Executive Summary

The Unicake Bot has been tested manually by product experts from a functionality, product, UI and security standpoint to ensure overall experience.

Limitations

QuillAudits audit is not a security warranty, investment advice, or an endorsement of the Unicake platform. This audit does not provide a security or correctness guarantee of the audited smart contracts. The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them. Securing smart contracts is a multistep process. One audit cannot be considered enough. We recommend that the Unicake Team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.

Audit Overview

Scope of the Audit

QuillAudits has been communicated that the scope of this audit exercise is to assess the functionalities of the bot; ensure their intuitive usage by the user and to determine security risks pertaining to the storage of the private key of user funds.

The scope of the security audit conducted by QuillAudits was restricted to:

- Manual usage and trials of the bot listed above for functionality issues.
- Overview of the bot for generic UI issues and inspecting the results.

System Overview

Unicake Bot is a multiplatform trading bot, it allows users to trade options on ethereum and binance smartchain. It is compatible for Uniswap and Pancake Swap.

Unicake Features include-

I. Uniswap Mempool Sniping Bot (snipe on new listing)

II. Different combinations of orders (buy and sell) on Uniswap (Ethereum - ETH)

- Market Order
- Limit Order
- Market Order with Stop Loss
- Limit Order with Stop Loss
- Market Order with Target(take profit)
- Limit Order with Target(take profit)
- Market Order with Stop Loss and Target(take profit)
- Limit Order with Stop Loss and Target(take profit)

III. Different combination of orders (buy and sell) on Pancakeswap (Binance Smart Chain - BNB)

- Market Order
- Limit Order
- Market Order with Stop Loss
- Limit Order with Stop Loss
- Market Order with Target(take profit)
- Limit Order with Target(take profit)
- Market Order with Stop Loss and Target(take profit)
- Limit Order with Stop Loss and Target(take profit)

Best Practices in Unicake Bot

Projects of good quality follow best practices. In doing so, they make audits more meaningful, by allowing efforts to be focused on subtle and project-specific issues rather than the fulfilment of general guidelines. Avoiding code duplication is a good example of a good engineering practice which increases the potential of any security audit. We now list a few points that should be enforced in any good project.

Hard Requirements

- These requirements ensure that the Unicake Bot can be audited by QuillAudits.
- The code is provided as a Git repository to allow the review of future code changes.
- Code duplication is minimal, or justified and documented.
- Libraries are properly referred to as package dependencies, including the specific version(s) that are compatible with Unicake Bot's project.
- The code compiles with the latest Solidity compiler version. If Unicake Bot uses an older version, the reasons are documented.
- There are no compiler warnings, or warnings are documented.

Soft Requirements

- Although these requirements are not as important as the previous ones, they still help to make the audit more valuable to Unicake Bot.
- The tests are related to the migration scripts and a clear separation is made between the two.
- The test coverage is available or can be obtained easily.
- There is no dead code.
- The code is well documented.
- Functions are grouped together according to either to the guidelines or to their functionality.

Functional Analysis

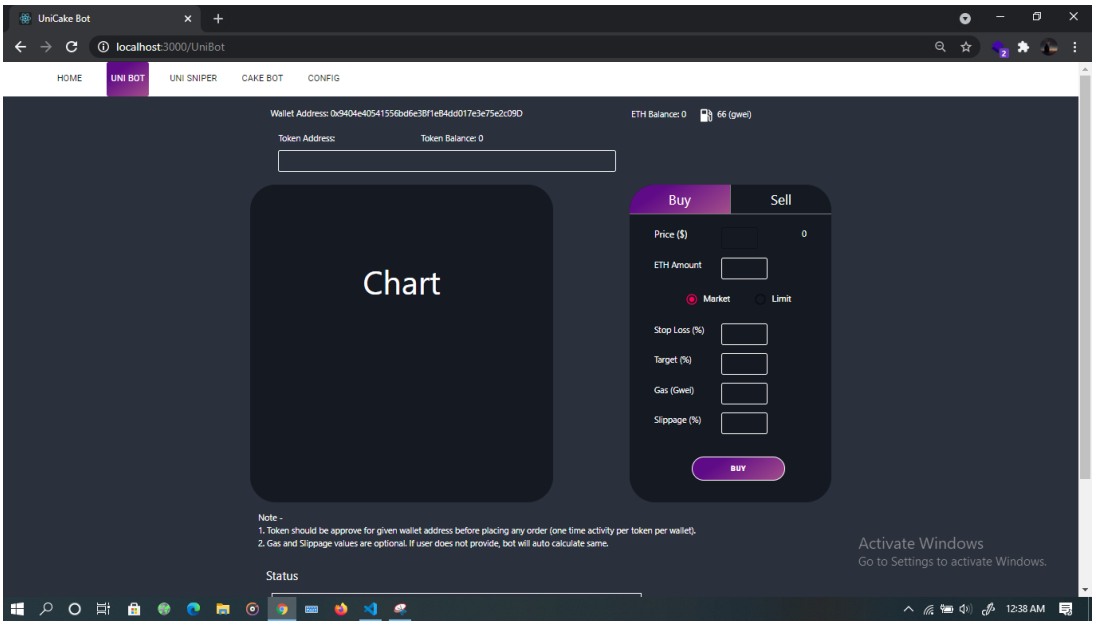
Uniswap Bot

Chart not working
No functional error in oder execution
Some functional suggestions are mentioned below

The feature works perfectly fine as stated. A limit buy and sell order can be placed using this. All the fields are customizable thus giving users the ability to make changes as desired.

For a better user experience the fields should have demo suggestive values for the user to understand easily.

No place to cancel order was found if the order is placed by mistake,
A 2fa should be provided before order execution.
Minimum required fields should be mentioned with an asterisk and mandatory fields should be kept to avoid misplaced orders.

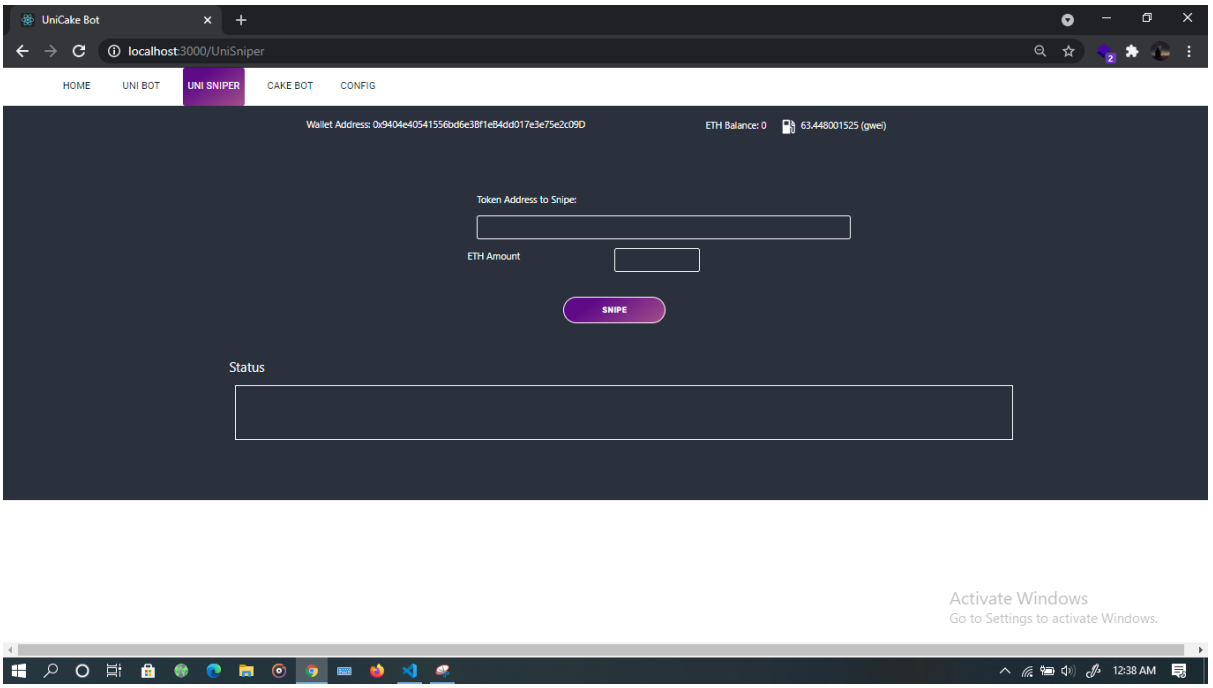


Sniper

Status not able to identify
No functional error in sniping

The feature works perfectly fine as stated. Snipining is working but for better understanding it would be good if different status codes should be provided in the description to be able to identify.

USD equivalent with the eth amount would be more useful for the user as trades are mostly done in usd terms.



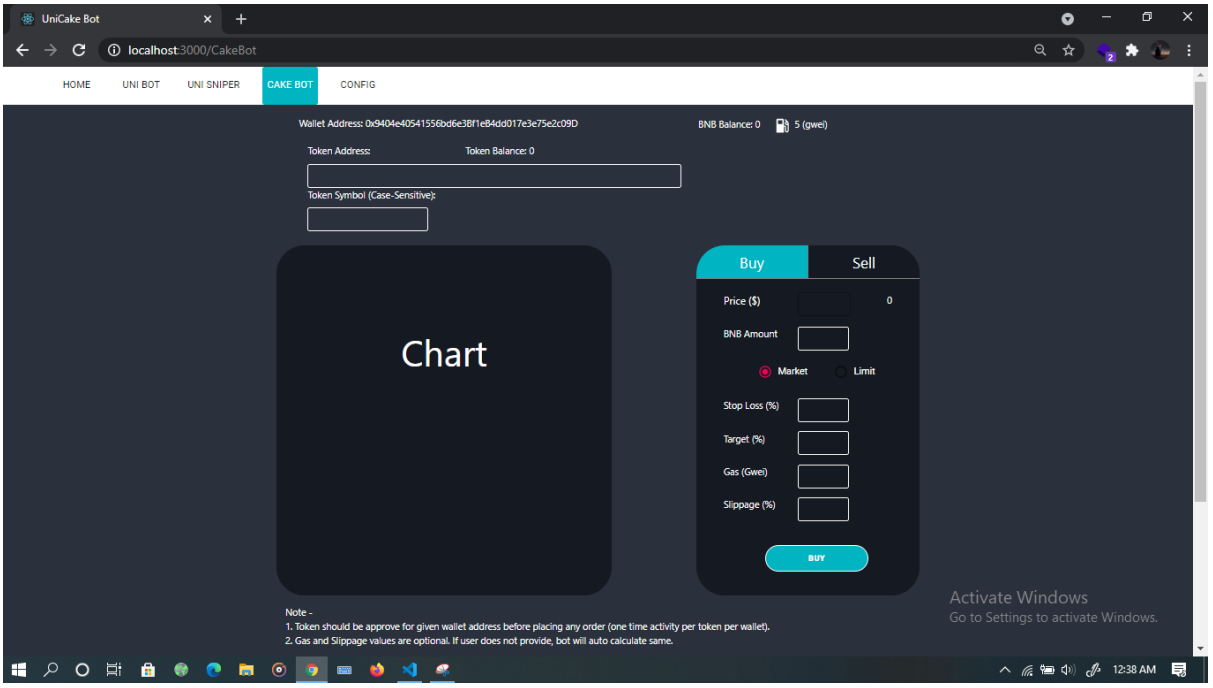
Cake Bot

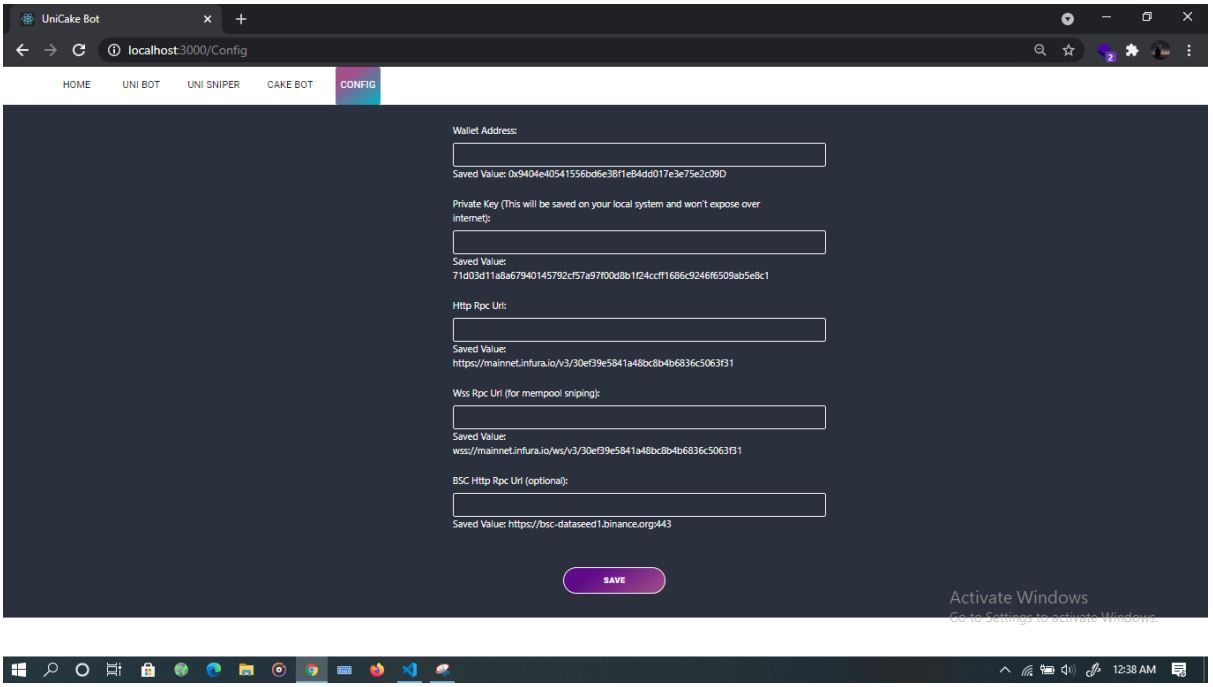
Chart not working
No functional error in oder execution
Some functional suggestions are mentioned below

The feature works perfectly fine as stated. A limit buy and sell order can be placed using this. All the fields are customizable thus giving users the ability to make changes as desired.

For a better user experience the fields should have demo suggestive values for the user to understand easily.

No place to cancel order was found if the order is placed by mistake,
A 2fa should be provided before order execution.
Minimum required fields should be mentioned with an asterisk and mandatory fields should be kept to avoid misplaced orders.





Conclusion

The testing was concluded on 4th of May ‘21 for Unicake Bot. The conclusion of the exercise undertaken by the QuillAudits team states that the bot has major features as; Maket Buy/Sell, Instant Buy & Sell and Sniping of new tokens. These functionalities can be accessed in a satisfactory manner without any glitches.

The private key of the user accessing the bot is stored locally. In no scenario is the key uploaded on cloud or any third party server which might put the funds at risk. This eliminates a major security risk and can be defined as a great security practice undertaken while developing the product.