFeeRate
_target
_tokenForProject
WETH
_projectWallet
```

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 / informative
Location	contract.sol#L1196,889,864,651,920
Status	Unresolved

Description

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

```
_setRoleAdmin
toHexString
toString
_burn
```

Recommendation

Remove unused functions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	contract.sol#L1582
Status	Unresolved

Description

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

```
transferFeeRate
```

Recommendation

All the local scoped variables should be initialized.

L15 - Local Scope Variable Shadowing

Criticality	minor / informative
Location	contract.sol#L1510,1509
Status	Unresolved

Description

The are variables that are defined in the local scope containing the same name from an upper scope.

```
symbol
name
```

Recommendation

The local variables should have different names from the upper scoped variables.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
SafeMath	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
	div	Internal		
	mod	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-

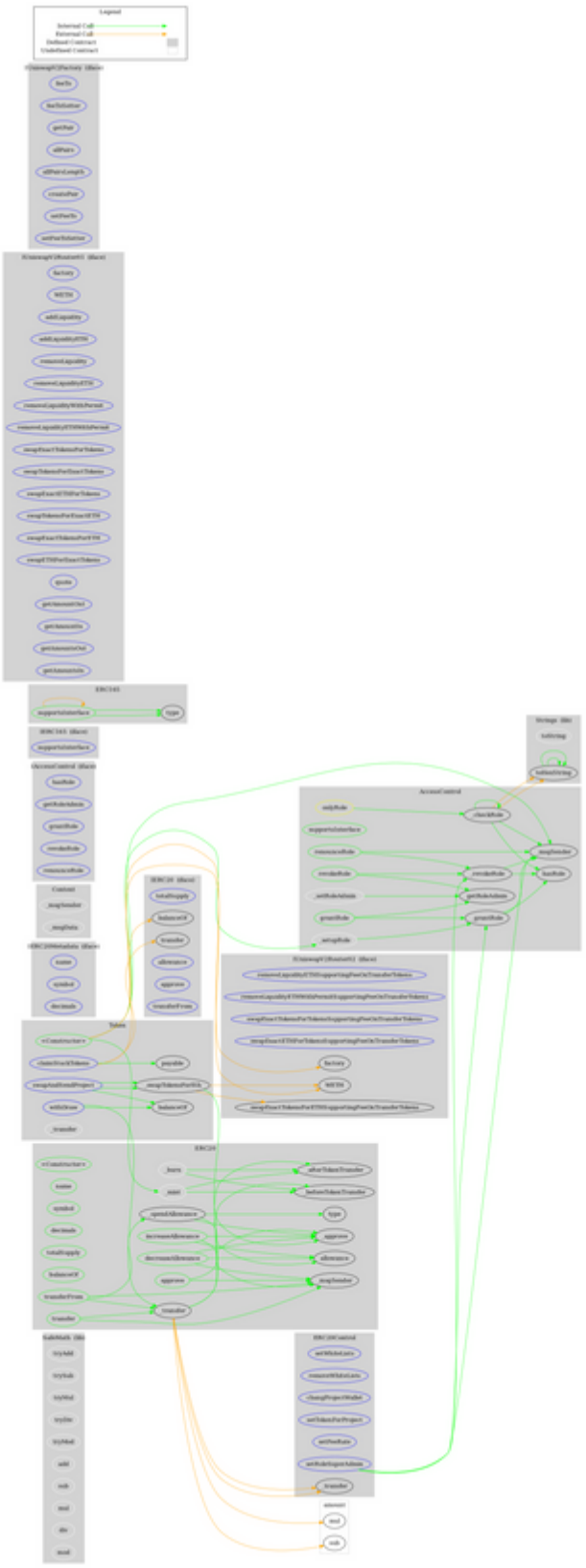
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_spendAllowance	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
IAccessControl	Interface			
	hasRole	External		-
	getRoleAdmin	External		-
	grantRole	External	✓	-
	revokeRole	External	✓	-
	renounceRole	External	✓	-

Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
	toHexString	Internal		
IERC165	Interface			
	supportsInterface	External		-
ERC165	Implementation	IERC165		
	supportsInterface	Public		-
AccessControl	Implementation	Context, IAccessControl, ERC165		
	supportsInterface	Public		-
	hasRole	Public		-
	_checkRole	Internal		
	_checkRole	Internal		
	getRoleAdmin	Public		-
	grantRole	Public	✓	onlyRole
	revokeRole	Public	✓	onlyRole
	renounceRole	Public	✓	-
	_setupRole	Internal	✓	
	_setRoleAdmin	Internal	✓	
	_grantRole	Internal	✓	
	_revokeRole	Internal	✓	
IUniswapV2Router01	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		-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-

	setFeeToSetter	External	✓	-
ERC20Control	Implementation	AccessControl		
	setWhiteLists	External	✓	onlyRole
	removeWhiteLists	External	✓	onlyRole
	changProjectWallet	External	✓	onlyRole
	setTokenForProject	External	✓	onlyRole
	setFeeRate	External	✓	onlyRole
	setRoleSuperAdmin	External	✓	onlyRole
Token	Implementation	ERC20, ERC20Control		
	<Constructor>	Public	✓	ERC20
	swapAndSendProject	External	✓	onlyRole
	withDraw	External	✓	onlyRole
	claimStuckTokens	External	✓	onlyRole
	_swapTokensForEth	Private	✓	
	_transfer	Internal	✓	

Contract Flow



Domain Info

Domain Name	metaburst.io
Registry Domain ID	ed752fba07a84638ab935a2a219c69ba-DONUTS
Creation Date	2021-12-15T08:23:13Z
Updated Date	2022-08-01T09:03:35Z
Registry Expiry Date	2022-12-15T08:23:13Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain was created 9 months before the creation of the audit. It will expire in 3 months.

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

Summary

There are some functions that can be abused by the owner like transferring tokens to the team’s wallet and transferring funds to the team’s wallet. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats. There is also a limit of max 20% fees.

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

Cyberscope

Audit Report

Metaburst

September 2022

SHA256 1d535f92e17baa5cdc2771fccd1c49ecc0893433caa08a82005af6f6d48d6a65

Audited by © cyberscope

Table of Contents

Table of Contents 1
Contract Review 3
Source Files 3
Audit Updates 3
Contract Analysis 4
Contract Diagnostics 5
STC - Succeeded Transfer Check 6
Description 6
Recommendation 6
L01 - Public Function could be Declared External 7
Description 7
Recommendation 7
L04 - Conformance to Solidity Naming Conventions 8
Description 8
Recommendation 8
L07 - Missing Events Arithmetic 9
Description 9
Recommendation 9
L12 - Using Variables before Declaration 10
Description 10
Recommendation 10
L14 - Uninitialized Variables in Local Scope 11
Description 11
Recommendation 11
Contract Functions 12
Contract Flow 15

Domain Info 16

Summary 17

Disclaimer 18

About Cyberscope 19

Contract Review

Contract Name	Metaburst
Compiler Version	v0.8.6+commit.11564f7e
Testing Deploy	https://testnet.bscscan.com/token/0x1286e6a56F269B4ECd0e4967fbDd0a305abA4923
Symbol	MEBU
Decimals	18
Total Supply	100,000,000
Domain	https://metaburst.io

Source Files

Filename	**SHA256**
contract.sol	1d535f92e17baa5cdc2771fccd1c49ecc0893433caa08a82005af6f6d48d6a65

Audit Updates

Initial Audit	9th September 2022 https://github.com/cyberscope-io/audits/blob/main/mebu/v1/audit.pdf
Corrected	21st September 2022

Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OCTD	Transfers Contract's Tokens	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	ULTW	Transfers Liquidity to Team Wallet	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Contract Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
●	STC	Succeeded Transfer Check	Unresolved
●	L01	Public Function could be Declared External	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L07	Missing Events Arithmetic	Unresolved
●	L12	Using Variables before Declaration	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved

STC - Succeeded Transfer Check

Criticality	minor / informative
Location	contract.sol#L520
Status	Unresolved

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.

```
function contractSwap(uint256 contractTokenBalance) internal inSwapFlag {
        if (_allowances[address(this)][address(dexRouter)] != type(uint256).max) {
            _allowances[address(this)][address(dexRouter)] = type(uint256).max;
        }

        address[] memory path = new address[](2);
        path[0] = address(this);
        path[1] = dexRouter.WETH();

        try dexRouter.swapExactTokensForETHSupportingFeeOnTransferTokens(
            contractTokenBalance,
            0,
            path,
            address(this),
            block.timestamp
        ) {} catch {
            return;
        }

        bool success;
        (success,) = marketingWallet.call{value: address(this).balance, gas: 35000}("");
    }
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L536,271,352
Status	Unresolved

Description

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

```
enableTrading
transfer
getCirculatingSupply
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L116,33,117,360,114,133,152,136,135,115,118,126,132,134
Status	Unresolved

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.

```
_symbol
WETH
_decimals
_antiBlock
startingSupply
maxSellTaxes
_hasLiqBeenAdded
masterTaxDivisor
maxRoundtripTax
...
```

Recommendation

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

L07 - Missing Events Arithmetic

Criticality	minor / informative
Location	contract.sol#L407,416
Status	Unresolved

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
swapThreshold = (_tTotal * thresholdPercent) / thresholdDivisor
piSwapPercent = priceImpactSwapPercent
```

Recommendation

Emit an event for critical parameter changes.

L12 - Using Variables before Declaration

Criticality	minor / informative
Location	contract.sol#L565
Status	Unresolved

Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
check
```

Recommendation

The variables should be declared before any usage of them.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	contract.sol#L565,564
Status	Unresolved

Description

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

```
check
checked
```

Recommendation

All the local scoped variables should be initialized.

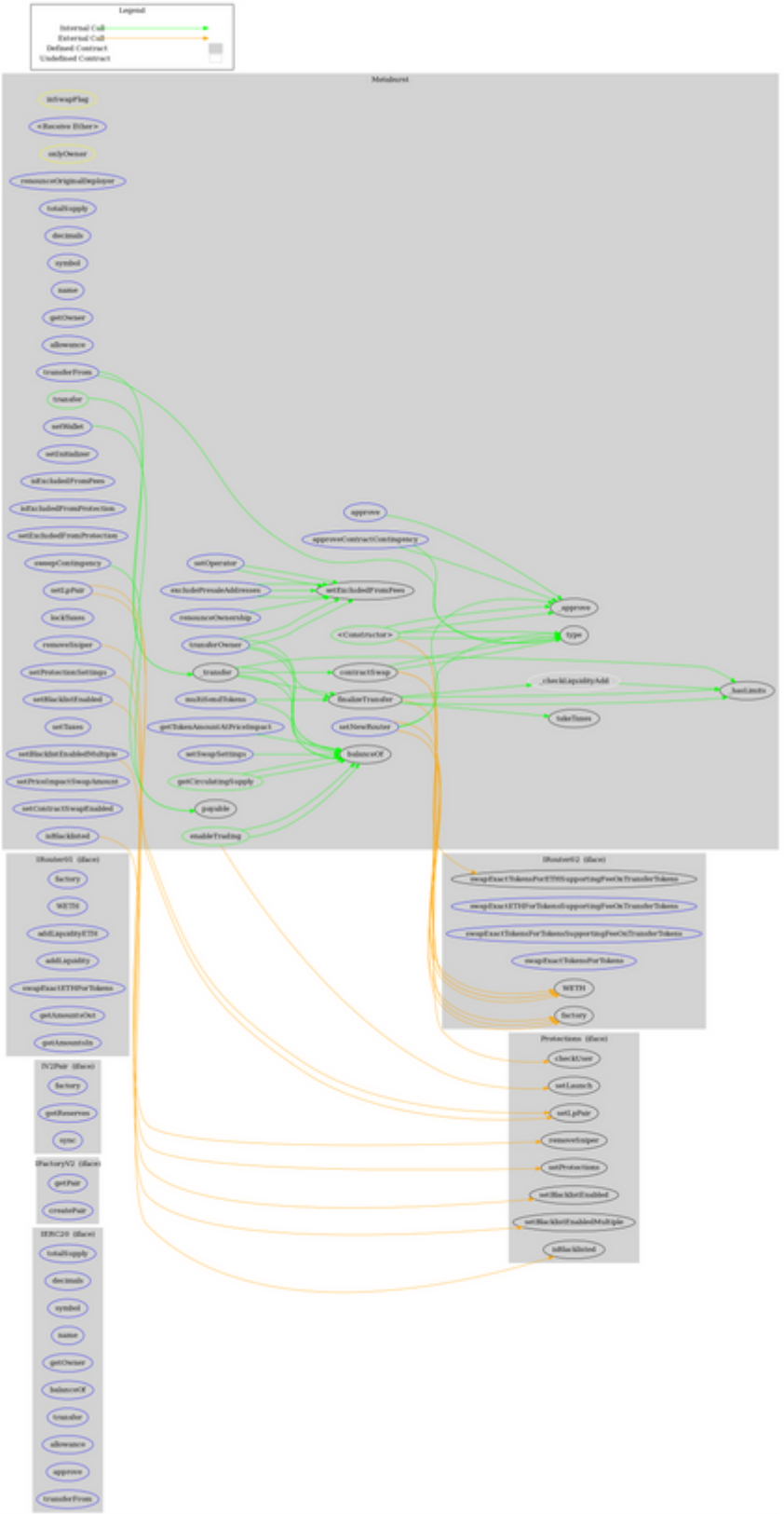
Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IFactoryV2	Interface			
	getPair	External		-
	createPair	External	✓	-
IV2Pair	Interface			
	factory	External		-
	getReserves	External		-
	sync	External	✓	-
IRouter01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	addLiquidity	External	✓	-
	swapExactETHForTokens	External	Payable	-
	getAmountsOut	External		-

	getAmountsIn	External		-
IRouter02	Interface	IRouter01		
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokens	External	✓	-
Protections	Interface			
	checkUser	External	✓	-
	setLaunch	External	✓	-
	setLpPair	External	✓	-
	setProtections	External	✓	-
	removeSniper	External	✓	-
	isBlacklisted	External		-
	setBlacklistEnabled	External	✓	-
	setBlacklistEnabledMultiple	External	✓	-
Metaburst	Implementation	IERC20		
	<Constructor>	Public	Payable	-
	<Receive Ether>	External	Payable	-
	transferOwner	External	✓	onlyOwner
	renounceOwnership	External	✓	onlyOwner
	setOperator	External	✓	-
	renounceOriginalDeployer	External	✓	-
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	allowance	External		-
	balanceOf	Public		-
	transfer	Public	✓	-

	approve	External	✓	-
	_approve	Internal	✓	
	approveContractContingency	External	✓	onlyOwner
	transferFrom	External	✓	-
	setNewRouter	External	✓	onlyOwner
	setLpPair	External	✓	onlyOwner
	setInitializer	External	✓	onlyOwner
	isExcludedFromFees	External		-
	isExcludedFromProtection	External		-
	setExcludedFromFees	Public	✓	onlyOwner
	setExcludedFromProtection	External	✓	onlyOwner
	getCirculatingSupply	Public		-
	removeSniper	External	✓	onlyOwner
	setProtectionSettings	External	✓	onlyOwner
	lockTaxes	External	✓	onlyOwner
	setBlacklistEnabled	External	✓	-
	setBlacklistEnabledMultiple	External	✓	-
	isBlacklisted	External		-
	setTaxes	External	✓	onlyOwner
	getTokenAmountAtPriceImpact	External		-
	setWallet	External	✓	onlyOwner
	setSwapSettings	External	✓	onlyOwner
	setPriceImpactSwapAmount	External	✓	onlyOwner
	setContractSwapEnabled	External	✓	onlyOwner
	excludePresaleAddresses	External	✓	onlyOwner
	_hasLimits	Internal		
	_transfer	Internal	✓	
	contractSwap	Internal	✓	inSwapFlag
	_checkLiquidityAdd	Internal	✓	
	enableTrading	Public	✓	onlyOwner
	sweepContingency	External	✓	onlyOwner
	multiSendTokens	External	✓	onlyOwner
	finalizeTransfer	Internal	✓	
	takeTaxes	Internal	✓	

Contract Flow



Domain Info

Domain Name	metaburst.io
Registry Domain ID	ed752fba07a84638ab935a2a219c69ba-DONUTS
Creation Date	2021-12-15T08:23:13Z
Updated Date	2022-08-01T09:03:35Z
Registry Expiry Date	2022-12-15T08:23:13Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain was created 9 months before the creation of the audit. It will expire in 3 months.

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

Summary

Metaburst Token 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. There is also a limit of max 15% fees.

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