



BLOCKCHAIN SECURITY ANALYSIS GROUP

PVU CEO

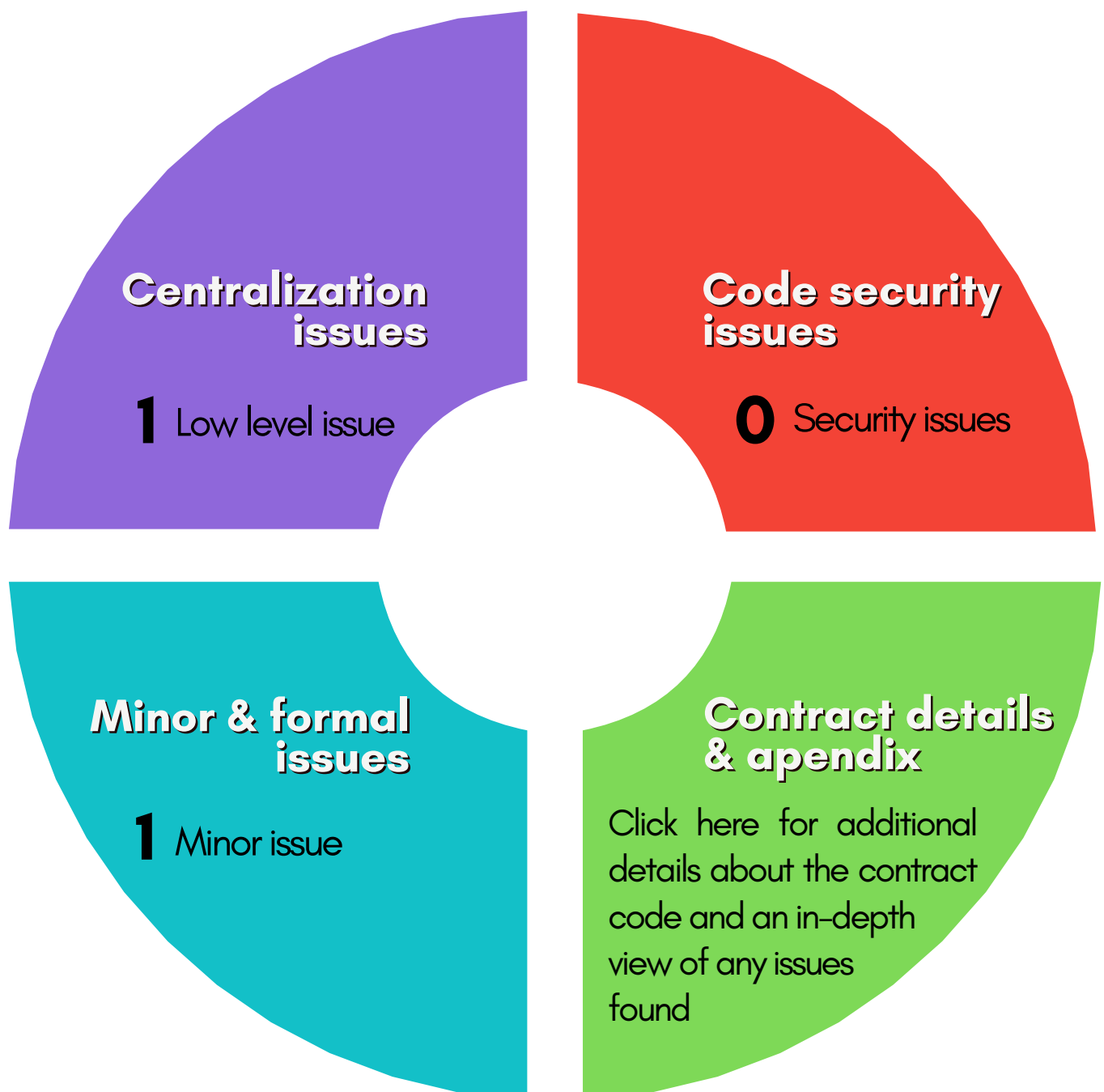
Full contract Audit &
Code security assessment

Jun 2023
v0.2

Audit results quick-view and navigation hub

Here you can see an overview of the smart contract code audit

Click on the different colored sections to go to the relevant audit pages for an in-depth report of any issues found.



Centralization issues

The PVUCEO.sol contract has a very **low** level of centralization.

CE1: Owner can modify sell tax up to 15%

What does this mean?

BOwner can set Sell tax to any value from 0 to 15%. while this function can't be used to make the contract a honeypot, it stills gives some degree of administrative privilege to the owner wallet. Buy tax is always zero in this token

We recommend renouncing ownership after setting up final tax values

[Click here or go to page 8 on the technical data appendix to see the relevant contract code and read additional details](#)

Overall centralization status:

Centralized fuction	Status	Comments
Owner can't mint?	✓	
Owner can't burn?	✓	
Owner can't pause trades?	✓	
Owner can't set blacklist?	✓	
Owner can't change sell tax?	✗	<u>Centralization issue (CE1)</u>
Owner can't change buy tax?	✗	<u>Centralization issue (CE1)</u>
Owner can't set max wallet size?	✓	
Owner can't set max Tx amount?	✓	

Minor and formal issues

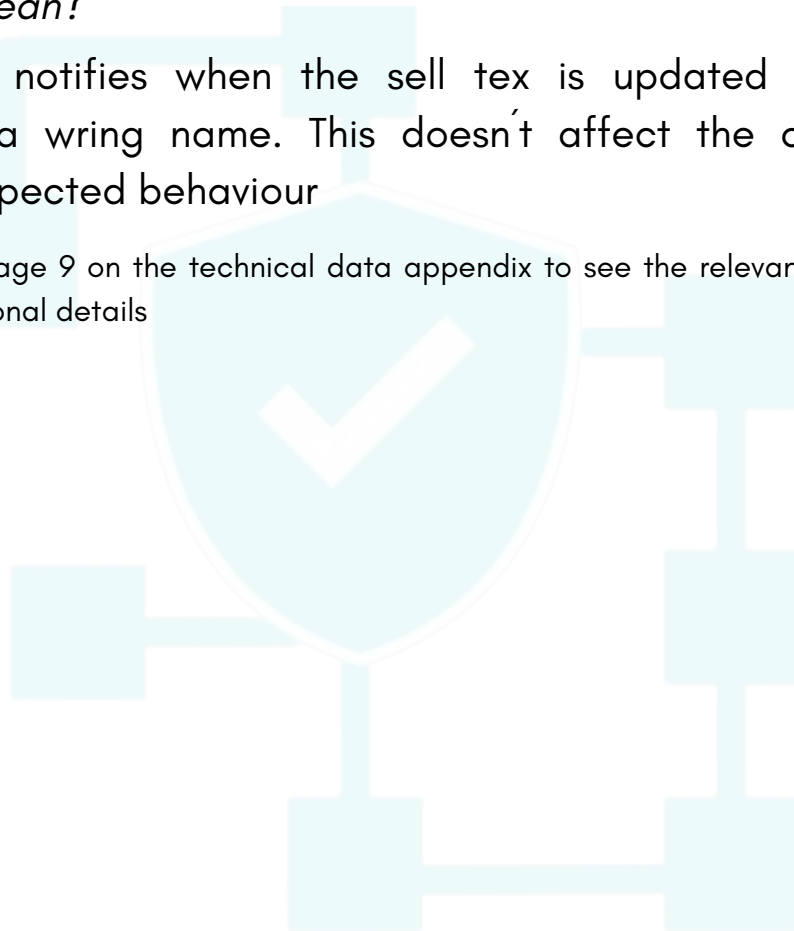
There are 1 issues in the formal aspects and coding style in the contract. These don't represent any risk for the security or correct functioning of the contract

MI1 | Invalid argument in TaxUpdated Event

What does this mean?

The event that notifies when the sell tex is updated have a argument with a wring name. This doesn't affect the contract excecution or expected behaviour

Click [here](#) or go to page 9 on the technical data appendix to see the relevant contract code and read additional details



Code security issues

After exhaustive analysis by automated tools, AI tools and manual line-by-line code review by our security experts, we didn't find any known security vulnerabilities in the PVUCEO.sol Smart Contract

0

Issues found



Technical details and apendix

In the following sections you will find additional details about the contract code as well as an in-depth view of any issues found, including the relevant code sections



Overview


Project details

Name	PVU CEO
Web	https://www.pvuceo.com/
Blockchain	Binance Smart Chain
Environment	EVM
Language	Solidity
Compiler version	0.8.10
Contract's Source	Contract deployed at bsc address: 0xE31D5E8798deed165A133B0584EfDb88CD1D06F9
Contracts Audited	PVUCEO.sol
Direct imports (contracts and libraries)	Context.sol, ERC20.sol, IERC20.sol, IERC20Metadata.sol, Ownable.sol,
Methodology used	Automated Static analysis Computer assisted code review Manual line-by-line code review Test deployment and stress testing

Variable List

Variable	Scope and additional details
poolAddress	Public, mapping(address => bool)
TAXWALLET	Public, constant
DIVISOR	Private, constant
MAXTAX	Private, constant
sellTax	Public
projectBuyTax	Public
projectSellTax	Public

Event List

Events	Emitted?
SetPool	
TaxUpdated	<u>Minor Issue (MI2)</u>

Function list

Type	Name	Results
Constructor	constructor	No issues found
External onlyOwner	updateTax	<u>Centralization issue (CE1)</u>
Public onlyOwner	setPool	No issues found
Internal	_transfer	No issues found
Internal	_beforeTokenTransfer	No issues found

CE1 | Owner can change sell tax up to 15%

Function `updateTax` gives any wallet with the owner role the privilege to change sell tax up to 15%.
(There is no buy tax).

```
uint private constant DIVISOR = 1000;  
//Using 1000 instead of 100 as divisor to enable fractional tax changes,  
uint private constant MAXTAX = 150; //establishes a maximum tax of 15%  
uint public sellTax = 90; //Initial 9% tax  
  
event SetPool(address poolAddress, bool status);  
event TaxUpdated(uint256 selltax);  
  
constructor() ERC20("PVU CEO", "PVUCEO") {  
    _mint(msg.sender, 1000000000 ether);  
}  
  
function updateTax(uint256 _sellTax) external onlyOwner {  
    sellTax = _sellTax;  
    require(_sellTax < MAXTAX);  
    emit TaxUpdated(sellTax);  
}
```

While the 15% limit mitigate the risk of abuse, the function still give some administrative privilege to the owner wallet
We recommend setting the tax to an appropriate level and then renouncing the ownership of the contract to avoid any potential abuse of the administrative privileges

Status: Acknowledged:

The team claims that the nature of their ecosystem requires constant updates to the tax value, so this implementation will remain, and as a mitigation measure, they will establish a system of public announcement whenever taxes need to be updated

MI1 | Invalid argument in TaxUpdated Event

Event TaxUpdated have an invalid argument in line 522. it should read `selltax`, as in the associated function, instead it reads `sellTax`

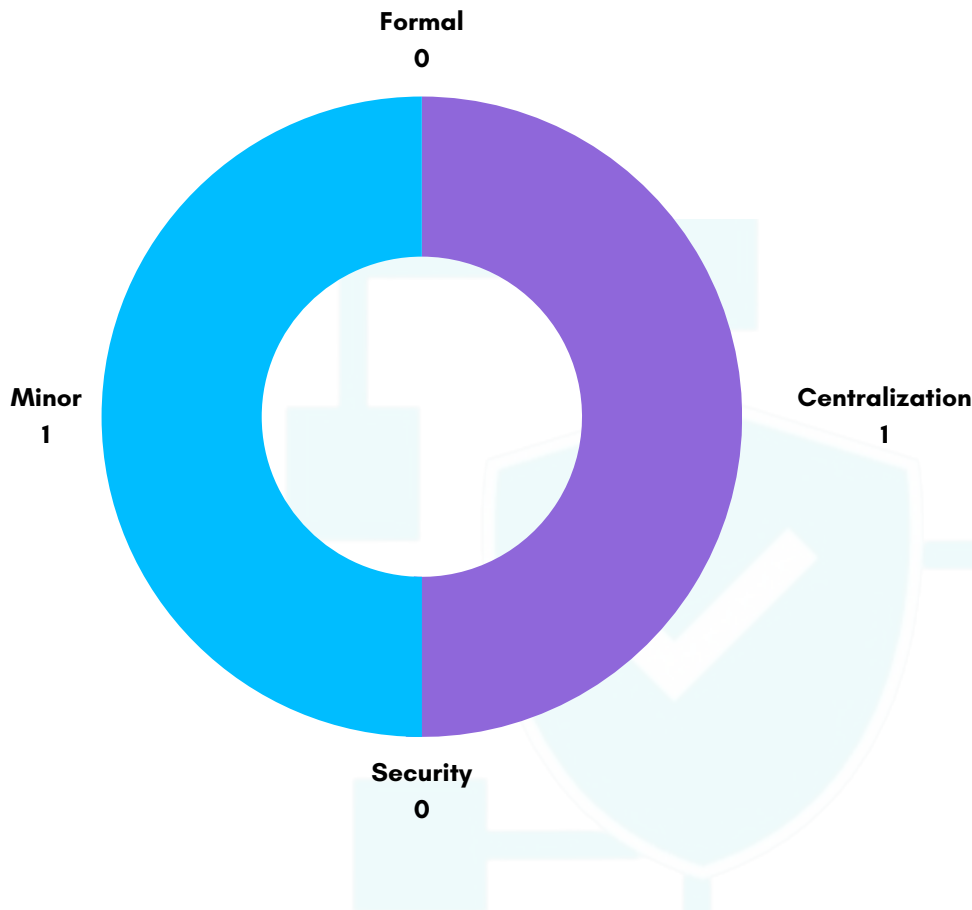
```
516     uint private constant DIVISOR = 1000;  
517     //Using 1000 instead of 100 as divisor to enable fractional tax changes, Please note this makes sellTax 90 =  
518     uint private constant MAXTAX = 150; //establishes a maximum tax of 15%  
519     uint public sellTax = 90; //Initial 9% tax  
520  
521     event SetPool(address poolAddress, bool status);  
522     event TaxUpdated(uint256 selltax);  
523  
524  
525     constructor() ERC20("PVU CEO", "PVUCEO") {  
526         _mint(msg.sender, 1000000000 ether);  
527     }  
528  
529     function updateTax(uint256 _sellTax) external onlyOwner {  
530         sellTax = _sellTax;  
531         require(_sellTax < MAXTAX);  
532         emit TaxUpdated(sellTax);  
533     }
```

This doesn't affect the contract execution or expected behaviour but leave event TaxUpdated without effect as no argument gets passed

Status: Acknowledged:

Since this issue doesn't represent a significant threat to the security of the project, contract or users, the team has decided to keep current implementation

Findings summary



Formal: Comments on coding style and best practices issues. Mostly subjective.

Minor: Low-risk issues that cannot harm the contract's execution or expected behaviour.

Code Security High-risk issues that can seriously harm the contract or compromise the ecosystem's security.

Centralization: Excess of centralized privilege can potentially represent an unfair playground for holders and investors.

PVUCEO.sol contract findings:

Severity Level	Found	Objected	Acknowledged	Mitigated	Resolved
Formal	1	0	1	0	0
Minor	1	0	1	0	0
Centralization	1	0	1	0	0
Critical	0	0	0	0	0

Synopsis

On June 1st 2023 the PVUCEO development team formally requested our services to evaluate its smart contract for vulnerabilities or issues, as well as the soundness of its code architecture.

Our analysis unit conducted both automated and manual analysis, including line-by-line examination by our team of coding experts, checking the whole code for potential vulnerabilities, centralization issues, unused/redundant code, optimization opportunities, compiling errors, timestamp/order dependencies, reentrancy errors, known attack vectors and coding best practices.

No critical issues or vulnerabilities were found in the code of the smart contracts analyzed. Some minor issues and certain degree of centralized privileges were detected.

This document describes in detail any findings as well as the mitigating measures taken by the team or those already present in the code (when applicable).

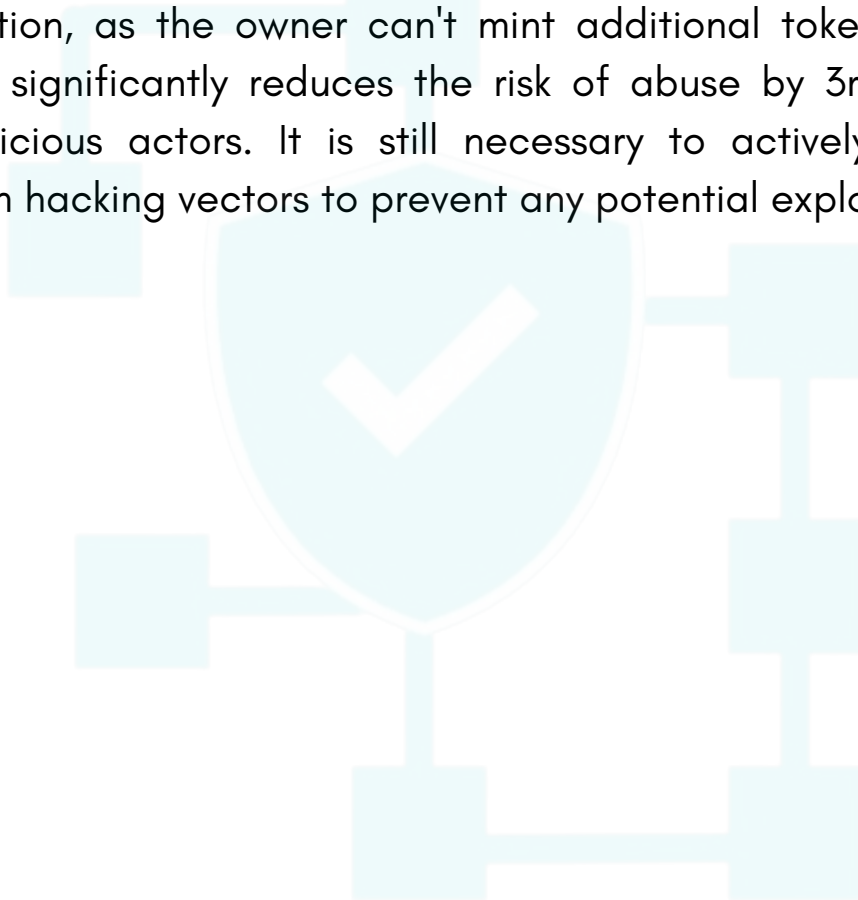
.

Final considerations

In our analysis of the PVUCEO.sol token contract we found no security issues.

While the code doesn't seem to have any vulnerabilities that could be exploited by 3rd parties, there are some minor centralization issues.

However, all centralized functions with `onlyOwner` require have limited scope of action, as the owner can't mint additional tokens or pause trading. This significantly reduces the risk of abuse by 3rd parties or external malicious actors. It is still necessary to actively guard the backend from hacking vectors to prevent any potential exploits.



Version history

Version	Changelog
v0.1 (internal)	<ul style="list-style-type: none">• Initial review of contract PVUCEO.sol
v0.2 First Public Release	<ul style="list-style-type: none">• Discussion of results with development team• Mitigation options evaluated• Emission of Audit certificate• Report publication

Disclaimer

The information here provided is not, and must not be considered, endorsement, approval or disapproval of the audited project.

This audit report DOES NOT CONSTITUTE INVESTMENT ADVICE. Its scope is limited to the technical and centralization aspects of the submitted Smart Contracts.

Our team HAS NOT MADE ANY EVALUATION of the project's viability or economic design. Neither do we make any claims about the development team's ability, proficiency or well meaning.

The scope of this review DOES NOT INCLUDE neither the dapp nor the backend of any web3 included in the project design that interacts with the analyzed smart contract, and in consequence we cannot assure their security.

This audit does not constitute any warranty of the absolute bug-free nature of the code or associated technologies used. Neither can it foresee any unwanted results of its interaction with 3rd party solutions like exchanges, in-game databases, or others.

Crypto Alert NFT is in NO WAY RESPONSIBLE for any harm, malfunction or asset loss you might incur while interacting with the smart contracts here evaluated, nor is in any way liable for any detrimental results from your interaction with any aspects of the evaluated project.

Jun 2023



BLOCKCHAIN SECURITY ANALYSIS GROUP



<https://t.me/cryptoalertnft> (community).

<https://www.cryptoalertnft.com/> (web).

