

solXEN

solXEN 420 miner v.1.1

This guide will install rust, solana, create and fund wallets on solana testnet and then run 420miner to find 420 hashes.
This is early iteration, everything will change, big boy pants don't be nude

get a server mr hashhead, and enter matrix

Install rust

```
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh -s -- -y
```

Install solana (tools t run things)

```
sh -c "$(curl -sSfL https://release.solana.com/stable/install)"
```

Run export

```
export PATH="/home/ubuntu/.local/share/solana/install/active_release/bin:$PATH"
```

Check solana installed

```
solana --version
```

Create new wallet

```
solana-keygen new
```

Create pass-phrase or not is also fine

Output:

BIP39 Passphrase (empty for none):

```
Wrote new keypair to /home/ubuntu/.config/solana/id.json
```

```
=====
```

pubkey: WILL DISPLAY HERE

```
=====
```

Save this seed phrase and your BIP39 passphrase to recover your new keypair:
chat follow region rain tide sweet robot embrace cage shop message deposit

Connect to testnet

```
solana config set --url https://api.testnet.solana.com
```

```
solana airdrop 1  
#gives you 1 SOL
```

```
solana balance  
#should have 1 SOL on testnet
```

Setup another wallet (to run test with multiple wallets)

```
cp /home/ubuntu/.config/solana/id.json /home/ubuntu/.config/solana/id1.json  
#This is a private key we're copying and putting different file
```

```
solana-keygen new --force  
#creates another wallet
```

```
solana balance  
#Should be zero there
```

```
solana airdrop 1  
#give you 1 SOL in wallet2
```

```
solana balance  
#Should be 1 SOL there
```

Copy wallet to new location

```
cp /home/ubuntu/.config/solana/id.json /home/ubuntu/.config/solana/id2.json  
#now we have backups on 2 different files. Now we have 2 funded wallets
```

Create cargo project

```
cargo new my_project  
#Should work if doesn't path is not setup, then do this:
```

Setup path
source ~/.bashrc

cargo new my_project

cd my_project

nano Cargo.toml

Add these dependencies:

```
[dependencies]
solana-client = "1.10"
solana-sdk = "1.10"
```

#Copy save (ctrl X on mac)

Run program

#Copy this code

<https://gist.github.com/jacklevin74/b3b3709aa3e66eab8f762c0fb4de53ff>

Put in file under src main

nano src/main.rs

#Remove stuff there and paste code from link

#Ctrl X (save)

Build client into byte code that will execute from command line
cargo build

If error install cc compilers that rust is using
sudo apt install gcc

Retry cargo build

cargo build

#Should get no errors. If still errors youre ngmi jk

Go to this page:

<https://explorer.solana.com/address/7R2KMCUW1GimTEiS8tp8jJrde2N66yQiJ1MEUTbaPgfq?cluster=testnet>
#s.c. on their testnet

You havent run client yet, but will now

Run client (
cargo run

Go back to page and check your tx's

<https://explorer.solana.com/address/7R2KMCUW1GimTEiS8tp8jJrde2N66yQiJ1MEUTbaPgfq?cluster=testnet>
#Should get tx's
#Scheduler puts you in a 2 different threads since you had 2 accounts

Find your address
solana address

paste address to see your tx's

<https://explorer.solana.com/address/EFPqkTNmMsVFVCEzgrWrhmXQzq1LiRsbJ6JG7CSYPVMR?cluster=testnet>

click on the top tx for example

<https://explorer.solana.com/tx/28HmNepZCUxGbqjRvdib3ccJp4DK6zuevtL8YsRC75RS2AjRmJDue6awXiEkHCi1WrvmtFwaGD6V2xgTWVptt?cluster=testnet>

Program Instruction Logs

[Raw](#)

```

#1 Compute Budget Program Instruction ✕
> Program returned success

#2 System Program Instruction ✕
> Program returned success

#3 Unknown Program (7R2KMCUW1GimTEiS8tp8jJrde2N66yQiJ1MEUTbaPgfq) Instruction ✕
> Program logged: "Initiating solXEN Miner."
> Program logged: "Current slot: 267205880"
> Program logged: "Found '420' in hash at iteration 40 hex b1bb5ed676e41db2605dfc2261e39a43569dcae856bce053b933420af1b9aec2"
> Program logged: "Searching '420' in hash at iteration 71 hex cf516b25f2f6a6ce046cf13b33646da9de8bcd9b430866600573c89526986edf"
> Program logged: "Searching '420' in hash at iteration 72 hex 6ac317bfd6ebbee7ee99c08ed91bf10abb93530db63993dd0fad3b67d7d1f3e73"
> Program logged: "Searching '420' in hash at iteration 73 hex 89370694a389733daa7d093e96cc8841f6a59502b9a04bba5d7cd92548b00355"
> Program logged: "Searching '420' in hash at iteration 74 hex f3ecc10f1d21aeb7aa0c2951fe5cfac455a771d7d6108213c99178785a0f5dc"
> Program logged: "Searching '420' in hash at iteration 75 hex 9773c13ccaa3b3b6c730d9eee7b184e84967ef078f86631da646fe720be47e5c"
> Program logged: "Searching '420' in hash at iteration 76 hex a37d8c7e623ba017c824b45e225fcfa34e9b986740ae0102dfb2beb1122e51b29"
> Program logged: "Searching '420' in hash at iteration 77 hex 5a703480e23a74ecf567f8bf4d26585ca7326c2af93c0ac3e3873d4a92fde56"
> Program logged: "Searching '420' in hash at iteration 78 hex eala2bace5647d0317b576a6681d7901e927fddc77e07514449002b5c8f15a7"
> Program logged: "Searching '420' in hash at iteration 79 hex 88103c97e4533ff3a278ae5fa000c3c7c9ffdf4f3dd72284e11c90e87b46f3d8c"
> Program consumed: 1141840 of 1199700 compute units
> Program returned success

```

Found 1 hash at hash number 40

We're generating 10 hashes and were searching each one. When 40th was generated we found it – a 420 hash

Program Instruction Logs

[Raw](#)

```

#1 Compute Budget Program Instruction ✕
> Program returned success

#2 System Program Instruction ✕
> Program returned success

#3 Unknown Program (7R2KMCUW1GimTEiS8tp8jJrde2N66yQiJ1MEUTbaPgfq) Instruction ✕
> Program logged: "Initiating solXEN Miner."
> Program logged: "Current slot: 267205800"
> Program logged: "Found '420' in hash at iteration 19 hex 629e48aa899b424e45bbb0919420cd53b9679be7f6dfa4d0414de445531c0d9"
> Program logged: "Found '420' in hash at iteration 51 hex 042054d94c01ef3690b957b8427e989d75b0700bf697878d4fd3eb25e5c8bdb2"
> Program logged: "Searching '420' in hash at iteration 71 hex e3b5cb7297fbc2e284ba170ab363e43baedc464b4e361770078a55dec4aae846"
> Program logged: "Searching '420' in hash at iteration 72 hex a993ff dab8bdb7b0b54ec1f6b264c14bade25e093e19b9ec3265d15c8639b4b6"
> Program logged: "Searching '420' in hash at iteration 73 hex 81ade9ca1df24318f3bb14f9fc6ea2177210bf a543ca8aa8cd0153e051020b4a"
> Program logged: "Searching '420' in hash at iteration 74 hex 6d2590ee55d649368505014f2259d8fd9fa0add43c4e8a5fa11881db3e49696"
> Program logged: "Searching '420' in hash at iteration 75 hex 32e01859c17373bf6dbb468a78761267b93d56b0e8f332db3ea05c12e2fe5026"
> Program logged: "Searching '420' in hash at iteration 76 hex 3c38cb01d6894aa315e6e2ad97e6679bc720dd3876931b7639dc80ca2d85ce30"
> Program logged: "Searching '420' in hash at iteration 77 hex 8b036e1b569d4f7285f5ae996cbf4cec55a0d8b417de849dc5176f7a483187e0"
> Program logged: "Searching '420' in hash at iteration 78 hex 15c9e094fa6a6ea0ed87eb05783cdc266545795ef0ede67ed7b0c16719ef55e1"
> Program logged: "Searching '420' in hash at iteration 79 hex 419f18d5ddb8ca3f67ae4538e8efb4118e78bf271bc039d6a98f8941bdd487da"
> Program consumed: 1142536 of 1199700 compute units
> Program returned success

```

Here I found 2 420 hashes. How degen of me

DETAILS	
Transaction	
Overview	
Signature	28HmNepZCUxGbqjRvdrib3ccJp4DK6zuevtL8YsRC75R52AjdrmJDue6awXiEkHCi1WrtvmtFwaGD6V2xgThWptt
Result	Success
Timestamp	Apr 27, 2024 at 18:16:11 Central European Summer Time
Confirmation Status	FINALIZED
Confirmations	MAX
Slot	267,205,880
Recent Blockhash ⓘ	ESVaEGofksadMSdx9nZcDmo5wcxetpSKFqa4kBZfAhSQ
Fee (SOL)	0.00001
Compute units consumed	1,142,140
Transaction Version	LEGACY

See fee paid and compute units consumed
 its over 1M CU. Im sending 10tx per wallet. 2 wallets = 20tx.
 Compute units: 1,14M x 20 = 23M CU = almost half a block.
 A full block = 48M CU.
 If you create 4 wallets you could take over the whole block.

Every tx = 1,14M CU.
 48M / 1,14M = 42 tx. With 42 tx you can take over the whole block.

Outcomes:

you can find 0 hashes, 1 hash or multiple hashes.

Account Input(s)				
#	ADDRESS	CHANGE (SOL)	POST BALANCE (SOL)	DETAILS
1	EFPqkTNmMsVFVCEzgrWrhmXQzq1LiRsbJ6JG7CSYPVMR	-0.00093568	0.98315776	Fee Payer Signer Writable
2	HCS7TpSoYuMXihp4BuxXKLqYjuEFAWrHuzhQSL1dggvJ	+0.00092568	0.00092568	Signer Writable
3	System Program	0	0.000000001	Program
4	Compute Budget Program	0	0.000000001	Program
5	7R2KMCUW1GimTEiS8tp8jJrde2N66yQ1J1MEUTbaPgfq	0	0.00114144	Program

Second row in green you can see how much the tx cost.

In terminal:

solana balance

```
[ubuntu@c2-small-x86-mex2-1:~/my_project$ cargo run
    Finished dev [unoptimized + debuginfo] target(s) in 0.16s
      Running `target/debug/my_project`
Transaction sent successfully.
[ubuntu@c2-small-x86-mex2-1:~/my_project$ solana balance
0.98315776 SOL
[ubuntu@c2-small-x86-mex2-1:~/my_project$ solana address
EFPqkTNmMsVFVCEzgrWrhmXQzq1LiRsbJ6JG7CSYPVMR
```

See how my balance matches what it says in the explorer.

also note the 20tx's since I had 2 wallet generating 10 tx each, after cargo run

I had 2 accounts so Im hitting 2 out 4 threads

If 100 ppl are hitting block - all 4 threads will be used. Scheduler will allocate where space is available.
If I have 4 accounts and I pay most priority fees I can occupy all 4 threads. Rest of 99 ppl will get failed and pay a little bit.

420 hash -> solXEN

we will create a formula in the direction of:

$n * 420\text{hash} = n * \text{AMP solXEN}$,

Where AMP starts at 300 and reduces every 100.000 blocks.

since there 216.000 blocks in 24h given that every block is 400ms, AMP will be reduced to 1 in 139 days.

XN

before mining you will later enter your ETH address. We then create an indexer to match your found hashes to your ETH address, create a leaderboard ala xenblocks and keep track of which ETH address has accumulated how ever many 420hashes and solXEN. Xn can later be airdropped fairly participants based on their found hashes. Rewarding miner.

Next step: priority fees

to be implemented is to be able to specify priority fees so you can compete with however much you want in order to be selected by the leader. That's how we can create a fair auction for blockspace on Solana.

What is actually happening here?

just like you're deploying the miner script to look for XEN11 hashes with xenblocks – you're now instead sending a program to the leader (solana validator that according to the leader schedule has been selected to verify tx's and construct a block) to hash for you and run the 420miner script. The leader finds a 420 hash. The other 3,000 validators will also run the program and in doing so verify that the leaders found solution (420 hash) is correct.

With xenblcks a key is generated with argon2. You then hash to find the XEN11 solution that will match the corresponding key. Here the slot number is the key and the hash that the leader finds corresponds to that slot number. The other validators can then run the same script and thus verify that the slotnumber matches with the same 420 hash. The whole validator cluster will have to run the 420 hashing compute. Then they vote and see that the leaders found solution is the right has for the key(slot number).

Instead of argon2 we here use the cryptographic hashing algorithm sha3. below chart is extremely rough and has flaws but im late for dinner

