callisto

Projecton Audit Report

Contents

1. Summary

Projecton smart contract security audit report performed by Callisto Security Audit Department

Token description:

```
Symbol       : XN35
Name         : Projecton
Total supply: 100,000,000
Decimals     : 18
Standard     : ERC20
```

2. In scope

- LICENSE github commit hash fed7803c48d517f5aee9d787e54d3bebf3106d44.

3. Findings

In total, **6 issues** were reported including:

- 3 low severity issues.
- 3 minor observation.

No critical security issues were found.

3.1. Zero address checking.

Severity: low

Description

In functions `transferOwnership`, `transfer` and `transferFrom` there are possibility of passed address being zero address.

Code snippet

https://github.com/Projecton13/XN35_Standard_Tokens/blob/master/LICENSE#L50

https://github.com/Projecton13/XN35_Standard_Tokens/blob/master/LICENSE#L82

https://github.com/Projecton13/XN35_Standard_Tokens/blob/master/LICENSE#L94

Recommendation

Need to check if newOwner address is not zero address.

```
require(newOwner != address(0));
```

And add to `transfer` and `transferFrom`

```
require(_to != address(0));
```

Code snippet

https://github.com/danbogd/XN35_Standard_Tokens/blob/fed7803c48d517f5aee9d787e54d3bebf3106d44/LICENSE#L50

3.2. ERC20 Compliance.

Severity: low

Description

According to ERC20 standard, when initializing a token contract if any token value is set to any given address a transfer event should be emitted. An event isn't emitted when assigning the initial supply to the msg.sender.

Code snippet

https://github.com/danbogd/XN35_Standard_Tokens/blob/fed7803c48d517f5aee9d787e54d3bebf3106d44/LICENSE#L71

3.3. Known vulnerabilities of ERC-20 token

Severity: low

Description

1. It is possible to double withdrawal attack. More details here.
2. Lack of transaction handling mechanism issue. WARNING! This is a very common issue and it already caused millions of dollars losses for lots of token users! More details here.

Recommendation

Add into a function `transfer(address _to, ... )` following code:

```
require( _to != address(this) );
```

3.4. Constant functions have incorrect type

Severity: minor observation

Description

Constant functions have incorrect type. They should be changed to view type instead. Starting from Solidity 0.4.16, functions that do not modify the blockchain can have two modifiers: pure or view. Pure functions cannot modify or read data from blockchain. View functions have the same semantics as constant, but they can read data from blockchain.

3.5. Provide meaningful error messages for every exception.

Severity: minor observation

Description

It is recommended to provide meaningful error messages along with each require statement. This will help the user to understand what went wrong more easily since there are many validations happening for each buy.

3.6. Consider using latest version of solidity.

Severity: minor observation

Description

The contracts use solidity version 0.4.25. It is suggested to use the latest version and fix all compiler warnings that arise. Compiler version should be fixed to avoid any potential discrepancies in smart contract behavior caused by different versions of compiler.

4. Conclusion

The review did not show any critical issues, some of low severity issues were found.

5. Revealing audit reports

https://gist.github.com/yuriy77k/7e4e88815ffaa7b653bb1ecdfbe8c840

https://gist.github.com/yuriy77k/db0b3aac8e4ea99ee93a8d0ad7601bd0

https://gist.github.com/yuriy77k/f2ac4f8e9eb0cab6c9e67adb3746508f