# BlockchainLaunchSniper Installation and Setup Guide

## About the bot

BlockchainLaunchSniper is a bot which can automatically buy / mempool snipe a newly listed token and sell when profitable on over 220 EVM-compatible blockchains and any Uniswap-forked DEX.

The bot allows you to snipe new token launches paired with any token (eg. ETH, BNB, USDT). It has the ability to mempool snipe and will instantly buy when liquidity is added. It constantly updates the price and shows you the profit you've made, has the ability to autosell at a specified profit margin (eg, 2x, 10x etc) and also allows you to sell manually with your keyboard. You can use the spacebar to sell 100% of your tokens or use keys 1-9 to sell 10-90% of your tokens.

BlockchainLaunchSniper features:

- Cross-platform (Windows, Mac, Linux etc).
- Supports all EVM-compatible blockchains.
- Supports mempool sniping (wait for liquidity to be added), buy as soon as token address is entered and other options.
- Auto sell (either manual with hotkeys or automatically with autosell multiplier eg. 2x profit).
- Stop loss (sell when initial investment is below certain value eg. 0.5x).
- Honeypot repeat checker (repeatedly checks the token until it is available for trading / isn't a scam).
- Snipe with any liquidity pair.
- Partial sell (eg. sell 10-90% of tokens with hotkeys then sell the rest later when you want)
- Configurable anti-bot delay (eg. Wait X amount of seconds before sniping).
- Easy to use config file to change a wide variety of settings eg. wallet address / private key, gas price / limit, node URL, liquidity pair address etc.
- Price updates multiple times a second showing profit multiplier (eg. 1.5x, 2x), current value of tokens and profit percentage.
- ‘no win no fee’ - only pay 10% fees on profit if you make a profit (less if you have a silver tier or higher - diamond tier has no fees).
- And much more.

By avoiding web interfaces and using nodes directly you can snipe tokens much 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 (although this depends on your setup and configuration). 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 / sell tokens.

The bot's source code is heavily compiled, packed and obfuscated to prevent people stealing code or harm the security of this bot. If you have concerns about the security of this bot then you should create a new wallet with a small amount of crypto etc and use that wallet's details in the config file. If you make profit then that can be transferred to your main wallet. You can also use a virtual machine to run the bot.

Another reason the is closed source is that it prevents people from ripping off the code and trying to make / promote their own bot which would cause the developers to lose out on income as funding is important to keep development going and pay expenses etc.



# Getting started

To get the bot up and running follow these simple steps.

## Prerequisites

- Windows, Mac or Linux computer (windows preferred)
- A reasonably fast internet connection
- blockchain wallet address and private key (not seed phrase) - recommended to create a fresh wallet on Metamask
- A Blockchain node (either use a free node, or if you want to mempool snipe then a private node)
- Enough crypto in your wallet to snipe tokens (and other paired tokens if you choose a different liquidity pair address eg. BUSD).
- Your wallet address must be registered (if not already registered, buy a premium tier from blockchaintokensniper.com/buy)

# How to install / run

The BlockchainLaunchSniper executable can be supplied with an argument for the config file you wish to use eg. BlockchainLaunchSniper ethereum_config.json (all config files must be in .json format).

***Windows:***

- *Edit the config.json file with your parameters.*
- *(Optional) - if you plan to use mempool sniping, use the included latency tester tool: double-click the .bat file and the latency tester will measure the average latency of your node (read the README file).*
- *Double-click the .exe file to run (please note that some antivirus systems may detect it as a false positive, this is due to the way it is compiled but the bot is not a threat and is completely safe).*

***Mac OS**:*

- *Edit the config.json file with your parameters.*
- *(Optional) - if you plan to use mempool sniping, use the included latency tester tool: double-click the .bat file and the latency tester will measure the average latency of your node.*
- *Navigate to the executable directory in terminal and run ./BlockchainLaunchSniper config.json (replace config.json with the filename of your json config file if needed)*
- *If you have permission issues, go to Security and Privacy in your Mac settings, and allow BlockchainLaunchSniper to run.*

***Linux:***

- *Edit the config.json file with your parameters.*
- *(Optional) - if you plan to use mempool sniping, use the included latency tester tool: double-click the .bat file and the latency tester will measure the average latency of your node.*
- *Navigate to the executable directory in terminal and run ./BlockchainLaunchSniper config.json (replace config.json with the filename of your json config file if needed)*

# Config

**Note:**

*Integer:* A whole number (eg. 0, 10)
*Float:* Any number that can include a decimal (eg. 1, 0.1, 105.5)
*Boolean:* True or False (make sure it is spelt 'True' or 'False' - other spellings will not work)
*String:* A piece of text (strings used in config should not have any quotation marks ' or " in them)

* *Don't edit this value unless you are sure you know what you are doing.*

** *Optional.*

## Launch sniper config settings:

"walletAddress" - (string): Your wallet address that you registered online (must start with '0x').

"walletPrivateKey" - (string): Your wallet private key (***not*** seed phrase).

———————

***(Optional) Sniping multiple tokens at the same time, or want to use another wallet to snipe?***

The bot has a feature where you can deploy multiple instances of the bot with different wallets at the same time, so you can snipe multiple tokens at the same time. You must still provide your wallet address and private key that you registered with the bot, however if you enable these settings then the bot will use the wallet specified below to snipe. ***If you only snipe with your registered wallet then ignore this section.***

** "activeWalletAddress" - (string): Your wallet address (must start with '0x').

** "activeWalletPrivateKey" - (string): Your wallet private key (***not*** seed phrase).

Example configuration (config.json):

```
{
    "walletAddress": "0x46202b57a5f6a0FF4cE317c976F453185552e2bC",
    "walletPrivateKey": "7495d0cf8087e7571288a104086dda2c2200aa4b14ca38742d07f4905f2d7321",
    "activeWalletAddress": "0x411eC2BcFbec8fb81c9d6aAFE67736832f3d56aD",
    "activeWalletPrivateKey": "2c0b3d2a47ab18a78bd78aa4112a3faaa6878a5e608dde5f1c0e3b71ab8706f1",
    (Other settings).....
}
```

Where *'walletAddress'* would be the wallet you used to register the bot, and *'activeWalletAddress'* would be the wallet that you will actually use for sniping.

———————

“blockchainNode” - (string): The blockchain node URL you wish to use. Must start with ‘*ws*’, ‘*wss*’, ‘*http*’ or ‘*https*’ prefix, however secure websocket (‘*wss://*’) nodes are recommended.

## “exchangeSettings”

This section allows you to change settings for the DEX. You shouldn’t need to edit these settings unless you want to use a new DEX that hasn’t been listed.

* “routerAddress” - (string): the router address for the DEX you wish to use.

* “routerABI” - (string): the filename of the ABI for your chosen DEX, eg. ‘PancakeSwap_RouterABI.json’.

* “buyWithCoinFunction” - (string): name of function to buy with blockchain native coin, eg. ‘swapExactETHForTokens’

* “buyWithAltPairFunction” - (string): name of function to buy tokens with alt pair, eg. ‘swapExactTokensForTokens’

* “sellWithCoinFunction” - (string): name of function to sell tokens to blockchain native coin, eg. ‘swapExactTokensForETH’

* “sellWithAltPairFunction” - (string): name of function to sell tokens to alt pair, eg. ‘swapExactTokensForTokens’

* “honeypotCheckerAddress” - (string): contract address for honeypot checker deployed on blockchain. Leave as blank if you’re not using it.

## “gasSettings”

“buy_gasAmount" - (integer): Gas limit when buying token.

“buy_gasPrice" - (integer): Gas price when buying token.

“approve_gasPrice" - (integer): Gas price for approving - note that there is no standard gas limit for approval TX’s but it is quite low (usually costs about 10-15 cents).

“sell_gasAmount" - (integer): Gas limit when selling token.

“sell_gasPrice" - (integer): Gas price when selling token.

## “tradingSettings”

“liquidityPairAddress" - (string): The liquidity pair address of the token you wish to use (set to blockchain native coin by default eg. WBNB, WETH, but can be changed to BUSD etc).

“sellTokens" - (boolean): Sell your tokens through the bot? (highly recommended).

“stopLossMultiplier" - (float): Stop-loss multiplier to be used, eg. set to 0.5 to sell when investment goes below 0.5x its initial value (set to 0 to disable).

“transactionRevertTime” - (integer): Time in seconds that a transaction will be processing before it will revert. For big launches it’s recommended to set quite low eg. 20 seconds so if the transaction doesn’t get processed fast enough it will revert.

## “trailingStopLoss”

———

***What is trailing stop loss?***

*"A trailing stop is an order type designed to lock in profits / limit losses as a trade moves favourably."*

For example, you could buy a token and sell at 2x the original value.

However, the token may increase in value even more after you sell it, so you could have missed out on alot more profit. If you sold a token at 2x its original value and it reached 10x, you would probably be annoyed you sold too early. So how do we make sure we squeeze the most amount of profit out of tokens?

Trailing stop loss is the solution to this. But first, let's explain what stop loss is.

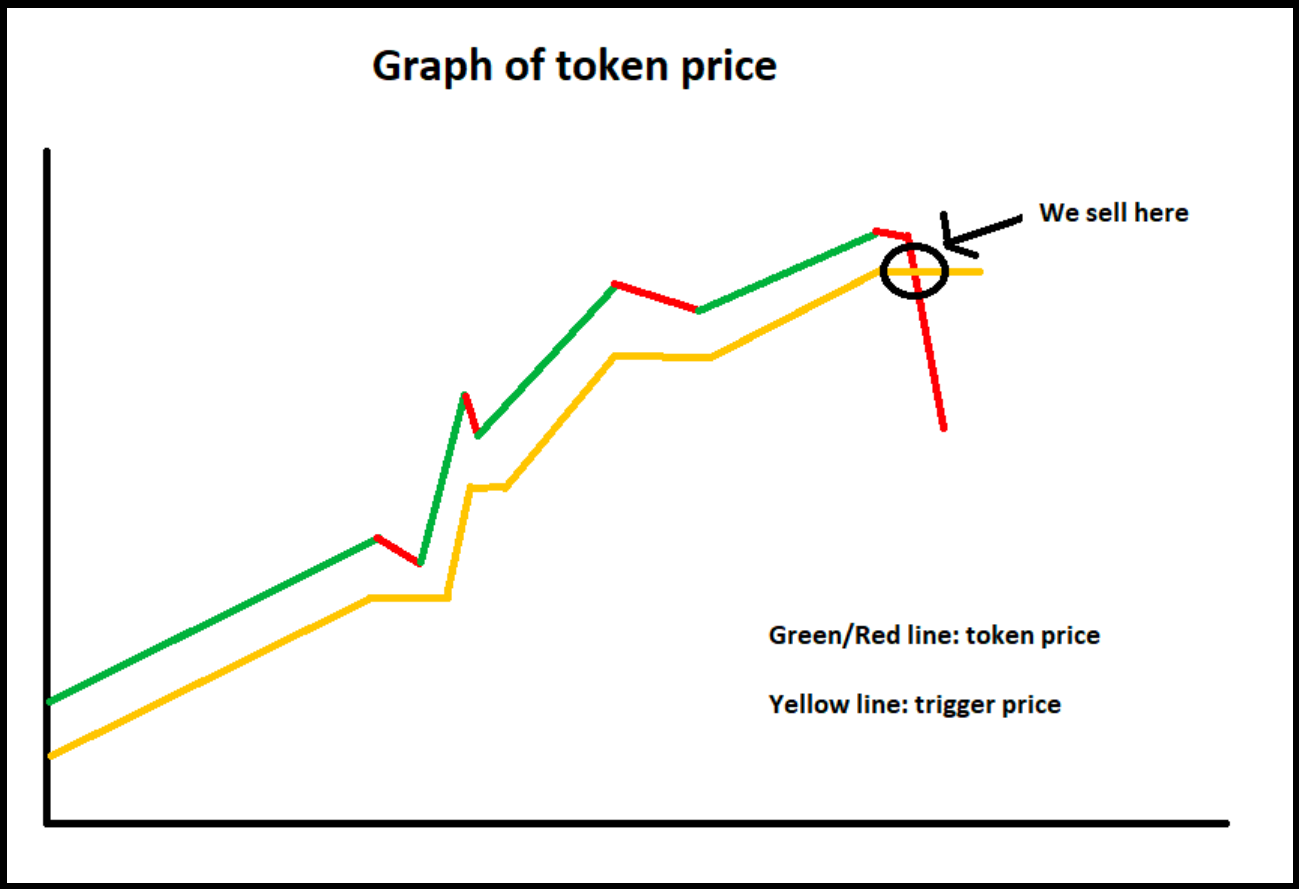
A stop loss is an order to sell your tokens when the value of your tokens goes below a certain value (trigger price) eg. 0.5x.

For example, we buy a token for 1 ETH, but the price keeps going down. Instead of watching the token price tank, we can sell when the price goes to 0.5x (which would be 0.5 ETH).

We will have lost money, but it's better selling our tokens for 0.5 ETH instead of 0 ETH.

Trailing stop loss is a stop loss, but the trigger price rises as the token price rises. If the token price pumps and then dumps, we can sell at the start of the dump and still make nice profits and sell before the token dumps.

In a trailing stop loss, the trigger price will never decrease. To make it more clear, here's a diagram showing how trailing stop loss works:



As you can see, the bot will sell (stop-out being triggered) as soon as the token price is the same as (or lower than) the trigger price.

You'll notice there's a vertical gap between the token price and the trigger price. This is called the trail amount (trailMultiplierAmount in config).

The trail amount basically means how much leeway we give the token price before we sell.

If we set the trail amount to 0, as soon as the token price dips even slightly, the bot will instantly sell. However, tokens sometimes dip slightly before the price goes up (look at above graph for example), and the bot would sell at the 1st dip.

Increasing the trail amount is often better as the bot will resist the small dips which will make you more profit in the end. Be careful not to set it too big or you may end up losing profit.

For example, imagine we have a token and we set trailMultiplierAmount to 0.5x. The token price rises up to 10x, but then the price tanks. The trigger price will stay at 9.5x (0.5x behind the current price multiplier), so the bot will end up selling at 9.5x.

———————

"enableTrailingStopLoss" - (boolean): Enable trailing stop loss? If not, you can still use the basic auto-sell feature (eg. Auto-sell at 2x).

"minTrailingStopLossMultiplier" - (float): Minimum price multiplier before stop-loss will work. Set to 1 by default.

"trailMultiplierAmount" - (float): trail amount measured in price multiplier (eg. 0.5 for 0.5x).

"trailingStopLossHardcap" - (float): Price multiplier hardcap to instantly sell at without further monitoring price while trailing stop loss is enabled. Set to '-1' to disable.

*NOTE: partial selling won't work when trailing stop loss is enabled, you can only manually sell 100% of your tokens.*

### "honeypotCheckerSettings"

"maxBuyFee" - (integer): Max acceptable buy fee before bot will buy.

"maxSellFee" - (integer): Max acceptable sell fee before bot will buy.

"honeypotCheckerAddress" - (string): Contract address of honeypot checker.

To get honeypot checker addresses, type /dexinfo into either the main group or testing group on Telegram. This will list all DEXs that currently have a honeypot checker address.

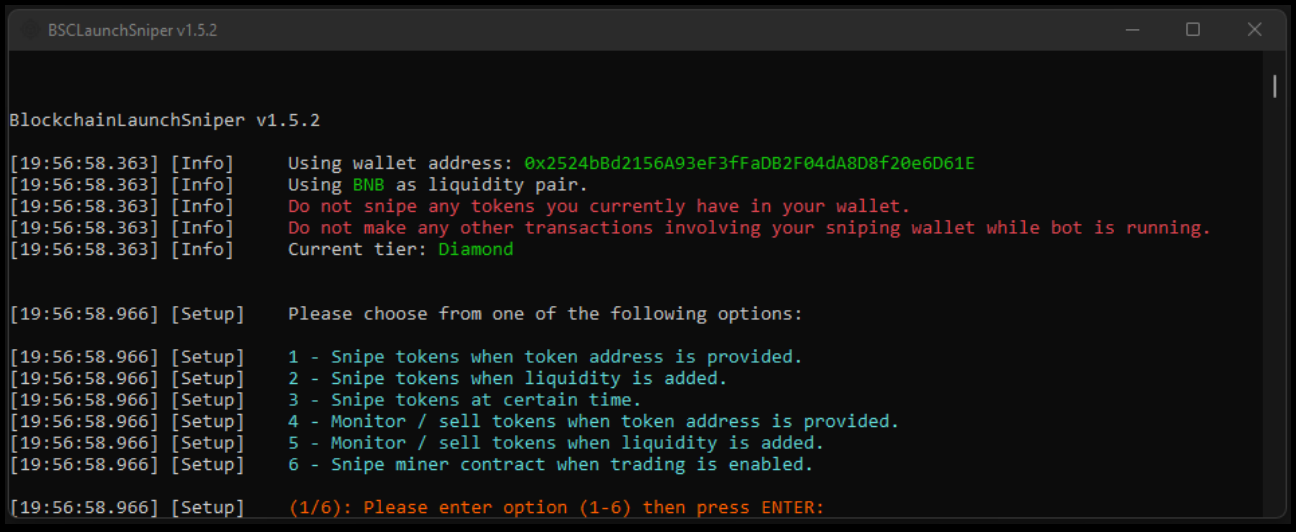
### "miscellaneous"

"enableConsoleColours" - (boolean): Display colours in your console? Some older terminals may not support console colours so you can disable if needed.

"enableHotkeys" - (boolean): Enable hotkeys for selling tokens? (recommended).

# Bot Interface

After the launch sniper has loaded and your config is correct, you will see a screen similar to the following menu:




As you can see, it gives you 5 options.

You can either:

- **(1) Snipe tokens when token address is provided.**
*Enter the contract address of the token you would like to snipe and the bot will buy it as fast as possible. This is useful when you do not know the token address before the launch starts and it is posted in eg. a TG channel.*

*If you know the token address before the launch starts, you can also enable the honeypot checker which will repeatedly check multiple times a second if the token is tradable. As soon as the token is tradable the bot will buy the token, if the token remains a honeypot it will never buy and you can exit the bot.*

- **(2) Snipe tokens when liquidity is added.**

*This allows you to mempool snipe, the bot will listen to every transaction associated with the token as soon as the TX's are pending, and will auto buy as soon as an 'addLiquidity' TX is detected. The bot uses function hashes and checks against a library to detect whether the function hash relates to adding liquidity / enabling trading.*

- **(3) Snipe tokens at certain time.**

*This allows you to snipe a token at a specific time.*

*Enter the token address and select a time that you would like the bot to snipe in HH:MM:SS format (eg. 12:00:00, 01:30:00, 16:20:03 etc).*

*If you would like to use anti-bot delay, make sure to factor that into the time (eg. If you would like use 3 seconds of anti-bot delay and the token launches at 12PM, you can put '12:00:03' instead). Beware that some tokens may not launch exactly at the time the developers claim (eg. It may take another few seconds until trading is enabled) so be careful. Also note that the timezone is in 24H format and is the same as your computer, so you may need to adjust the time if the token launch time is in a different timezone.*

*Beware that if you set it to snipe too early the TX may fail if there is no liquidity or the token's anti-bot system is active.*

- **(4) Monitor / sell tokens when token address is provided.**

*Allows you to sell tokens that you have bought without having to wait for liquidity to be added (it will instantly start monitoring and will sell when ready). For this, you are required to enter the amount you have spent buying tokens (eg. if you have bought 0.1 BNB worth of tokens in the*

*presale, you should enter 0.1). Make sure the amount you enter is accurate or the bot won't calculate the correct profit and you may lose money.*

- **(5) Monitor / sell tokens as soon as liquidity is added.**

*Similar to option 4, but this will wait until liquidity is added first.*

**- (6) Snipe miner contract when trading is enabled.**

*This option allows you to snipe new miner contracts which are similar to / forked from Baked Beans (bakedbeans.io). When you have found a miner you would like to snipe, first find the ABI (found on BscScan.com or similar blockchain explorer) and paste it into a .json file in the ABI folder of the bot. You must then find the function that the miner contract uses to allow users to invest (eg. buyEggs, hireMiners etc) - this can usually be found on the 'Write Contract' tab of the blockchain explorer. If you have a gold tier or higher, you can use a custom referral address. Finally, enter the contract address of the miner and wait for the bot to snipe the miner contract. The bot should work on most Baked Beans forked miners but some may not work. Feel free to ask in the Telegram group if you have any questions about this.*

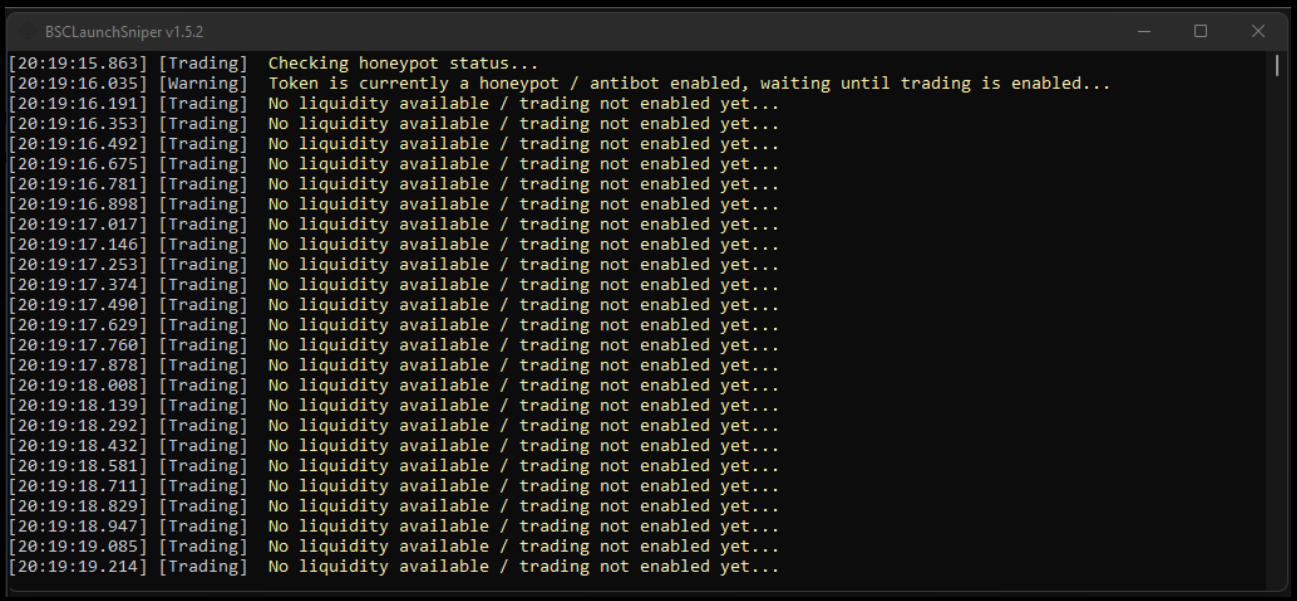
After choosing an option, you will have some further steps to complete before sniping can begin. (Ignore this section if using option 6).

- Choose the amount you want to spend on the snipe. The amount will be in the blockchain native coin eg. WETH, WBNB or whatever liquidity pair token you've chosen eg. BUSD, Cake etc.

- Enter the autosell multiplier you wish the bot to sell at eg. 1.5 for 1.5x, 10 for 10x etc. To only sell manually, press the ENTER key.

- Enable honeypot / antibot check - this is a feature of the bot that will repeatedly check if the token is tradable until it can be successfully traded. If the token turns out to be a scam it will never buy the token and you will have to exit the bot. During testing with a speedy node, it achieved approx. 5 checks per second. A status of 'True' means the bot is either not available to trade or that time or it is a honeypot. Do not be alarmed if a token has a honeypot status of 'True' as many tokens nowadays have anti-bot systems to stop bots.

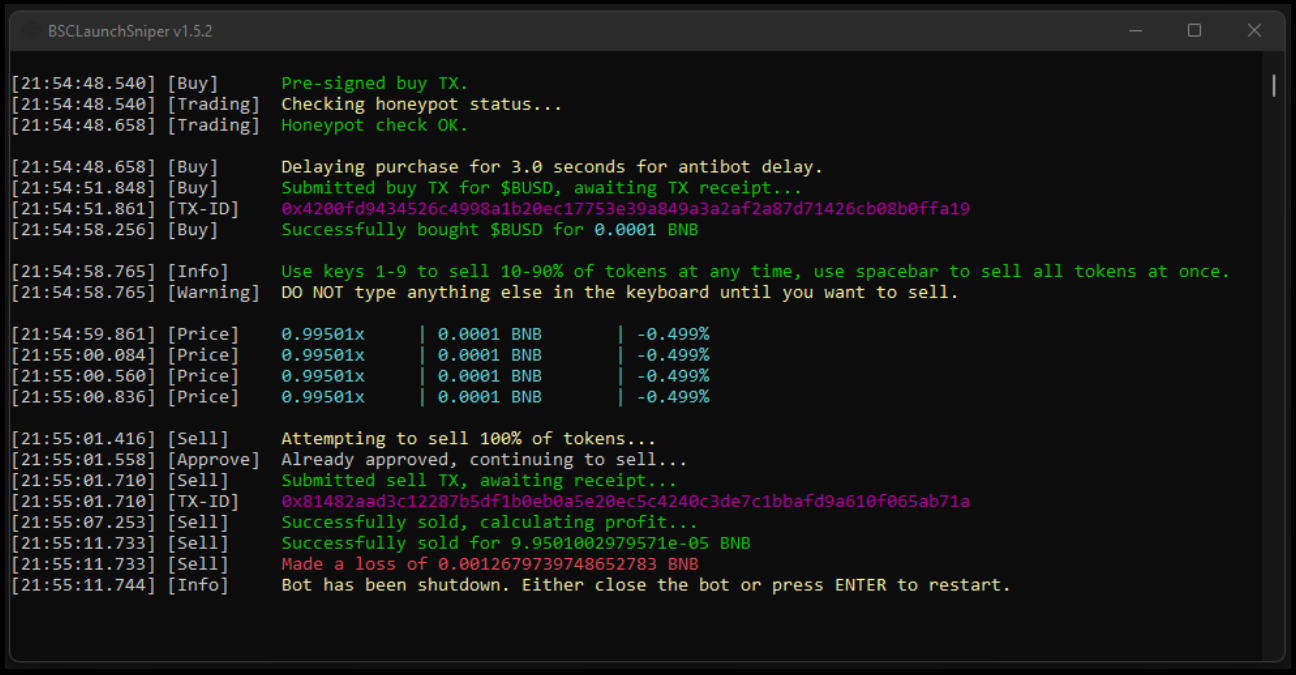
Example of honeypot checker running on a scam token (honeypot):



- Enter the anti-bot delay in seconds - alot of tokens usually have an anti-bot delay at launch to prevent bots from buying all the tokens. You can bypass this by entering the exact anti-bot delay into the bot. You can usually determine this by looking at the smart contract of the token or asking the developers of the token.

- Enter the contract address of the token and press enter.

A typical output from the bot will look like this (this was a test snipe using BUSD so no profit was made):




If the anti-bot delay is set then it will wait that number of seconds before buying.

It will then directly buy through the exchange's router smart contract, bypassing all web3 interfaces which will be much faster than manually buying could ever be.

It will give you the TX ID for all transactions for later analysis.

The price multiplier, total value of tokens and percentage gain in profit is displayed.

If enableHotkeys is enabled in the config file, you are able to sell your tokens via your computer's keyboard. The 1-9 keys and the space key are used for manual selling. Note that if an autosell multiplier has been specified it will still sell when it hits that target as well.

*Please be aware that the bot will still detect keystrokes even if the bot window isn't in focus / in background, so do not do anything else with your computer while the bot is running or you may sell by mistake.*

- The 1-9 keys will sell 10-90% respectively of your tokens eg. the 1 key will sell 10%, the 5 key will sell 50%, etc.
- The spacebar will sell all (100%) of your tokens.

Otherwise, the bot will only sell when the autosell multiplier has been reached.
As soon as the bot tries to sell, it will submit an approve TX to your chosen DEX. As soon as it is confirmed, it will submit a sell TX to your chosen DEX for the amount of tokens needed. If you

have partially sold tokens (eg. 10-90% of tokens) then the bot will still continue monitoring price and will sell when needed. If 100% of tokens have been sold then the bot will end.

The bot will automatically calculate the profit/loss that has been made on that TX. If the transaction is profitable then it will send 10% of your profits to developers as a fee (or less if you have a silver, gold or diamond tier). If the transaction isn't profitable or you break even, then no fees are taken.

# Troubleshooting

**"Failed to connect to node"** - could be a variety of reasons, such as the node not running / broken, your internet connection, firewall etc. Try a different computer / node.

**Bot doesn't detect liquidity (stuck on 'waiting for liquidity'):**

- check that the token address is correct.
- check that the node is working and fast enough (use the supplied node latency tool to test).
- check if the token used an addLiquidityETH transaction to add liquidity - other function hashes may not be detected by the bot like the finalize() function for example.

**Transaction fails:**
- Use dashboard.tenderly.co and paste in the TX hash of the failed transaction.
- You should get an error code in the top left of the screen (red box).
- "BP: address is not whitelisted": the token you tried to snipe was a whitelist token, you didn't whitelist your wallet.
- "BEP20: transfer amount exceeds allowance": check you approved the liquidity paired token before using the bot.
- Other errors could be due to a badly configured token, anti-bot system, broken contract etc.

**Incorrectly configured config.json file:**

- Use a json formatter / validator (https://jsonformatter.curiousconcept.com is recommended).
- Paste the contents of your config.json file into the validator and click on Process.
- The validator should show you where there is an issue. Make sure that you have correctly used the following symbols: " ' } { [ ]
- Make sure when you save the config.json file that you use the correct encoding (ANSI).

# Strategies

The launch sniper does not guarantee that you will make a profit. However, if you use the right strategy you are very likely to make good profits, and you have a huge advantage over manual apes and are very likely to make profit.

In general, to make the most profit and snipe the fastest you will need:

- A fast computer / VPS.
- A fast internet connection.
- A private node (nodes have to be very fast to reliably mempool snipe tokens) - recommended to build your own node such as a geth node, or use a Hetzner node for example (AX61-Nvme will work fine although some cheaper ones may work just as well) - see https://docs.binance.org/smart-chain/developer/fullnode.html for more details.
- A fairly large amount of crypto to invest in a snipe. You can invest any amount you like, but in general, investing eg. 0.1 BNB or more will yield the most profits.
- High gas fees (specifically a high gas price for buying).

Blockchain validators usually prioritise TX's that pay higher gas, so you will effectively 'jump the queue' when sniping (buying) and get in first so you can purchase for the lowest price, which will give you the best advantage and make you the most profit. Gas fees will be higher but if investing a larger amount of money it it significantly worth it.

In general, a gas price of at least 10 gwei is recommended for buying. However you can go even higher to get in faster, and some bots pay 100 or even 1000+ gwei gas when sniping.
The minimum gas price that will be accepted is 5 gwei. However, it is not recommended to use this low a gas price as you will be at the 'bottom of the pile' and your transaction may end up being processed later than desired, which could cause you to miss out on alot of profit or lose money. Gas fees for selling aren't so important but more than 5 gwei gas is recommended.

Trailing stop loss is recommended when selling to make the most profit.

# How to spot / avoid scams

Unfortunately, crypto is full of scams and there are scammers everywhere trying to steal your money.

Scams happen all the time in crypto and are constantly evolving but with some basic knowledge you should be able to avoid the majority of them.

The most common form of scam is a 'rugpull'. This happens when people buy a token but the liquidity is unlocked. This allows the scammer to easily withdraw the liquidity which is where all your funds go. Some scammers make as much as $100k+ in a few minutes with this type of scam so unfortunately it's not going to stop anytime soon.

Another type of scam, often integrated with the rug pull, is a honeypot. This happens when the token developers alter the code of the smart contract so tokens can be bought, but not sold, so effectively your money is stolen as you cannot withdraw it. Most honeypots have trading disabled at the start so as soon as they launch they can be detected, but some honeypots are 'dynamic' which means that the token developers can disable trading at any point in time. The only way to check this is by analysing the smart contract.

**What should I do before investing in a fair launch?**

Most likely the fair launch will have an associated TG channel. Check the pinned message of the channel to see if you can find the contract address.

If you can find the contract address of the token, first go to a blockchain explorer and enter the contract address. Find the 'contract' section of the token and see if it's verified.

If it's verified, then copy the code and go to **rugpulldetector.com** and paste the code there (a good tool for checking for common scams). If it shows any red flags then it's recommended to avoid the token and don't invest. You may want to look through the code yourself to check if there are any dodgy / scam functions eg. selfdestruct()

If no scams are detected you should be good to invest, but be wary that many fair launches do not lock liquidity which means you may be vulnerable to a rugpull, which is why in general you should sell quickly after buying / avoid tokens with unlocked liquidity.

If you can't find the contract address or it isn't available, you will have to wait until launch and quickly use option 1 when sniping.

You can also search blockchain explorers eg. BscScan for contracts with the same name as the token, but sometimes this may not work.

When you buy tokens and have made a profit it is generally advised to sell quickly. These new tokens often end up to be ‘pump and dumps’ which means that the price will rapidly rise at launch and then dump right back to where it started.

Also be aware that new tokens that lock liquidity sometimes lock them for short periods of time (eg. A few days or hours), after which time they could easily rugpull.

The following tools are recommended for detecting and avoiding scams:

- rugpulldetector.com - paste token contract code, will check for any bad functions / scams
- honeypot.is - checks if token is a honeypot *at that moment in time*
- staysafu.org/scanner - a good scanner that checks for many issues
- bscheck.eu - another good scanner that checks for many issues

# I’m still stuck with something.

Please look through the BlockchainTokenSniper Telegram group as your question is likely to have been answered already. However if you still can’t find the answer then feel free to ask in the TG group and someone will likely answer your question quickly.

**Got any other issues / suggestions / improvements?**

Please feel free to mention issues in the group or contact the main dev @BytePhoenix on Telegram.

Telegram group: t.me/blockchaintokensniper

***Happy sniping!***

***BlockchainTokenSniper team.***