

"If You Ain't First, You're Last" Reverse-Engineering a Leading Pump.fun Sniper Bot – Part 1





Nested

Jul 20, 2024 — 5 min read

Subscribe

king of the hill



Created by  HGnq
 market cap: 65.21K [badge: 🐾]
 replies: 5
testdontbuy [ticker: nobuy]


Following Terminal For You


sort: bump order ▾


order: desc ▾


Show animations: On Off


Include nsfw: On Off





Created by  4vJgGy
 market cap: 14.60K
 replies: 38
cat wif drip (ticker: drip): damn this cat got drip no cap fr fr





Created by  CKDqZ3
 market cap: 25.65K
 replies: 1
Class Cat (ticker: CLASS): skool is vry hard




Created by  BtH278
 market cap: 10.10K
 replies: 20
TRENCHES WARRIOR (ticker: TRENCHES): WE LOVE D A TRENCHES





Created by  F5m5HT
 market cap: 29.12K [badge: 🐾]
 replies: 11
Goba (ticker: Goba): i need a treat please





Currently live streaming!


Created by  2dec3C
 market cap: 46.68K [badge: 🐾]
 replies: 212
Cathie Wood on Sol (ticker: WOULD): This coin is dedicated to the innovative Founder of Ark Invest, Cathie Duddy Wood. 🐾 would you? 🐾





Created by  FbCXPx
 market cap: 14.22K
 replies: 76
dealer (ticker: dealer): Feeling overwhelmed by trading? Take a break and visit \$DEALER for ultimate relaxation.




Created by  9bWN89
 market cap: 24.52K [badge: 🐾]
 replies: 55
Charlie (ticker: Charlie): He's such a good boy



Created by  G8Jm6z
 market cap: 18.57K
 replies: 90
FOLDER (ticker: FOLDER): Just a simple windows folder . If you never opened one don't buy. Community project.



Created by  Ealpgt
 market cap: 19.88K
 replies: 14
Mister Miggles (ticker: MIGGLES): The new generation of degens. Mr Miggles will conquer the meme coin world and overtake all other Miggles on other chains.

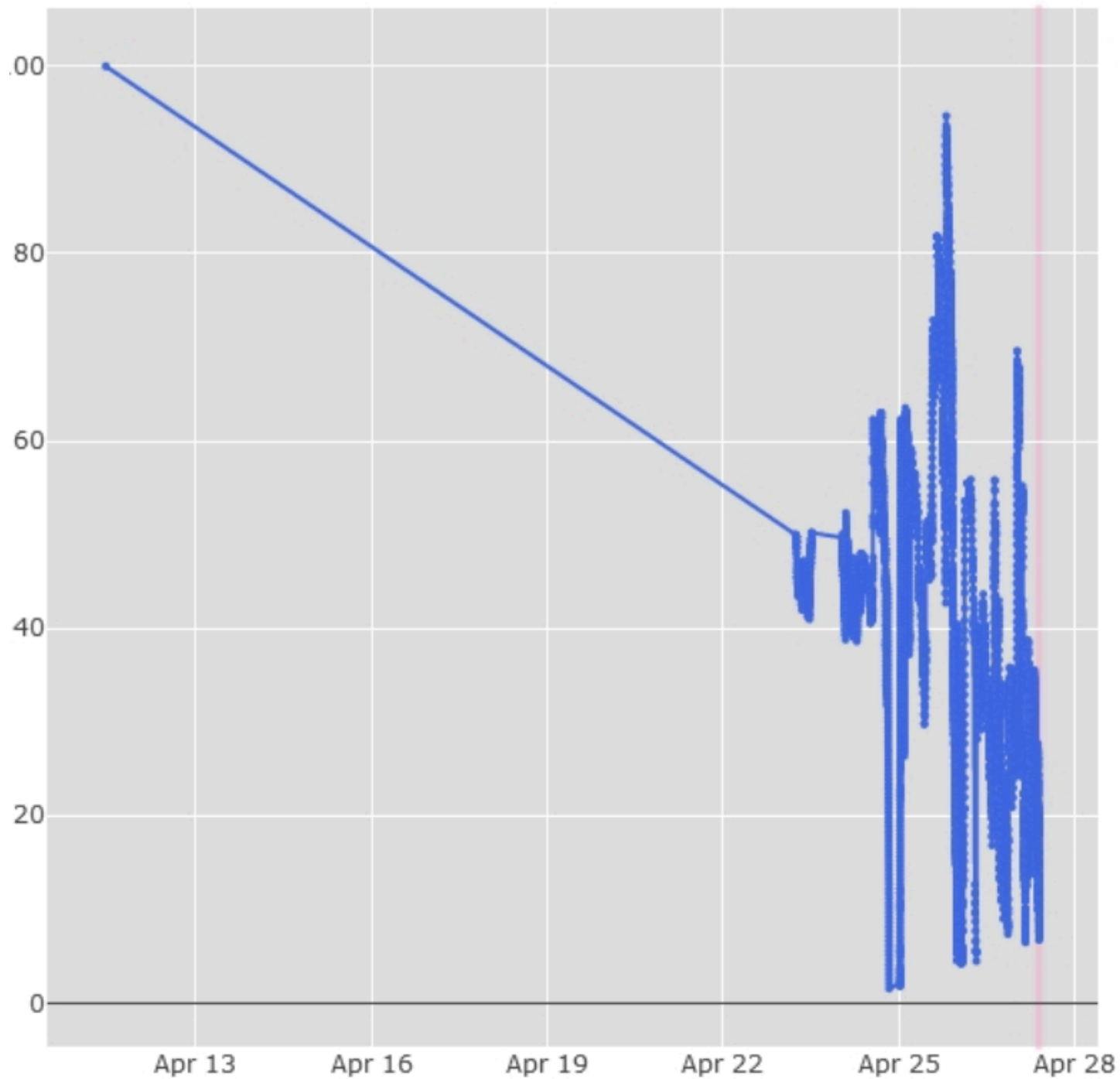
In early May, I came across the site [Pump.fun](https://pump.fun), where you can trade user-created coins and launch your own for just a few dollars.

Like many others, I was quickly enamored with the idea of buying a coin early and netting 100x returns on it. However, it was never meant to be, and I never bought one of those

elusive coins – but I did find something that set off a month of hard work with [@licketyspliket](#)

Who is orcACR...

After spending hours on the site, I came across a peculiar trader, [orcACR...](#), who kept appearing as the first buyer (excluding the creator) for freshly launched coins. At all hours of the day, this user would purchase around 20% of new coins created. I became curious about his activity and attempted to check his balance history, which initially led me to believe his bot was unprofitable.



Stake.tax chart for orcACR's Solana balance

That didn't make sense to me, though. Why would they continue running the bot for weeks on end, slowly losing money? I downloaded a CSV of their trade activity to see what was really going on. Out of the thousands of swaps and token transfers, there were a handful of withdrawals to another address. The screenshot shows just how profitable they were.

The screenshot shows the Solscan website interface. At the top, there's a navigation bar with the Solscan logo, a price ticker for SOL (\$139.29, +14.06%), and various menu items like Analytics, Defi, NFTs, Leaderboard, Blockchain, Resources, and Sign in. Below the navigation bar, the 'Account' section is displayed for the address 'orcAbsjojvVkw9M5cQnRZ5UEASXjpQEi2Pp8cx7NJMb'. It includes a search bar, buttons for Buy, Exchange, Play, and Gaming, and a sponsored banner for 'Harambe Token Presale'. The account details are divided into three panels: Overview, More info, and Sponsored. The Overview panel shows the SOL Balance as 1,089.99 SOL (\$151,826.09). The More info panel shows the Owner as 'System Program', 'isOnCurve' as 'True', and 'Stake' as '0 SOL'. Below these panels, there are tabs for Transactions, SOL Transfers, SPL Transfers, Portfolio, Stake Accounts, and Domains. The 'SOL Transfers' tab is selected, showing a list of transactions. The table has columns for Signature, Block, Time, From, To, and Change Amount (SOL). It lists six transactions, all from 'orcACR...NXTCc8' to 'This account', with change amounts of +18, +10, +41, +11, +10, and +10 SOL respectively.

Signature	Block	Time	From	To	Change Amount (SOL)
3uWPVXPca4AAf6WKxZJaES...	#263473685	about 3 hours ago	orcACR...NXTCc8	This account	+18
rM2KaDvXgDKdxwidiDjhJEEuw...	#263418784	about 10 hours ago	orcACR...NXTCc8	This account	+10
3YTUJKH4pXjX5no9j1zSQd3T...	#263409640	about 11 hours ago	orcACR...NXTCc8	This account	+41
554rfJEgaHYMPasnGeNbVhP...	#263409356	about 11 hours ago	orcACR...NXTCc8	This account	+11
2nn96V55R6EdCpUadELb8aU...	#263407424	about 11 hours ago	orcACR...NXTCc8	This account	+10
2LuB7JMnDCfy2VRzjng1WD...	#263403744	about 11 hours ago	orcACR...NXTCc8	This account	+10

orcACR Withdrawal Wallet (May 02, 2024)

After reviewing the deposits, I determined they were making around 100-200 Solana daily, roughly \$10,000 to \$20,000 every 24 hours. At this point, it was clear they had a great

strategy and were, in all likelihood, one of the most profitable users on the site. I reached out to Lick to begin hatching our plan.

Note: Googling the address now, two months after I first discovered it, reveals many posts on Twitter and other platforms discussing orcACR's activity. It's interesting to see that this user is still actively operating their bot and has remained profitable after all this time.

Some background on Pump.fun and coin pricing

For those unfamiliar with Pump.fun, a creator, known as the "dev" of a coin, pays around 0.02 SOL to list their coin on the site. The creator also has the opportunity to be the first buyer, giving them the best possible price and the inability to take any loss on the coins they purchase before launch.

This is thanks to Pump.fun's use of a bonding curve to determine coin pricing. Quoting from a great [KuCoin article on Pump.fun](#), "As more people buy a coin, its price increases along a predefined curve." Below, you can see how this curve works from the creator's perspective:

account	type	SOL	WCULTURE	date ↻
 DM6Jdc	sell	1.3000	44.57m	31s ago
 9FSdvN	sell	0.2450	7.99m	10m ago
 9FSdvN	buy	0.2450	7.99m	11m ago
 GetGuD	sell	0.4711	15.25m	11m ago
 GetGuD	buy	0.4711	15.25m	11m ago
 DM6Jdc	buy	1.3000	44.57m	11m ago

With this pricing model, the earlier you purchase a coin, the less downside risk you face on a trade, depending on the number of coins already purchased. This principle also applies to potential upside, making it advantageous to buy coins as quickly as possible. This concept is fundamental to orcACR's strategy, but why do they attempt to buy only 20% of coins?

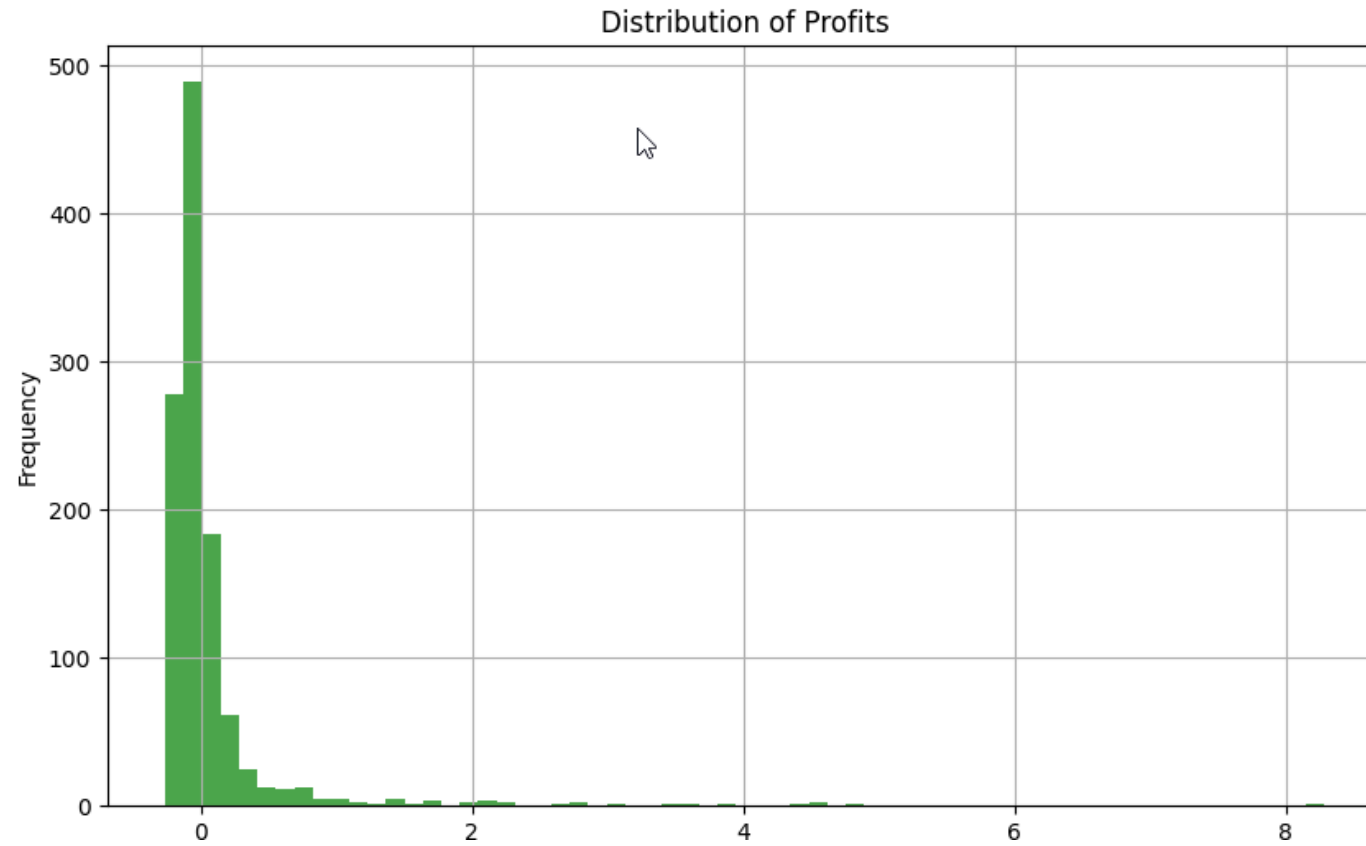
Reverse-engineering the bot

To understand orcACR's strategy, I needed access to their complete trading history. Fortunately, Pump.fun provides an easily accessible API to fetch new coins and their trading history.

After writing a few Python scripts, available in the [Part 1 Repository](#), and setting up a MySQL database, I began storing data on all coins and trades. I then conducted analyses to build a profile of the coins orcaCR has bought and to evaluate their trading profitability based on different metrics. Below, I've included screenshots of some of these analyses:

Note: The scripts were written quickly for the purpose of analysis and are not production-level code. They served their purpose during our race to reverse-engineer the strategy and do not follow best practices for code quality.





```
Total trades 1179
Total Profit: 86.65789999999996
Average Position Size: 1.4599208651399127
Hours for profit: 24.539166666666667
Highest creator initial buy where orc bought 4.0
Profit where creator purchased < 2 sol 77.596300000000001
Times purchased from previous creators 0
Times purchased with no creator buy / purchased before creator 87
```






output from `analyze-orc-json-trades.py`

Following the analysis, we were able to define some conditions where their bot **never** purchased a coin:

- If the creator has created another coin
- If the creator was funded by someone who had created a coin
- If the creator purchased over 4.0 SOL worth of the coin

Along with these rules, orcACR's bot would immediately sell all coins if the creator sold any of theirs. Another behavior we observed was what we termed "copy-block trades," where orcACR would enter a position, wait around 20 seconds, and then exit the position, regardless of whether the creator sold. This ruled out copy-trading as an effective strategy.

A final crucial detail we noticed was the speed at which orcACR operated. Within seconds, they would complete their analysis, decide to purchase a coin, and have their transaction confirmed. This indicated that orcACR's algorithm relied solely on data from the initial mint instruction on the blockchain, as any other method would be too slow.

 orcACR	sell	2.8737	59.15m	May 2 1:37:15 pm	2TxTu5
 rgamestop	buy	1.0000	23.10m	May 2 1:37:08 pm	5L1QsR
 4wv5zo	buy	2.8360	72.89m	May 2 1:37:06 pm	4aXtWZ
 orcACR	buy	1.9993	59.15m	May 2 1:37:05 pm	4CopAt
 EuMVAy	buy	2.0000	67.06m	May 2 1:37:03 pm	3GyPSE

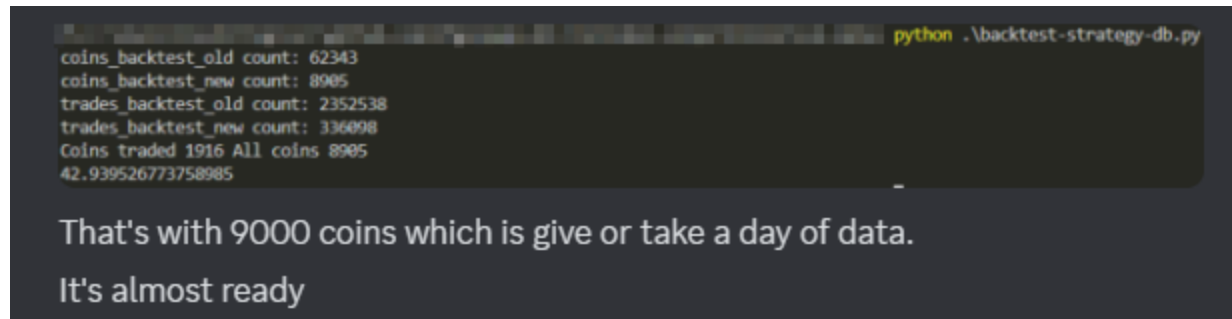
orcACR purchasing a coin just seconds after launch, followed by a copy-block trade to secure quick profit.

Backtesting the strategy

With the rules for the bot defined, it was time to test them. After creating backtesting tables in the database, we compared the coins we wanted to buy against those orcACR purchased and estimated our strategy's profitability.

Since we weren't trading live, we used a reasonably optimistic approach to determine profitability. We assumed we would exit the trade immediately after the creator sold and that we would be the second buyer, following the creator.

Additionally, we estimated the buy/sell price using a 14% deduction from the creator's buy/sell price. While not 100% accurate, this method aligned with the profit margins seen from orcACR when cross-checking results, confirming that we were on the right track.

A terminal window with a dark background. The top line shows a command prompt with a green 'python' prompt character followed by the command '.\backtest-strategy-db.py'. Below this, several lines of text display backtest statistics: 'coins_backtest_old count: 62343', 'coins_backtest_new count: 8905', 'trades_backtest_old count: 2352538', 'trades_backtest_new count: 336098', 'Coins traded 1916 All coins 8905', and a final line showing a profit/loss value '42.939526773758985'. Below the terminal output, there is a text overlay in a light gray font that reads: 'That's with 9000 coins which is give or take a day of data. It's almost ready'.

Final backtest results using `backtest-strategy-db.py`

What's next?

At the time, it felt like the most critical part of the project had been completed: reverse-engineering the strategy to a reasonable degree. If everything went smoothly developing the bot, we should be able to compete with orcACR. We narrowed in on many of the strategies and rules they used and shared a lot of overlap in coins purchased during backtesting.

In the next part of this blog, I'll delve into our implementation of the strategy, discuss the pitfalls we encountered and how we navigated them, and share some final thoughts on the project, as well as the entire codebase.

READ MORE

"If You Ain't First, You're Last" Reverse- Engineering a Leading Pump.fun Sniper Bot – Part 2

This blog post includes the
codebase that...

By Nested — Jul 29, 2024

Nested Tech Write-Ups

Powered by Ghost

Nested Tech Write-Ups

Articles on automation and web security.

jamie@example.com

Subscribe