Pyxis Network Security Audit Report

1. Summary

Pyxis Network smart contract security audit report performed by Callisto Security Audit Department

- website: https://www.pyxis.network/
- telegram: https://t.me/pyxiscommunity

2. In scope

Commit 864c59458a2b22b22f57b9df92e931dee8260dd3

- CMPSReservation.sol
- CMPSToPYXSwapper.sol
- CMPSToken.sol
- ETHAutoStaking.sol
- PYXStaking.sol
- PYXToken.sol
- interfaces/IAutoStaking.sol
- interfaces/ICMPSToken.sol
- interfaces/IPYXStaking.sol
- interfaces/IPYXToken.sol

2.1. Excluded

The smart contract use open source library from Openzeppelin. Following files was excluded from audit:

- @openzeppelin/contracts/access/AccessControl.sol
- @openzeppelin/contracts/math/SafeMath.sol
- @openzeppelin/contracts/token/ERC20/ERC20.sol
- @openzeppelin/contracts/token/ERC20/IERC20.sol
- @openzeppelin/contracts/utils/structs/EnumerableSet.sol

Interface contract of Uniswap Router:

- @uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router02.sol

3. Findings

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

- 2 low severity issues.
- 1 notes.
- 3 owner privileges.

No critical security issues were found.

3.1. The known vulnerability of the ERC-20 token

Severity: low

Description

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 the following code to the transfer() function:

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

3.2. Default Admin Role

Severity: owner privileges

Description

The contract deployer gets `DEFAULT_ADMIN_ROLE`. This means that he can grant any roles to any addresses. It makes the smart contract more centralized and is risky if that private key will be compromised.

1. When you revoke `SETTER_ROLE` in the function `init()` it does not guarantee that settings could not be changed in the future because a user with `DEFAULT_ADMIN_ROLE` can grant `SETTER_ROLE` to any address and change the settings again (may call the function `init()`).
2. A user with `DEFAULT_ADMIN_ROLE` can grant `MINTER_ROLE` to any address and mint unlimited CMPS tokens and burn CMPS tokens from any address. The same according to PYX tokens mint and burn too.

Code snippet

- Setup `DEFAULT_ADMIN_ROLE`: https://github.com/pyxiscto/pyxis-smartcontracts/blob/4344d2b1d2fff2095a78ebf8398b9deb39bab995/contracts/CMPSReservation.sol#L60

https://github.com/pyxiscto/pyxis-smartcontracts/blob/4344d2b1d2fff2095a78ebf8398b9deb39bab995/contracts/CMPSToken.sol#L35

https://github.com/pyxiscto/pyxis-smartcontracts/blob/4344d2b1d2fff2095a78ebf8398b9deb39bab995/contracts/CMPSToPYXSwapper.sol#L60

https://github.com/pyxiscto/pyxis-smartcontracts/blob/4344d2b1d2fff2095a78ebf8398b9deb39bab995/contracts/PYXToken.sol#L88

https://github.com/pyxiscto/pyxis-smartcontracts/blob/4344d2b1d2fff2095a78ebf8398b9deb39bab995/contracts/ETHAutoStaking.sol#L110

https://github.com/pyxiscto/pyxis-smartcontracts/blob/4344d2b1d2fff2095a78ebf8398b9deb39bab995/contracts/PYXStaking.sol#L147

- Revoke `SETTER_ROLE`:

https://github.com/pyxiscto/pyxis-smartcontracts/blob/4344d2b1d2fff2095a78ebf8398b9deb39bab995/contracts/CMPSReservation.sol#L81

https://github.com/pyxiscto/pyxis-smartcontracts/blob/4344d2b1d2fff2095a78ebf8398b9deb39bab995/contracts/CMPSToken.sol#L46

https://github.com/pyxiscto/pyxis-smartcontracts/blob/4344d2b1d2fff2095a78ebf8398b9deb39bab995/contracts/CMPSToPYXSwapper.sol#L82

https://github.com/pyxiscto/pyxis-smartcontracts/blob/4344d2b1d2fff2095a78ebf8398b9deb39bab995/contracts/PYXToken.sol#L113

https://github.com/pyxiscto/pyxis-smartcontracts/blob/4344d2b1d2fff2095a78ebf8398b9deb39bab995/contracts/ETHAutoStaking.sol#L149

https://github.com/pyxiscto/pyxis-smartcontracts/blob/4344d2b1d2fff2095a78ebf8398b9deb39bab995/contracts/PYXStaking.sol#L185

Recommendation

Do not set up `DEFAULT_ADMIN_ROLE`.

3.3. `SETTINGS_MANAGER_ROLE` can be assigned only by Default Admin

Severity: owner privileges

Description

In the contracts `PYXStaking`, `ETHAutoStaking`, `PYXToken` the `SETTINGS_MANAGER_ROLE` can be assigned only by the user with `DEFAULT_ADMIN_ROLE`. As was pointed in 3.2. the user has unlimited power which is risky if his private key will be compromised. We suggest to use `SETTINGS_MANAGER_ROLE_ADMIN` instead `DEFAULT_ADMIN_ROLE` to assign `SETTINGS_MANAGER_ROLE` rights.

Recommendation

Add constant: `bytes32 public constant SETTINGS_MANAGER_ROLE_ADMIN = keccak256('SETTINGS_MANAGER_ROLE_ADMIN');`

In the constructor replace:

```
_setupRole(DEFAULT_ADMIN_ROLE, msg.sender);
```

with

```
_setupRole(SETTINGS_MANAGER_ROLE_ADMIN, msg.sender);
_setRoleAdmin(SETTINGS_MANAGER_ROLE, SETTINGS_MANAGER_ROLE_ADMIN);
```

3.4. Users with `SETTINGS_MANAGER_ROLE` can change constants.

Severity: owner privileges

Description

The constants for calculation formulas (like rewards calculation, penalty calculation, etc) can be changed at any time by the user with `SETTINGS_MANAGER_ROLE`. This means that the calculations may differ from those indicated in the white paper.

Recommendation

Use hardcoded constants instead of variables.

3.5. EnumerableSet is undeclared - compilation error

Severity: note

Description

You are using library `EnumerableSet` in the contracts PYXStaking and PYXToken but does not import `@openzeppelin/contracts/utils/structs/EnumerableSet.sol`. It cause compilation error.

Recommendation

Add `import '@openzeppelin/contracts/utils/structs/EnumerableSet.sol'` into `PYXStaking` and `PYXToken` contracts.

3.6. The function getNumDayInWeek() may returns wrong day number

Severity: low

Description

The results of function getNumDayInWeek() depends on value of `SETTINGS.STEP_SECONDS`. It will returns correct day number only if `SETTINGS.STEP_SECONDS = 86400`.

Recommendation

To remove this dependence better to use hardcoded value: `return (block.timestamp / 1 days) % 7;`

4. Conclusion

The audited smart contract can be deployed. Only low severity issues were found during the audit. Investors have to pay attention to high owner privileges.