Cyberscope

Audit Report
Virtual Versions

August 2023

Network	ETH
Address	0x5c39ef4bda89a9D7EDB18A16E53c97D5D32245F6
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
●	US	Untrusted Source	Acknowledged
●	MEE	Missing Events Emission	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L19	Stable Compiler Version	Unresolved

Table of Contents

Review

Contract Name	VirtualVersions
Compiler Version	v0.8.15+commit.e14f2714
Optimization	200 runs
Explorer	https://etherscan.io/address/0x5c39ef4bda89a9d7edb18a16e53c97d5d32245f6
Address	0x5c39ef4bda89a9d7edb18a16e53c97d5d32245f6
Network	ETH
Symbol	VV
Decimals	18
Total Supply	1,000,000,000

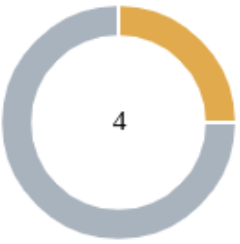
Audit Updates

Initial Audit	06 Aug 2023 https://github.com/cyberscope-io/audits/blob/main/1-vv/v1/audit.pdf
Corrected Phase 2	09 Aug 2023 https://github.com/cyberscope-io/audits/blob/main/1-vv/v2/audit.pdf
Corrected Phase 3	18 Aug 2023

Source Files

Filename	SHA256
contracts/VirtualVersions.sol	e372ceac12ed7b281126335a1fa59992db52ab01233bd39ceef1aebe1a2a9904
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a
@openzeppelin/contracts/token/ERC20/IERC20.sol	94f23e4af51a18c2269b355b8c7cf4db8003d075c9c541019eb8dcf4122864d5
@openzeppelin/contracts/token/ERC20/ERC20.sol	3cd9bf87ad804088f574a5266771f038a2c44b53d85f355aadb35645e497d1c2
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	af5c8a77965cc82c33b7ff844deb9826166689e55dc037a7f2f790d057811990
@openzeppelin/contracts/access/Ownable.sol	75e3c97011e75627ffb36f4a2799a4e887e1a3e27ed427490e82d7b6f51cc5c9

Findings Breakdown



Critical	0
Medium	1
Minor / Informative	3

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

US - Untrusted Source

Criticality	Medium
Location	contracts/VirtualVersions.sol#L33,42
Status	Acknowledged

Description

The contract uses an external contract in order to determine the transaction's flow. The external contract is untrusted. As a result, it may produce security issues and harm the transactions.

```
    function _beforeTokenTransfer(
        address from,
        address to,
        uint256 amount
    ) internal override {
        if (from == address(0) || to == address(0)) return;
        if (!antisnipeDisable && address(antisnipe) != address(0))
            antisnipe.assureCanTransfer(msg.sender, from, to, amount);
    }

    function setAntisnipeAddress(address addr) external onlyOwner {
        antisnipe = IAntisnipe(addr);
    }
```

Recommendation

The contract should use a trusted external source. A trusted source could be either a commonly recognized or an audited contract. The pointing addresses should not be able to change after the initialization.

Team Update

The team has acknowledged that this is not a security issue and states:

"Project team is acknowledged about using the external smart contract. This is the part of token protection system blocking malicious actors, frontrunners, sniping bots and other players harming the ecosystem of Virtual Versions.

This system has no access to the investors' funds and tokens, able only to scan and allow or block transactions. In case of necessity Virtual Versions team has the opportunity to turn the tracker off setting it's value to null."

MEE - Missing Events Emission

Criticality	Minor / Informative
Location	contracts/VirtualVersions.sol#L42
Status	Unresolved

Description

The contract performs actions and state mutations from external methods that do not result in the emission of events. Emitting events for significant actions is important as it allows external parties, such as wallets or dApps, to track and monitor the activity on the contract. Without these events, it may be difficult for external parties to accurately determine the current state of the contract.

```
    function setAntisnipeAddress(address addr) external
onlyOwner {
        antisnipe = IAntisnipe(addr);
    }
```

Recommendation

It is recommended to include events in the code that are triggered each time a significant action is taking place within the contract. These events should include relevant details such as the user's address and the nature of the action taken. By doing so, the contract will be more transparent and easily auditable by external parties. It will also help prevent potential issues or disputes that may arise in the future.

L02 - State Variables could be Declared Constant

Criticality	Minor / Informative
Location	contracts/VirtualVersions.sol#L17
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.

```
uint256 public totalSupply_ = 1_000_000_000 ether
```

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.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	contracts/VirtualVersions.sol#L2
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 ^0.8.0;
```

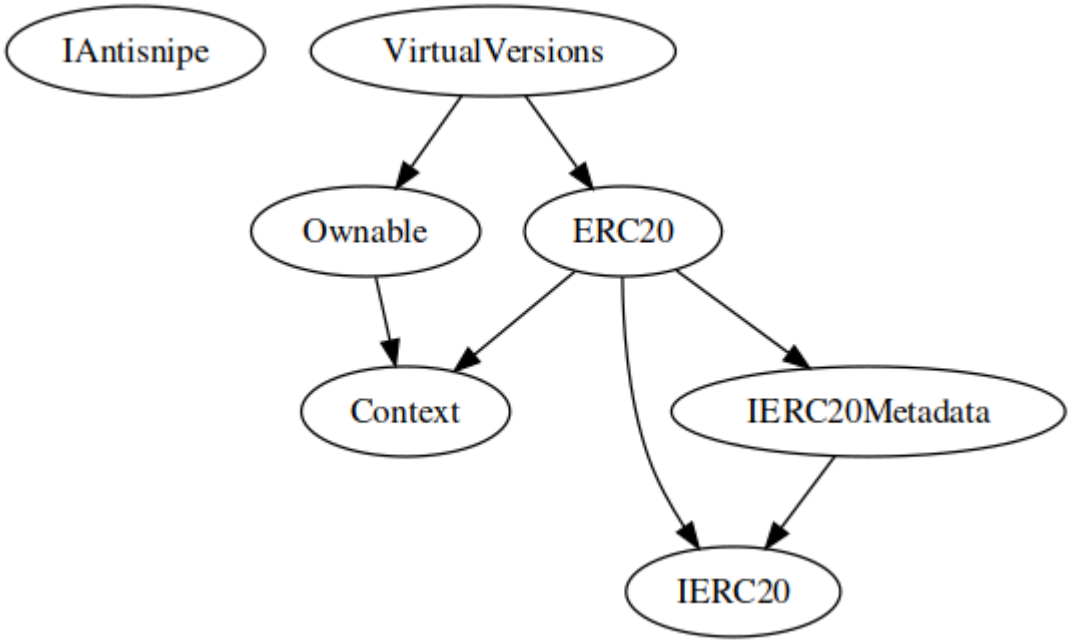
Recommendation

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.

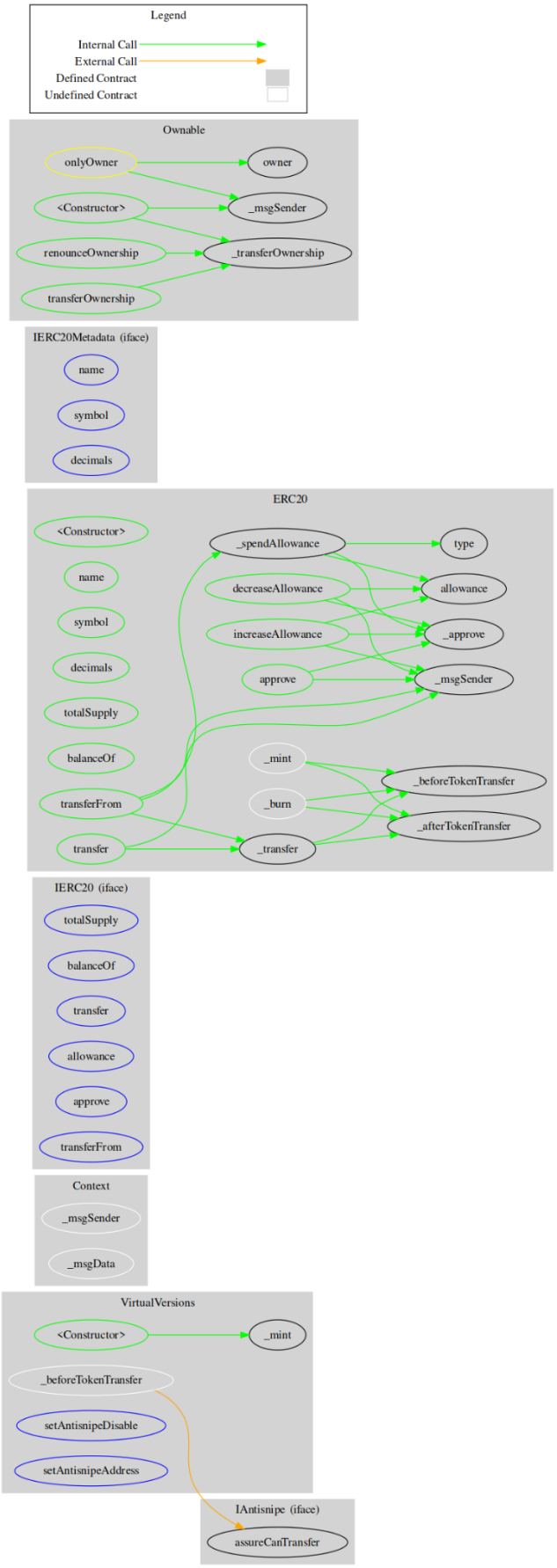
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IAntisnipe	Interface			
	assureCanTransfer	External	✓	-
VirtualVersions	Implementation	ERC20, Ownable		
		Public	✓	ERC20
	_beforeTokenTransfer	Internal	✓	
	setAntisnipeDisable	External	✓	onlyOwner
	setAntisnipeAddress	External	✓	onlyOwner

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
Ownable
onlyOwner
owner
<Constructor>
_msgSender
renounceOwnership
_transferOwnership
transferOwnership
IERC20Metadata (iface)
name
symbol
decimals
ERC20
<Constructor>
name
symbol
decimals
totalSupply
balanceOf
transferFrom
transfer
_spendAllowance
decreaseAllowance
increaseAllowance
approve
type
allowance
_approve
_msgSender
_mint
_burn
_transfer
_beforeTokenTransfer
_afterTokenTransfer
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
Context
_msgSender
_msgData
VirtualVersions
<Constructor>
_mint
_beforeTokenTransfer
setAntisnipeDisable
setAntisnipeAddress
IAntisnipe (iface)
assureCanTransfer

Summary

Virtual Versions contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. Virtual Versions 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 has access to an external contract. If the owner's credentials are compromised, then the contract could harm the transactions. The team has acknowledged the issue.

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