Signet

Karl-Johan Alm
DG Lab

@kallewoof
karl@dglab.com

Agenda

- Why Signet?
- What makes it special?
- Let's try it out
- Let's make our own signet

Why Signet?

- Testnet is broken
- Regtest is not a network
- Built to allow for arbitrary number of simultaneous networks
- Easy to use faucets, explorers, etc.
- "Double-spend-as-a-service"

What makes it special?

- 100% centralized ("ran" by someone)
- Requires a signature in the block itself
- Everything else the same; still does proof of work (although usually very low difficulty)

Let's try it out

Tasks:

1. Compile Bitcoin Core · Get docker
2. Compile the "signet" branch · Get kallewoof/signet:0.18
3. Run a node on the default global signet

Task 1. Compile Bitcoin Core

Instructions:

*NIX: https://github.com/bitcoin/bitcoin/blob/master/doc/build-unix.md

macOS: https://github.com/bitcoin/bitcoin/blob/master/doc/build-osx.md

Windows: install *NIX. Then see above. (Or use VirtualBox/etc)

Task 1. Compile Bitcoin Core

Let's test this by starting up a regtest instance:

```
$ cd src
$ ./bitcoind -regtest -daemon
$ AD=$(./bitcoin-cli -regtest getnewaddress)
$ ./bitcoin-cli -regtest generatetoaddress
110 $AD
$ ./bitcoin-cli -regtest getbalance
SHOULD BE > 0 HERE
$ ./bitcoin-cli -regtest stop
```

Task 1. Get docker

Let's install Docker:

mac:
https://hub.docker.com/editions/community/docker-ce-desktop-mac

linux: old versions result in weird shit; check e.g.
https://phoenixnap.com/kb/how-to-install-docker-on-ubuntu-18-04

Windows: I honestly don't know. Good luck.

Task 1. Get docker

Linux users: you may need to be in the "docker" group.

```
$ sudo usermod -a -G docker $USER
```

You may need to close and reopen the terminal for this to take effect. Check with below command:

```
$ groups
[...] docker [...]
```

Task 2. Compile "signet" branch

We need to fetch the signet branch which is in pull request #16411. We then switch to it, and compile again:

```
$ git fetch origin pull/16411/head:signet
$ git checkout signet
$ make -j5
$ ./bitcoind -signet -daemon
# after 10-15 seconds
$ ./bitcoin-cli -signet getblockcount
```

SHOULD BE > 0

Task 2. Get kallewoof/signet:0.18

Mac/Linux: we want to use a wrapper script that binds folders and stuff for us so we don't have to bother.

```
$ cd $HOME/workspace
$ git clone https://github.com/kallewoof/signet-platform.git
$ cd signet-platform/fullnode
$ ./run.sh
```

Windows: uhh... check the `run.sh` script and adapt. You can do these things manually, but probably can't run `.sh` scripts.

Task 3. Run a node on default net

We actually are doing this already. But you have no signet coins yet. Let's get some coins!

Faucet version: get IP address of server

```
$ IP=the IP address # e.g. IP=1.2.3.4
$ cd $HOME/workspace/bitcoin/contrib/signet
$ ./getcoins.sh \
--faucet=http://$IP/claim \
--password=edgy
```

Task 3. [docker] Run on def. net

We actually are doing this already. But you have no signet coins yet. Let's get some coins!

Faucet version: get IP address of server

```
$ IP=the IP address # e.g. IP=1.2.3.4
$ ID=$(docker ps -ql)
$ docker exec $ID \
/workspace/contrib/signet/getcoins.sh \
--faucet=http://$IP/claim \
--password=edgy
```

Docker vs compiled from source

*NIX/mac: make an alias so we are all on the same page:

Docker: `alias bcli="docker exec $ID bitcoin-cli"`
Source: `alias bcli="./bitcoin-cli -signet"`

Windows: whenever you see bcli in the slides, replace it with the `docker exec` command you hopefully figured out by now.

I will be using bcli in the slides from here on.

Task 3. Run a node on default net

Without faucet, get an address and throw it on Slack. I'll try to send you coins:

```
$ bcli getnewaddress
COPYTHISSTRINGTOSLACK
```

If you want to be a person of style, use bech32.

```
$ bcli getnewaddress "" bech32
```

Task 3. Run a node on default net

Check that you got the coins. You should first see them as unconfirmed, then as confirmed once a block is mined.

```
$ bcli getbalance
0.00000000
$ bcli getunconfirmedbalance
SHOULD BE > 0
(after a block has been mined)
$ bcli getbalance
SHOULD BE SAME AS ABOVE (> 0)
```

Task 3. Run a node on default net

If you look at the log file, you should have seen something like

```
2019-09-09Txx:xx:xxZ [default wallet] AddToWallet TXIDHEX  new
```

Go to

https://explorer.bc-2.jp

and put your **TXIDHEX** into the search bar and see if it's there.

Let's make our own network

1. Select signers
2. Create block signing keys
3. Create signing script ("scriptPubKey")
4. Mine the genesis block
5. Start it up
6. Mine blocks (or try to!)

1. Select signers

Volunteers?

We are making a 1-of-*n* signet network.

Any one of the *n* people may generate a new block.

2. Create block signing keys

Only the *n* people need to do this:

```
$ ADDR=$(bcli getnewaddress)
$ PRIVKEY=$(bcli dumpprivkey $ADDR)
$ bcli getaddressinfo $ADDR | grep pubkey

  "pubkey": "02c60c3940e5REDACTEDbd0148cd",
```

Send the pubkey on Slack either to me or in public.

2. Create block signing keys

You then need to write down the private key (don't send this one to anybody).

```
$ echo $PRIVKEY
COPY PASTE THIS TO A NOTEPAD DOC OR SOMETHING.
ALSO ADD THE PUBKEY. SAVE IT.
```

3. Create block signing script

This is standard Bitcoin Script. We want a 1-of-*n* multisig, so:

- *51* "1" (signature count, the "1" in "1-of-*n*")
- *21* "push 0x21=33 bytes" (pubkeys are 33 bytes)
- *pubkey1* "person 1's pubkey"
- *21* …
- *pubkey2*
- …
- *PUBKEY_COUNT*
- *ae* OP_CHECKMULTISIG

3. Create block signing script

Put together, we get something like this:

`5121<pubkey1>21<pubkey2>21<pubkey3>...21<pubkeyN>NNae`

where `NN` is the hex form of *n*, the number of pubkeys.

Now we need to grind proof of work for this block signing script.

4. Mine the genesis block

Anyone, not just the *n* people, can mine any genesis block.

Everyone simply has to agree which one to use.

To grind, use an *existing* signet instance (like the default one) and do

```
$ bcli grindblock SCRIPT
12345 <-- this is the proof of work nonce
```

4. Mine the genesis block

Assuming we get 12345 as the nonce value, the tuple (script, 12345) represents a unique Signet network.

We represent this network using two parameters, `-signet_blockscript` and `-signet_genesisnonce`.

It's also useful to have at least one seed node, so people don't have to manually connect to a peer to get synced up but we are skipping that here. You can use `-signet_seednode=HOST:PORT` to provide seed node(s).

4. Mine the genesis block

Let's make a new data directory with a config file to track this.

```
$ D=$HOME/signet-edgy
$ mkdir $D
$ echo "signet=1
[signet]
daemon=1
signet_blockscript=THE BLOCK SCRIPT
signet_genesisnonce=THE NONCE" > \
$D/bitcoin.conf
```

4. [docker] Mine genesis block

There is a `bitcoin.conf` file in `$HOME/docker-signet`. We want to tweak it:

```
$ D=$HOME/docker-signet
$ echo "signet=1
[signet]
daemon=1
signet_blockscript=THE BLOCK SCRIPT
signet_genesisnonce=THE NONCE" > \
$D/bitcoin.conf
```

5. Start it up

We did all this while running the default Signet. Let's shut down and restart using our new configuration.

```
$ bcli stop
$ ./bitcoind -datadir=$D
$ bcli addnode IPADDR onetry
$ bcli getconnectioncount
SHOULD BE > 0
```

5. [docker] Start it up

We did all this while running the default Signet. Let's shut down and restart using our new configuration.

```
$ bcli stop
$ ./run.sh # from signet-platform/fullnode
$ ID=$(docker ps -ql)
$ alias bcli="docker exec $ID bitcoin-cli"
$ bcli addnode IPADDR onetry
$ bcli getconnectioncount
SHOULD BE > 0
```

5. Start it up

If you are one of the *n* people, one more step:

```
$ bcli importprivkey $PRIVKEY
```

6. Mine blocks (or try to!)

Everyone should try this, not just the people with pubkeys.

```
$ ADDR=$(bcli getnewaddress)
$ BLOCK=$(bcli getnewblock $ADDR)
$ SIGNED=$(bcli signblock $BLOCK)
# above will fail for non-n people
$ bcli grindblock $SIGNED
abc123… <-- block hash
```

6. Mine blocks automatically

Instead of making blocks one by one, there is a script that automates this. It also has fancy features like delays to keep difficulty target low.

```
$ cd $HOME/workspace/bitcoin/contrib/signet
$ ./issuer.sh 0 -datadir=$HOME/signet-edgy
```

Thank you

Karl-Johan Alm
DG Lab

@kallewoof
karl@dglab.com