# Preliminary Notes: Space Vikings


## Brewlabs Services Hub

**Summary:** SpaceViking.sol is a token smart contract for the SpaceVikings project. It is written in Solidity and compiled using build 0.6.12. The smart contract supports regenerative tokenomics.
**Audit package:** Standard
***Deployment Status:*** Deployed to BSC Mainnet
***Files Audited:*** SpaceVikings.sol
***Audit Result:*** **PASSED**

***Disclaimer:*** *This audit documentation is for discussion purposes only. The scope of this audit was to analyze and document SpaceVikings smart contract codebase for quality, security, and correctness. This audit guarantees that your code has been revised by an expert.*

### Overview of Audit

No critical security issues identified.
5 major security issues identified.
Several minor and informational issues identified.

### Security issues

#### Major level security issues detected:

1. Contract allows owners to withdraw the entire BNB balance in the contract at any time via the withdrawOverFlowBNB function. It is noted that the liquidity deposits made by the contract are sent to the contract address and therefore such a function is required in order to extract the value and commit to the pool. However, while the use case is understood, the design still poses a centralisation risk.
2. Contract allows owners to enable/disable liquidity deposits (swapEnabled) which can impact those adding to the liquidity pool
3. Contract allows owners to alter transaction fees.
4. Contract allows owners to exclude wallets (potentially permanently) from rewards, and to do so with no visibility to the public (no event emitted).

The above issues violate the ethics of Decentralised Finance and present a centralisation risk to the community. To mitigate these risks, ensure the Development team is doxxed or consider (Know Your Customer) KYC services.

5. No upper limit when setting fee state variables leading to potential centralisation risk if the contract owner sets the fee to 100%. Recommendation is to embed an upper limit or communicate to the community when relevant changes are committed to improve transparency.

#### Minor level security issues detected:

1. Missing events for changes to state variables in various functions (line 825-898). This means only the contract owner can alter state variables without the community being aware of changes. Recommendation is to add events for all changes to state variables or communicate to the community when relevant changes are committed to improve transparency.
2. Lack of input validation or commentary on fee changes to ensure that arguments are entered as x*10. Recommendation is to ensure owner is educated as to the required input format for changing fees and that the same format is not required for setMaxTxPercent function (requires x*1)

3. Lack of input validation on _maxTxAmount which could lead to loss in liquidity pool deposits, for example if the amount is set to zero.
4. Lack of input validation for setJackpotFeeWallet, setRaidWallet, setMarketingWallet which could lead to unexpected loss if an invalid address is set, for example, address(0).
5. SafeMath is not used for division (line 1184) and may lead to math underflow and overflow risks.
6. Visibility not explicitly set for variable inSwapAndLiquify (line 694). By default the visibility is internal. Development team to confirm if this is the correct visibility.

**Informational issues detected:**

1. Floating pragma set. If relying on byte-code level verification, best practice is to lock the compiler version prior to deployment.
2. Third party dependency on Pancake Swap protocol (line 1135, 1149). Recommendation is for the Dev team to monitor updates made to third party contracts to ensure changes to not impact the receiving contract or are mitigated/managed accordingly.
3. Variable names for IUniswapV2Router02, uniswapV2Pair, and uniswapV2Router (line 719-726) does not match actual protocol used (PancakeSwap).
4. Taxes (marketing, raid etc) are paid out from the BNB derived from the liquidity pool rather than the BNB gathered in the contract from taxes.
5. Unclear acronym 'ERR' in require return messages (line 1051). Development team to advise if this correct.

**Code Style:**

1. Camel case is used for all variable names, but not used for variable: _jackpotfeewallet.
2. Unused code (line 702, 881-883), could be removed to improve business logic clarity.
3. Bool set to False by default. Lines 695 and 696 could be declared as just: bool {visibility} {variable_name}; to reduce gas usage.
4. Low use of comments. Comments help to improve readability and is encouraged.

**Conclusion:** The Brewlabs team thank you for the opportunity to review and audit your smart contact code. The data from this report will be formalised in the audit publication for your community. Keep in touch as we offer discounts for repeat business and on a range of other services!

Maverick & Peach
The Brewlabs Team.