Cyberscope

Audit Report

BANDIT

October 2024

File

BANDIT.sol

SHA256

6f275d6da7320c36178c3941898e1df6a63fec4374b741e7517e90f09e6c10d3

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

Badge Eligibility	Must Fix Criticals

Audit Updates

Initial Audit	23 Oct 2024 https://github.com/cyberscope-io/audits/blob/main/bandit/v1/audit.pdf
Corrected Phase 2	29 Oct 2024
Testing Deploy	https://sepolia.etherscan.io/address/0xb37eb3a12a1ABBE4604144Be70b8932640d2ec24

Source Files

Filename	SHA256
BANDIT.sol	6f275d6da7320c36178c3941898e1df6a63fec4374b741e7517e90f09e6c10d3

Analysis

Critical Medium Minor / Informative Pass

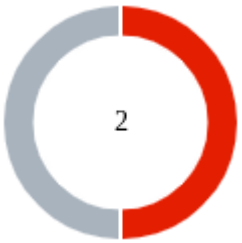
Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
Critical	UPA	Unexcluded Pinksale Address	Unresolved
Minor / Informative	MEE	Misleading Event Emission	Unresolved

Findings Breakdown



Critical	1
Medium	0
Minor / Informative	1

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

UPA - Unexcluded Pinksale Address

Criticality	Critical
Location	BANDIT.sol#L58
Status	Unresolved

Description

The contract incorporates operational restrictions on transactions, which can hinder seamless interaction with decentralised applications (dApps) such as launchpads, presales, lockers, or staking platforms. In scenarios where an external contract, such as a launchpad factory, needs to integrate with the contract, it should be exempt from the limitations to ensure uninterrupted service and functionality. Failure to provide such exemptions can block the successful process and operation of services reliant on this contract.

In particular, in the current implementation, it is possible that fees are applied to decentralized applications such as launchpads and presales. Fees may be applied if:

- Tokens are sent to the zero address prior to the pair creation.
- Tokens are sent to the pair address after the pair creation.

Both scenarios can prevent DApps like Pinksale from functioning properly, potentially leading to loss of funds.

```
if (swapPair == to || swapPair == from) {
    if (
    balanceOf(address(this)) > balanceOf(swapPair) *
    10 / 10000 && // 0.1% of swapPair holdings
    swapPair == to
    ) handleTax();
    uint256 extraFee = value * tax / 10000;
    super._update(from, address(this), extraFee);
    value -= extraFee;
}
```

Recommendation

It is advisable to modify the contract by incorporating functionality that enables the exclusion of designated addresses from transactional restrictions. This enhancement will allow specific addresses, such as those associated with decentralized applications (dApps) and service platforms, to operate without being hindered by the standard constraints imposed on other users. Implementing this feature will ensure smoother integration and functionality with external systems, thereby expanding the contract's versatility and effectiveness in diverse operational environments.

MEE - Misleading Event Emission

Criticality	Minor / Informative
Location	BANDIT.sol#L118
Status	Unresolved

Description

Misleading event emissions occur when events do not accurately reflect the contract's state changes. This can confuse users and systems relying on these events, leading to incorrect assumptions about the contract's functionality. Such discrepancies undermine trust and complicate system integrations.

```
function setPairAndTax() private {
swapPair = IFactory(ROUTER.factory()).getPair(
    address(this),
    WETH
    );
    if (balanceOf(swapPair) != 0) {
        isLaunched = true;
        tax = 100;
    }
    emit SetPair(swapPair);
}
```

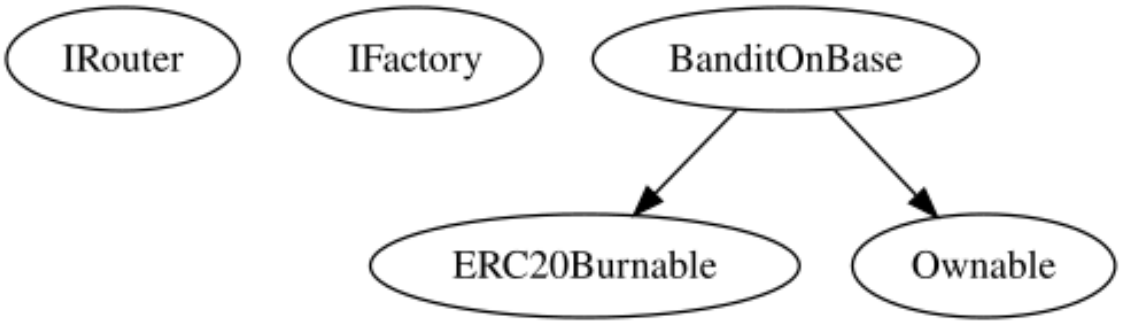
Recommendation

It is always a good practice for a contract to emit events that are specific and descriptive of changes in its internal state. The team is advised to ensure consistency and reliability in the code.

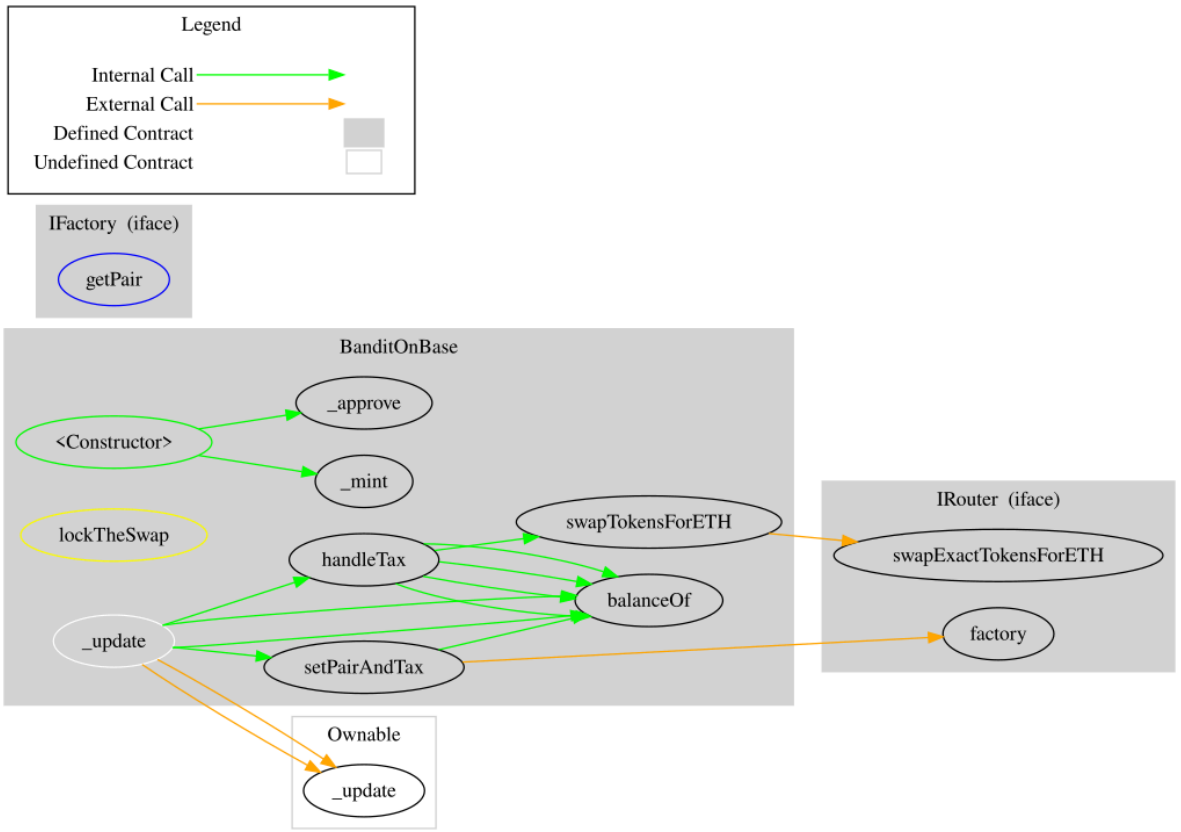
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IRouter	Interface			
	factory	External		-
	swapExactTokensForETH	External	✓	-
IFactory	Interface			
	getPair	External		-
BanditOnBase	Implementation	ERC20Burnable, Ownable		
		Public	✓	ERC20 Ownable
	_update	Internal	✓	
	handleTax	Private	✓	lockTheSwap
	swapTokensForETH	Private	✓	

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
IFactory (iface)
getPair
BanditOnBase
_approve
<Constructor>
_mint
lockTheSwap
swapTokensForETH
handleTax
balanceOf
_update
setPairAndTax
IRouter (iface)
swapExactTokensForETH
factory
Ownable
_update

Summary

Bandit contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. Bandit is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error and a critical issue. There is also a max fee of 1%.

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