Cyberscope

Audit Report

Venus Protocol

April 2025

Network	BSC
Address	0xcF6BB5389c92Bdda8a3747Ddb454cB7a64626C63
Audited by	© cyberscope

Table of Contents

Risk Classification

The criticality of findings in Cyberscope's smart contract audits is determined by evaluating multiple variables. The two primary variables are:

1. **Likelihood of Exploitation**: This considers how easily an attack can be executed, including the economic feasibility for an attacker.
2. **Impact of Exploitation**: This assesses the potential consequences of an attack, particularly in terms of the loss of funds or disruption to the contract's functionality.

Based on these variables, findings are categorized into the following severity levels:

1. **Critical**: Indicates a vulnerability that is both highly likely to be exploited and can result in significant fund loss or severe disruption. Immediate action is required to address these issues.
2. **Medium**: Refers to vulnerabilities that are either less likely to be exploited or would have a moderate impact if exploited. These issues should be addressed in due course to ensure overall contract security.
3. **Minor**: Involves vulnerabilities that are unlikely to be exploited and would have a minor impact. These findings should still be considered for resolution to maintain best practices in security.
4. **Informative**: Points out potential improvements or informational notes that do not pose an immediate risk. Addressing these can enhance the overall quality and robustness of the contract.

Severity	Likelihood / Impact of Exploitation
Critical	Highly Likely / High Impact
Medium	Less Likely / High Impact or Highly Likely/ Lower Impact
Minor / Informative	Unlikely / Low to no Impact

Review

Contract Name	XVS
Compiler Version	v0.5.17+commit.d19bba13
Optimization	200 runs
Explorer	https://bscscan.com/address/0xcf6bb5389c92bdda8a3747ddb454cb7a64626c63
Address	0xcf6bb5389c92bdda8a3747ddb454cb7a64626c63
Network	BSC
Symbol	XVS
Decimals	18
Total Supply	30.000.000

Audit Updates

Initial Audit	17 Apr 2025

Source Files

Filename	SHA256
XVS.sol	0667f5ac0274fc3d44ce7e6aa24d084a6ea5ea6e66f807a8e4935e3d7c82fdd4

Findings Breakdown

Critical	0
Medium	0
Minor / Informative	4

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	0	0	0	0
Medium	0	0	0	0
Minor / Informative	0	4	0	0

Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
	NWES	Nonconformity with ERC-20 Standard	Acknowledged
	ST	Stops Transactions	Acknowledged
	TUO	Type Usage Optimization	Acknowledged
	L19	Stable Compiler Version	Acknowledged

NWES - Nonconformity with ERC-20 Standard

Criticality	Minor / Informative
Location	XVS.sol#L60,177
Status	Acknowledged

Description

The contract does not fully conform to the ERC20 Standard. Specifically, the standards state that the contract should have a function named `totalSupply` that returns the total supply of tokens.

```
uint public constant totalSupply = 30000000e18;
```

Additionally, the `transferFrom` method has a conditional statement that checks if the `msg.sender` is the actual sender of the tokens. If it is, the allowance is not being updated. While this is technically correct, it is possible that other decentralized platforms update their own local state according to the standard's specifications.

```
function transferFrom(address src, address dst, uint rawAmount)
external validLock returns (bool) {
    //...
    if (spender != src && spenderAllowance != uint96(-1)) {
        uint96 newAllowance = sub96(spenderAllowance, amount,
"XVS::transferFrom: transfer amount exceeds spender
allowance");
        allowances[src][spender] = newAllowance;

        emit Approval(src, spender, newAllowance);
    }
    //...
}
```

Recommendation

The incorrect implementation of the ERC20 standard could potentially lead to problems when interacting with the contract, as other contracts or applications that expect the ERC20 interface may not behave as expected. The team is advised to review and revise the implementation of the transfer mechanism to ensure full compliance with the ERC20 standard. https://eips.ethereum.org/EIPS/eip-20.

Team Update

The team has acknowledged that this is not a security issue and states: *There is a totalSupply function in the ABI, automatically generated because the constant is public.*

The team also states: *It is relevant to consider that the deployed XVS contract is not upgradeable, and it is battle tested.*

ST - Stops Transactions

Criticality	Minor / Informative
Location	XVS.sol#L31,36,159,172
Status	Acknowledged

Description

The contract owner has the authority to stop the sales for all users. The owner may take advantage of it by setting the `isLocked` to one. As a result, the contract may operate as a honeypot.

```
modifier validLock {
    require(isLocked == 0, "Token is locked");
    _;
}

function freeze() public onlyOwner {
    isLocked = 1;
    emit Freezed();
}

function transfer(address dst, uint rawAmount) external
validLock returns (bool)

function transferFrom(address src, address dst, uint rawAmount)
external validLock returns (bool)
```

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.

Temporary Solutions:

These measurements do not decrease the severity of the finding

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-signature wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

Permanent Solution:

- Renouncing the ownership, which will eliminate the threats but it is non-reversible.

Team Update

The team has acknowledged that this is not a security issue and states: *The current owner is a multisig wallet.*

TUO - Type Usage Optimization

Criticality	Minor / Informative
Location	XVS.sol#L26
Status	Acknowledged

Description

The `Owned` contract is using a `uint8` as a state variable to determine if the contract is frozen or not. However, `uint8s` are more expensive than `uint256s` because write operations emit an extra SLOAD to first read the slot's contents, replace the bits taken up by the `uint8` , and then write back.

```
uint8 isLocked = 0;

function freeze() public onlyOwner {
    isLocked = 1;
    emit Freezed();
}

function unfreeze() public onlyOwner {
    isLocked = 0;
    emit UnFreezed();
}
```

Recommendation

It is recommended to use `uint256s` for state variables instead of `uint8s` to reduce gas costs upon write operations.

Team Update

The team has acknowledged that this is not a security issue and states: *It is relevant to consider that the deployed XVS contract is not upgradeable, and it is battle tested.*

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	XVS.sol#L1
Status	Acknowledged

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 ^0.5.16;
```

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.

Team Update

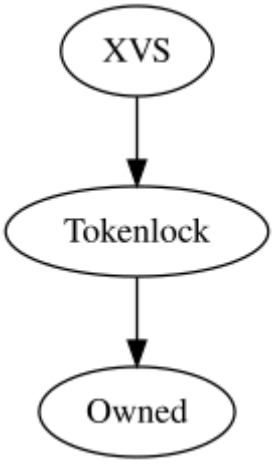
The team has acknowledged that this is not a security issue and states: *It is relevant to consider that the deployed XVS contract is not upgradeable, and it is battle tested.*

Functions Analysis

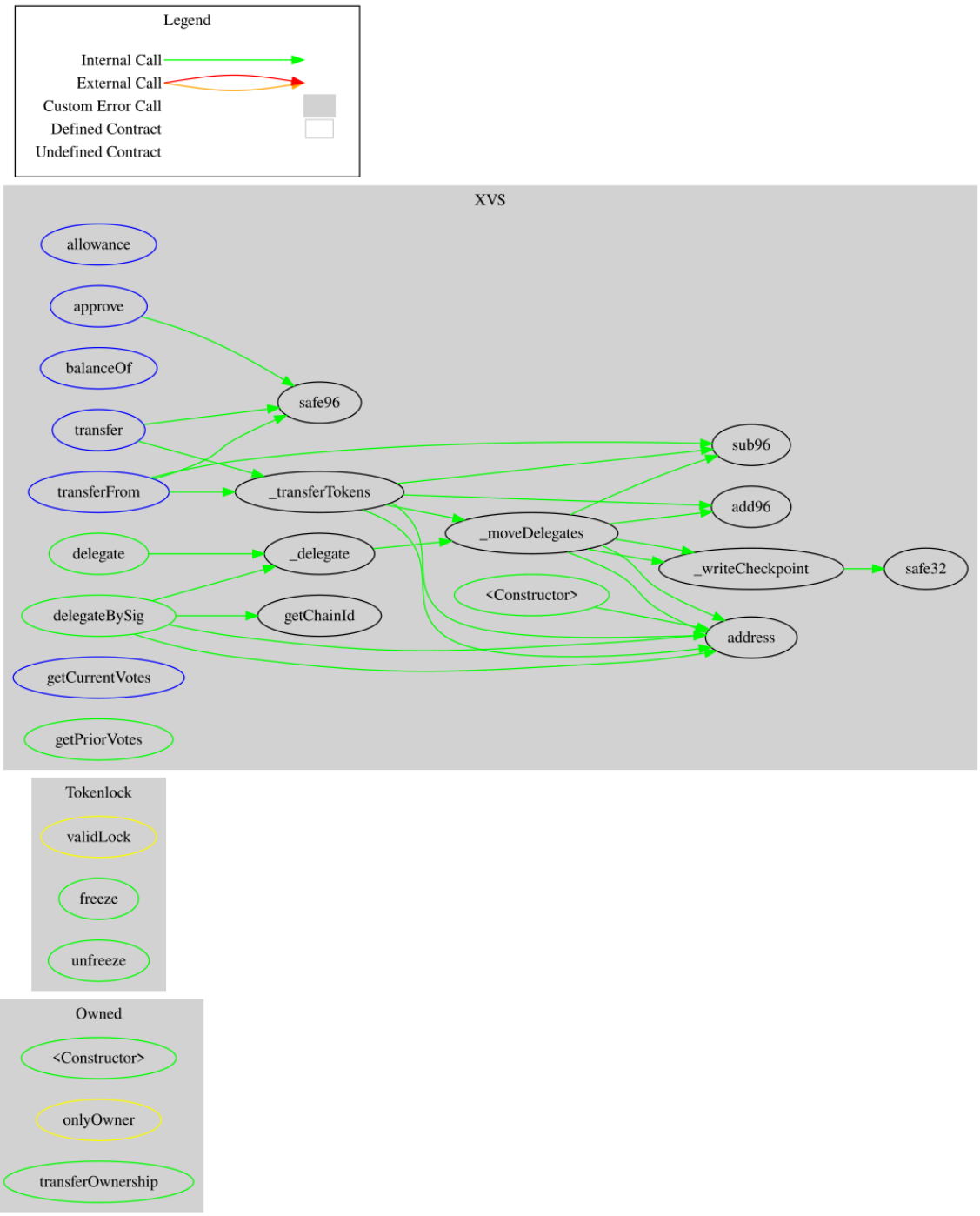
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Owned	Implementation			
		Public	✓	-
	transferOwnership	Public	✓	onlyOwner
Tokenlock	Implementation	Owned		
	freeze	Public	✓	onlyOwner
	unfreeze	Public	✓	onlyOwner
XVS	Implementation	Tokenlock		
		Public	✓	-
	allowance	External		-
	approve	External	✓	validLock
	balanceOf	External		-
	transfer	External	✓	validLock
	transferFrom	External	✓	validLock
	delegate	Public	✓	validLock
	delegateBySig	Public	✓	validLock
	getCurrentVotes	External		-
	getPriorVotes	Public		-
	_delegate	Internal	✓	

	_transferTokens	Internal	✓	
	_moveDelegates	Internal	✓	
	_writeCheckpoint	Internal	✓	
	safe32	Internal		
	safe96	Internal		
	add96	Internal		
	sub96	Internal		
	getChainId	Internal		

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Custom Error Call
Defined Contract
Undefined Contract
XVS
allowance
approve
balanceOf
transfer
transferFrom
delegate
delegateBySig
getCurrentVotes
getPriorVotes
safe96
_transferTokens
_delegate
getChainId
sub96
add96
_moveDelegates
_writeCheckpoint
safe32
<Constructor>
address
Tokenlock
validLock
freeze
unfreeze
Owned
<Constructor>
onlyOwner
transferOwnership

Summary

Venus Protocol is an interesting project that has a friendly and growing community. 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

cyberscope.io