

BSC Token Sniper – Setup + Info

BSTokenSniper is a bot written in Python to detect new PairCreated events in the Binance Smart Chain (when a liquidity pair has been created) and buy the token.

Description

The aim of BSC Token Sniper is to buy new tokens with a specified amount of BNB, with the aim of the price rising. Once the bot detects a PairCreated event, it is able to check the token (mini audit).

It can check if:

- Source code is verified.
- If valid PancakeSwap v2 router is being used
- If a mint function exists
- If it is a potential honeypot
- PancakeSwap v1 router address is not being used.

The user can decide whether to enable the mini audit or turn it off (bear in mind you will likely be investing in a lot of scams if you don't).

Once the token has/hasn't been through a mini audit the bot will then attempt to buy X amount of tokens with the specified amount of BNB.

The bot will buy the tokens directly through the Binance Smart Chain using the PancakeSwap v2 router and factory address, so it is much quicker than the PancakeSwap web interface.

By avoiding web interfaces & Metamask and directly with Ethereum & EVM Nodes you can snipe tokens faster than any of the web-based platforms. This allows tokens to be sniped almost instantly. During our testing we found the bot would typically be within the first 3 buy transactions of all tokens it finds.

The bot buys the tokens using the user's wallet address and private key. This information is kept secure, is only stored locally on your computer, and is only ever used to buy tokens (look through the code to see for yourself).

The bot does not incur any additional fees, only fees are BSC network transaction fees and PancakeSwap fees.

Prerequisites

- Python 3 or later installed
- Node.js installed (easiest way) – Install windows version from <https://nodejs.org/en/download/>
- Web3 installed (in windows command line type: *npm install Web3*)
- BscScan API key (completely free of charge, create an account on BscScan and generate a free API key)
- BSC wallet address and private key
- enough BNB in your wallet to snipe tokens.

Setup

- 1) Install all dependencies (above)
- 2) Edit config.json file with your wallet address, private key and BscScan API key.
- 3) (Optional) if you are in windows, open the command prompt and right-click the title bar, click 'properties' and set screen buffer size height to 2500. This allows you to scroll through the history of your token snipes.
- 4) Run 'launchBSCTokenSniper.bat' and you're good to go!

Configuration File

When you download the bot, you will find a config.json file. This is where you need to add the following data.

walletAddress: your BSC wallet address (e.g., Metamask)

walletPrivateKey: your private key of your wallet address (your private key is kept safe and not shared in any other place)

amountToSpendPerSnipe: The amount in BNB you want your wallet to spend on every new token. (e.g., 0.00025 means a new snipe will spend 0.00025 BNB on the new token)

transactionRevertTimeSeconds: Time to spend before transaction reverts. Recommended to leave at default.

gasAmount: amount of max gas to use per transaction. Recommended to leave at default.

gasPrice: max price of gas to use per transaction. Recommended to leave at default.

bscScanAPIKey: Your API key from BscScan.

observeOnly: enabling this will disable buying of coins but allows you to observe how the bot audits tokens. Recommended to try this at the start to make sure the bot can scan for new tokens.

Mini audit

The bot has an optional mini audit feature which aims to filter some of the scam coins (eg. wrongly configured, honeypots). Obviously, this is not going to be as good as a proper audit (eg. CertiK) but at least the coins the bot will buy will be higher quality and if you enable the options, you should be able to sell the tokens later on (provided it hasn't been rugged).

The following json entries are for mini audit. Set all to false to disable mini audits, although beware you will probably be buying a lot of scam coins.

checkSourceCode: checks if source code is verified. This function is needed for all the other functions so if you disable this be sure to disable all the other audit options. **Recommended.**

checkValidPancakeV2: checks if the correct PancakeSwap v2 router address is used in the code. Be aware some contracts may externally set their router address so this function may reject a potentially good token. **Not recommended.**

checkMintFunction: checks if a mint function is present in the code. **Recommended.**

checkHoneypot: checks the code to see if it might be a honeypot (where you can buy tokens but cannot sell). **Recommended.**

checkPancakeV1Router: checks to see if the PancakeSwap v1 router address is used in the code. You will not be able to sell the tokens later on if PCS v1 router address is used. **Highly recommended.**

Note: **be very careful** when editing *config.json* and make sure to not alter the syntax. For mini audit options, either use **"True"** or **"False"** making sure to capitalise the 1st letter. Any other spelling will not work.

Things to note

- Do not worry if you are not seeing any new tokens being detected. There are often around 10-20 new tokens being created per minute but that can vary quite a lot. Sometimes no new tokens may be detected for a few minutes.
- The bot only buys tokens whose liquidity is paired with Wrapped BNB (WBNB). You could alter the code to buy tokens paired with another currency if you wanted but it's not recommended.
- Please check that you have enough BNB in your wallet to afford sniping new tokens. If you don't the bot will not work.
- Please be careful when editing the config.json file. If you delete a comma or quotation mark etc. the bot will not work and throw an error.
- To launch the bot, run the 'launchBSCTokenSniper.bat'. The bot should then open in a cmd window and load.
- Don't left click in the cmd window as it will enable select mode and stop the output (you will see 'Select' in the title). If this happens right click your mouse to deselect it.

FAQs

I've sniped loads of coins - but how can I check which ones have made a profit?

- For this go to poocoin.app, click 'Wallet' and connect your Web3 wallet that you are using for your bot (eg. Metamask).
- It will then give you the list of tokens in your wallet and show you which ones have made the highest profit.
- Click the descending arrow next to the balance tab to show highest to lowest tokens value.
- Then you can manually sell the tokens which have made you a profit on PancakeSwap.

I keep getting 'Transaction failed' – what's going on?

Either:

- Gas amount / price too low
- Wallet address / private key incorrect
- Not enough BNB to pay for the token and TX fees

The bot isn't sniping that fast.

This is mainly due to internet speed and computer processing power.

Risks

Investing in BSC tokens / shitcoins is risky and be aware you could lose all your money. For this reason, **do not invest more money than you are prepared to lose.**

It is pretty much impossible to snipe bots very early and be sure it isn't a rug pull. When people create tokens in most situations, they will manually create liquidity in PancakeSwap. This is when the bot will detect the token. If they burn / lock liquidity, they will then usually send their LP tokens manually to a deadcoin address or put them in a liquidity locker. Therefore, you can't immediately snipe the tokens with 100% certainty they aren't rugpulls.

The mini audit feature can't be 100% accurate but aims to filter out the majority of scams / hacks and reduce the chance of losing your money.

If a programmer creates token code in a unique way, they may be able to bypass detection although this is generally quite rare, as the majority of tokens are forks of big projects with very little of the code having been changed e.g., Safemoon.

Legal note: BSCTokenSniper is copyright and is available for private use only. You must not share the source code files online to anyone other than yourself / reproduce and publish any part of the program or legal action may be taken. A lot of time and hard work was put in this bot so please be think about the developer.

