



Disappeared Coins: Steal Hashrate in Stratum Secretly

Xin Liu, Rui Chong, Yuanyuan Huang, Yingli Zhang, Qingguo Zhou

Content

01

Mining and Pool

Mining pool

What's the proof of work(PoW)

Introduce the concept of transaction and coinbase in Blockchain

03

Steal Hashrate in Stratum Secretly

A direct job insertion attack model proposed by others before and the reason why the attack failed.

Purpose of our attack models

Job injection based on set_extranonce attack model

Time segment attack model

02

Stratum Protocol

Stratum V2 is rarely used so we still aim at Stratum V1

The features and communication process in Stratum V1

04

Proof of Concept

Tools and equipment we used for PoC

Shows the execution of attack scripts of the two model through screenshots and videos

1 Mining and Pool

Mining Pool

It is already difficult for a single independent miner to stably mine a block to obtain revenue, so many miners combine their computing power in the form of cooperative mining to establish a mining pool.

Distribute Profit

Every miner who joins the mining pool can get profit according to the contribution to the mining pool.

Assign Jobs

According to the computing ability of miners, each miner is assigned a share target with different difficulty (which is far less than the bitcoin network difficulty)

Settlement Strategy

PPLNS (Pay Per Last N Shares), PPS (Pay Per Share), PPS+ (Pay Per Shares Plus), FPPS (Full Pay Per Shares)

1 Mining and Pool

Proof of Work

Nakamoto consensus:

every miner try to figure out the solution of the Bitcoin puzzle. If someone finds out the solution, the block he mined will be the next block in blockchain after other participants verify that his block is valid.



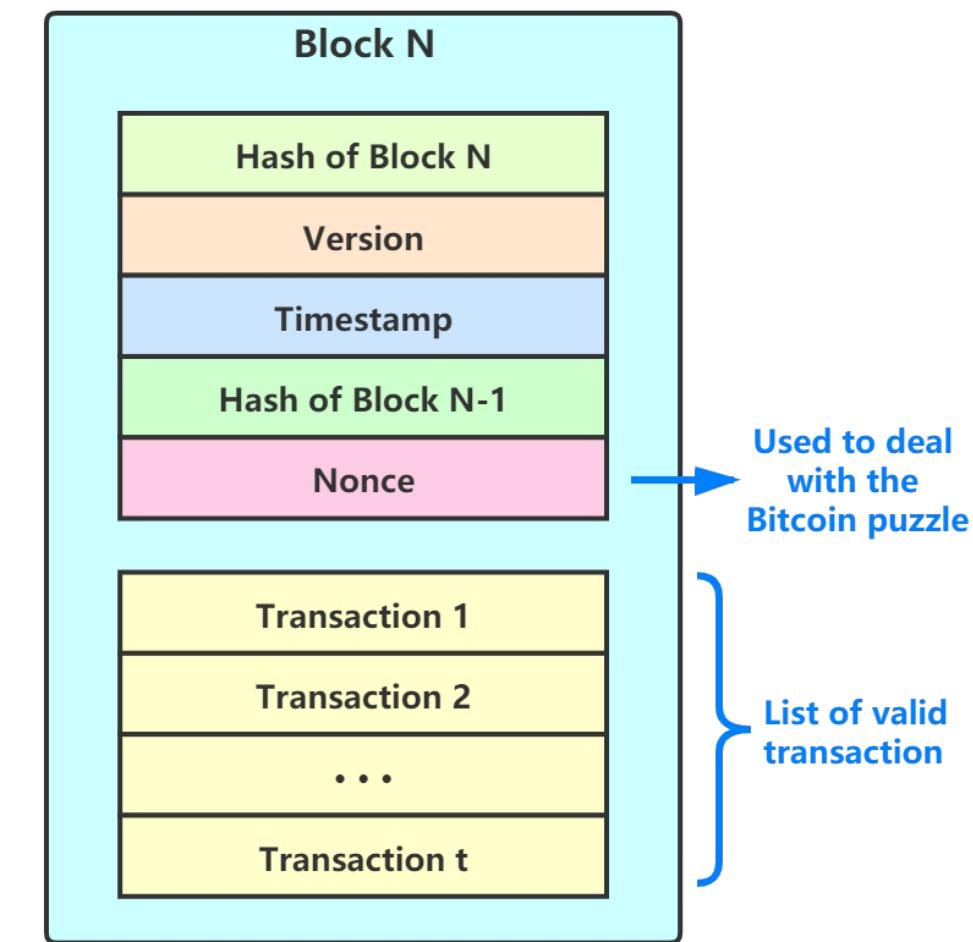
Proof of Work

1 Mining and Pool

Transaction

All transactions need to be verified whether they are valid by *miners*, then the valid transactions are packaged into a block.

And the new block is generated through proof of work(PoW) and added to the blockchain.

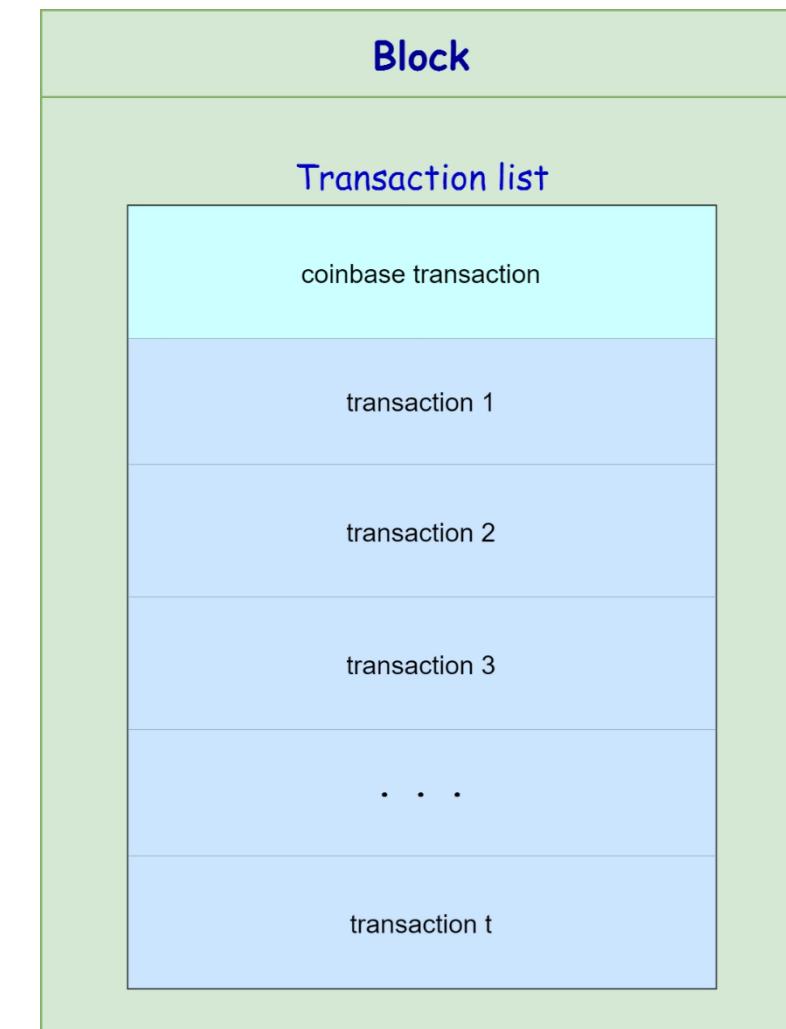


1 Mining and Pool

Transaction

Each block will collect a set of pending transactions in the Bitcoin network. But the first transaction in each block is special, called *coinbase* transaction.

It is used to specify that when the current block is mined through the proof of work, the miner will receive the corresponding block reward (currently 6.25 BTC) and fees.



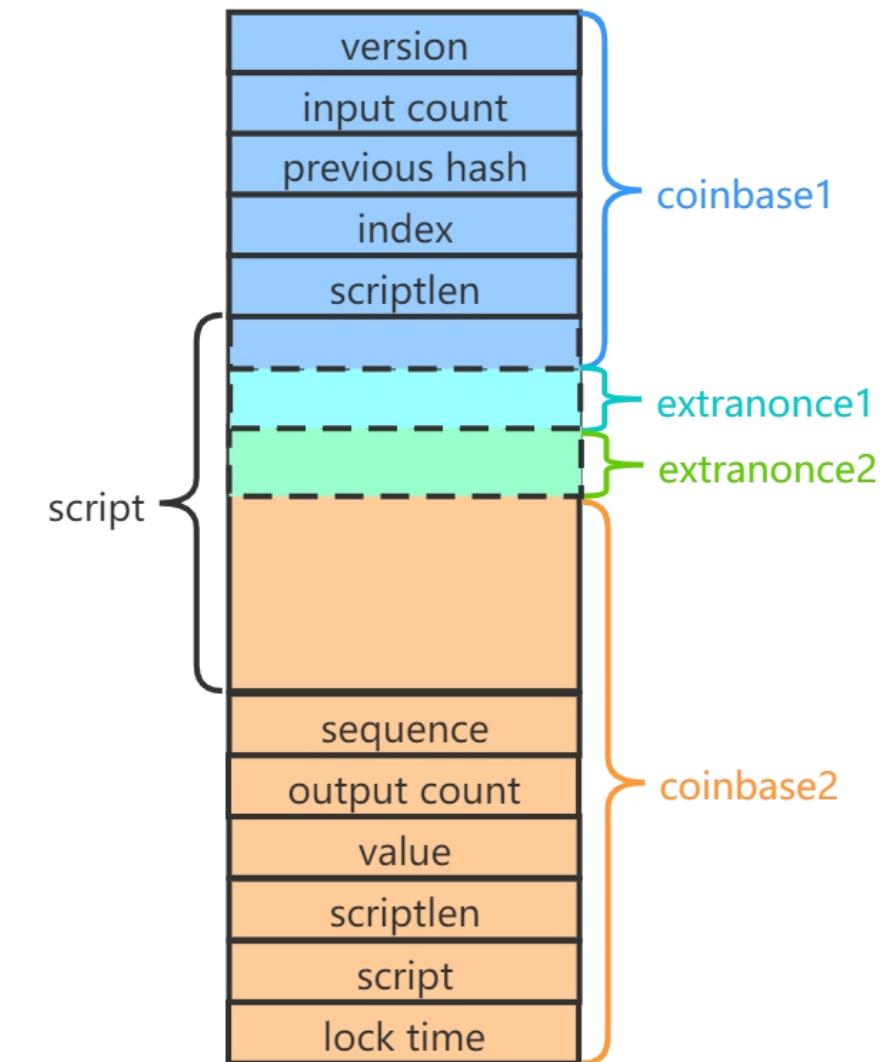
1 Mining and Pool

Coinbase Transaction

The coinbase transaction can be divided into 4 parts: coinbase1, extranonce1, extranonce2, coinbase2

Coinbase1: covers the first 5 fields of coinbase transaction and part of the script. Except for the version number, other fields are meaningless for all mining machines, because there is no input party for the coinbase transaction.

Extranonce1: covers a small part of the script in the coinbase transaction. It is specified by the mining pool and is unique for each connection between the mining pool and the mining machine.



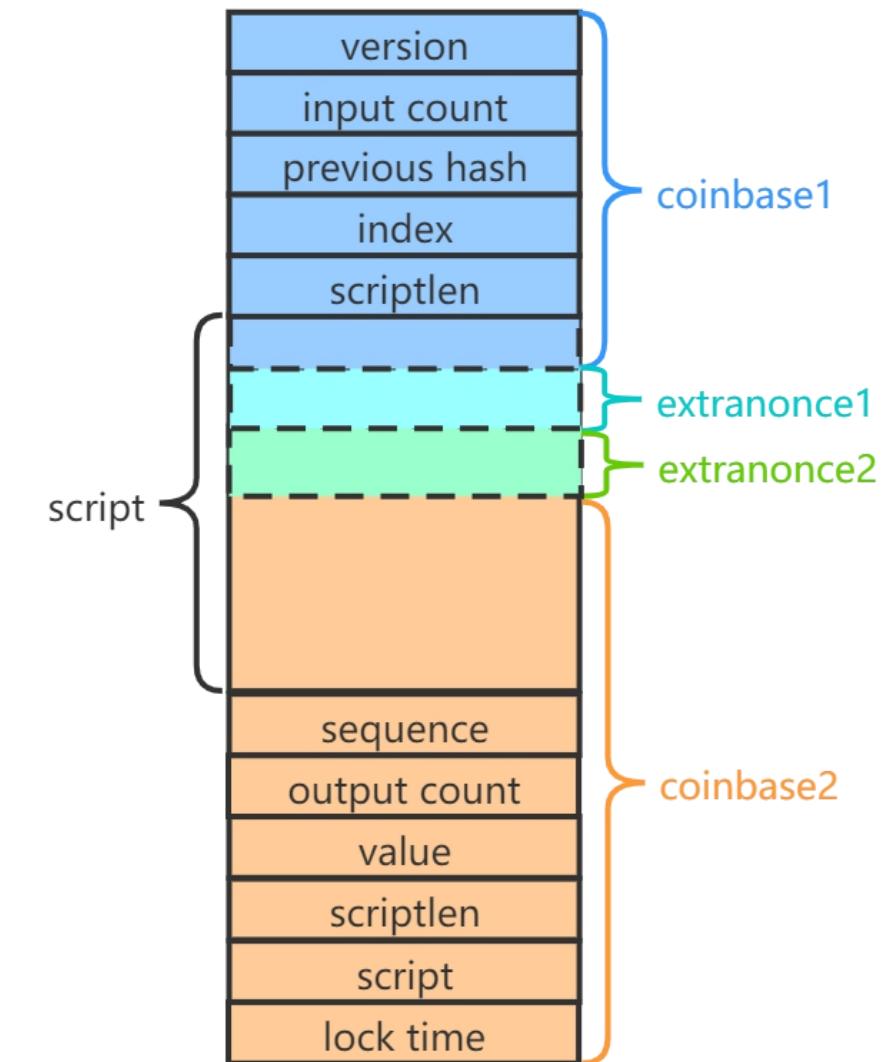
1 Mining and Pool

Coinbase Transaction

The coinbase transaction can be divided into 4 parts: coinbase1, extranonce1, extranonce2, coinbase2

Extranonce2: covers a small part of the script in the coinbase transaction, immediately after extranonce1. When miner solves Bitcoin puzzle with proof-of-work, if the *nonce* is exhausted, extranonce2 will be incremented to figure out the puzzle.

Coinbase2: covers the rest of coinbase transaction



2 Stratum Protocol

The communication between miners and mining pool needs to follow mining protocols, such as Stratum protocol, GetBlockTemplate protocol and GetWork protocol.

Stratum protocol is widely used, and Braiins put forward an upgraded version at the end of 2019, called Stratum V2.

Stratum protocol is based on TCP/IP plaintext transmission protocol, using JSON-RPC data format.

2 Stratum Protocol

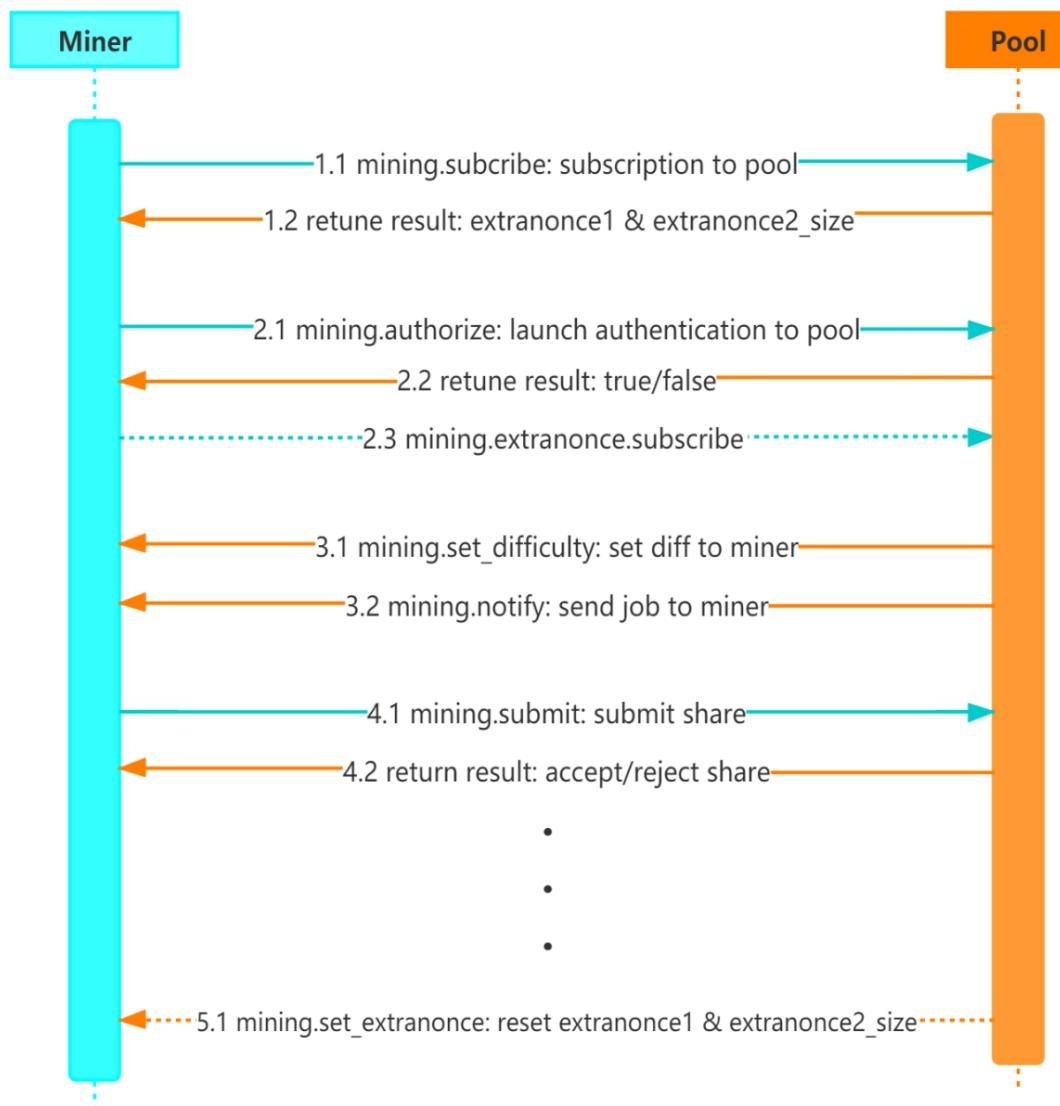
Mining Pool	Stratum V1	Stratum V2
F2Pool	✓	✗
Poolin	✓	✗
Btc.com	✓	✗
AntPool	✓	✗
SlushPool	✓	✓

The Stratum V2 has not yet been widely used and we don't have the corresponding mining machine which supports Stratum V2. Mainstream mining machine (e.g. Antminer and Whatsminer) and mining software (e.g. Cgminer and Ccminer) currently do not support Stratum V2.

Since Stratum V2 has not yet been widely popularized, our attack models are only aimed at Stratum V1.

Next, the communication process of Stratum V1 will be introduced in detail.

2 Stratum Protocol

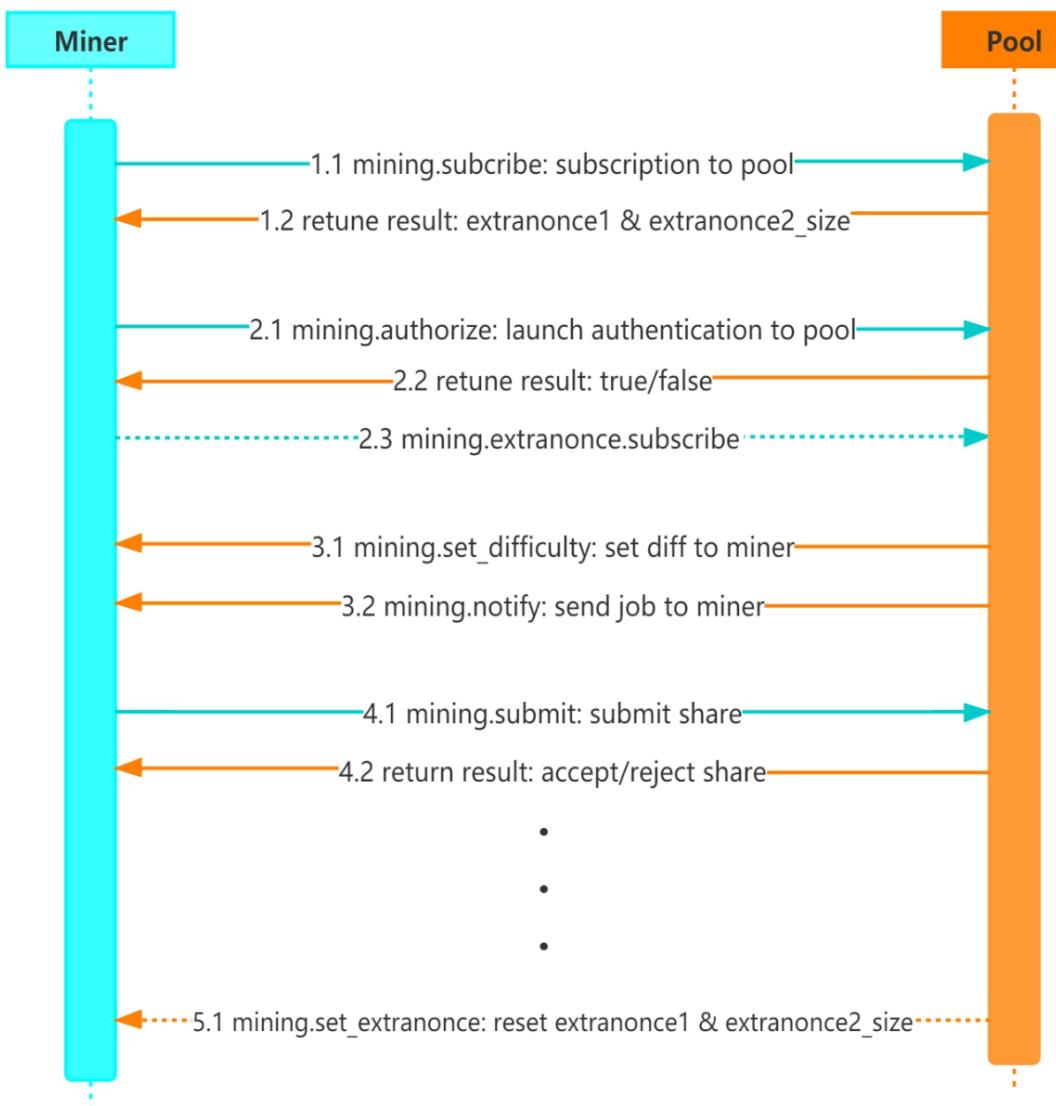


Step 1.1: First, miner initiates a subscription request to the mining pool to establish a connection through the *mining.subscribe* method

mining.subscribe("user agent/version", "extranonce1")

Step 1.2: After receiving the above subscription message, the mining pool will return the subscription_id, extranonce1 and extranonce2_size (unit: byte)

2 Stratum Protocol

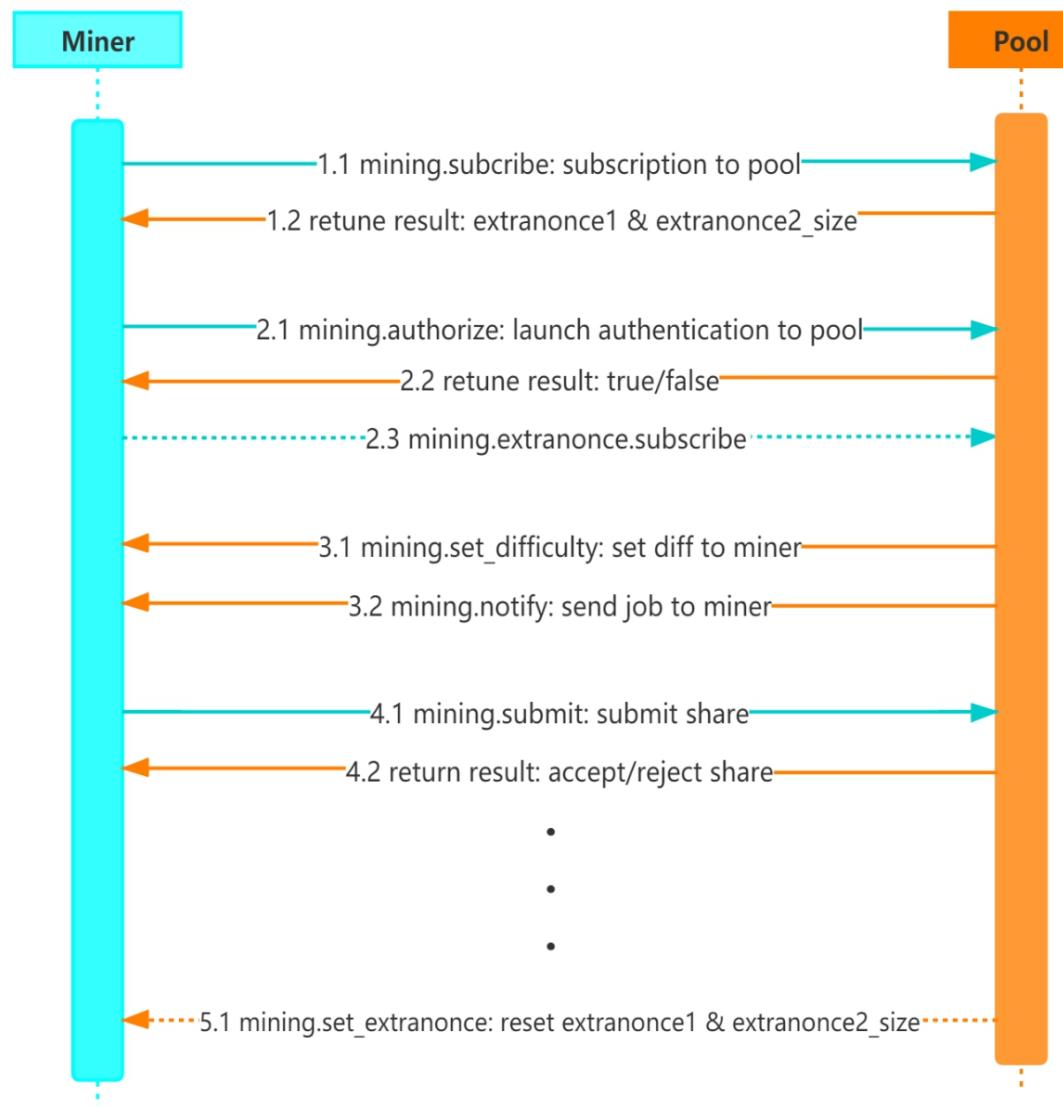


Step 2.1: Miners use the *mining.authorize* method to send authorization request to the mining pool

mining.authorize("username", "password")

Step 2.2: mining pool returns true or false to notify miner whether authorization is successful

2 Stratum Protocol



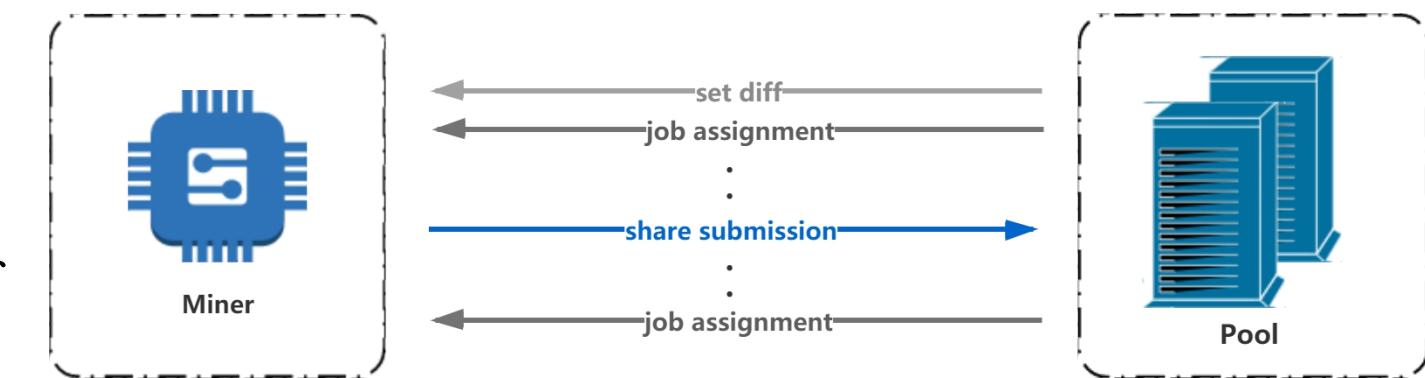
Step 2.3: After that, the miner will use the *mining.extranonce.subscribe* method to inform the mining pool that it supports the *mining.set_extranonce* method

mining.extranonce.subscribe()

2 Stratum Protocol

Step 3.1: After successful subscription and authorization, the pool will negotiate the difficulty value with miner through *mining.set_difficulty* method.

The difficulty needs to be compatible with the computing power of mining machine.



Step 3.2: Mining Pool assigns jobs to miner through *mining.notify* method.

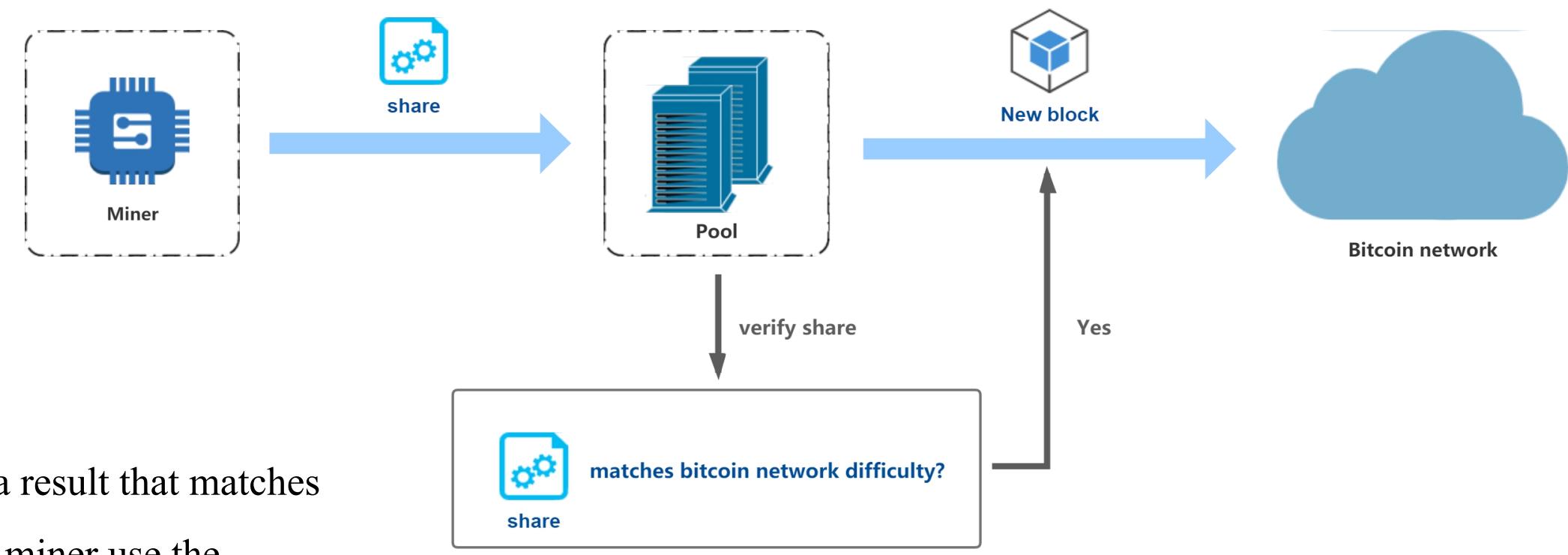
2 Stratum Protocol

mining.set_difficulty(difficulty)

mining.notify(Job ID, Hash of previous block, coinbase1, coinbase2, List of merkle branches, version, nBits, nTime, Clean Jobs)

- Job ID: the id number of job
- Hash of previous block: used to build the block header
- coinbase1, coinbase2: two unchangeable parts in coinbase transaction
- List of merkle branches: used to build the final merkle root
- Version: bitcoin block version
- nBits: the encoded bitcoin network difficulty
- nTime: the current time
- Clean Jobs: if true, it means that the miner needs to stop the current work and execute this new work

2 Stratum Protocol



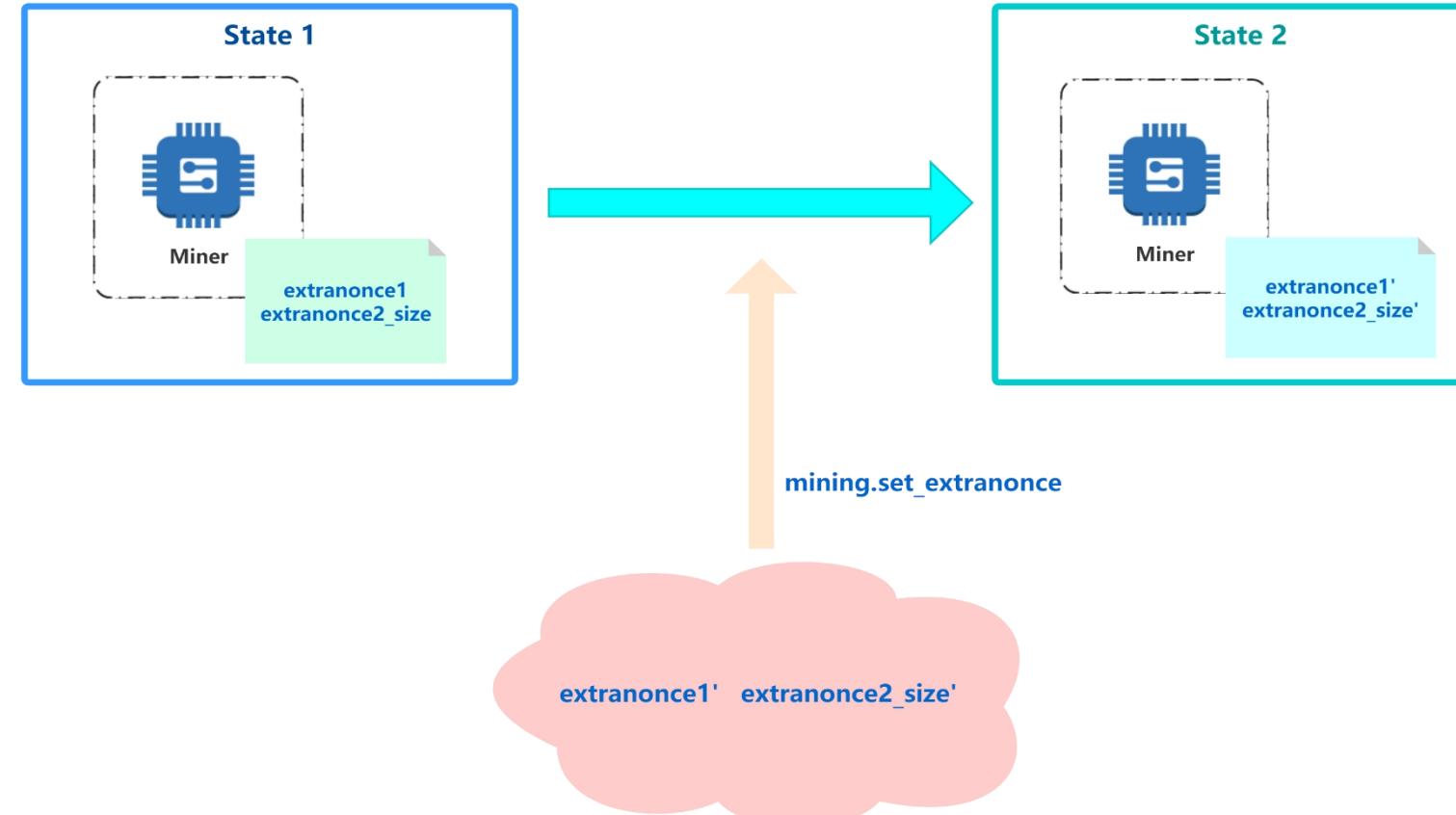
Step 4.1: When a miner finds out a result that matches the difficulty that mining pool set, miner use the *mining.submit* method to submit the result, which called *share*.

```
mining.submit("username", "Job ID", "extraNonce2", "nTime", "nonce")
```

2 Stratum Protocol

Step 5.1: using `mining.set_extranonce` method reset the value of extrance1 and bytes of extraonce2 saved by miner during the subscription phase (step 1.2)

```
mining.set_extranonce("extranonce1", extranonce2_size)
```

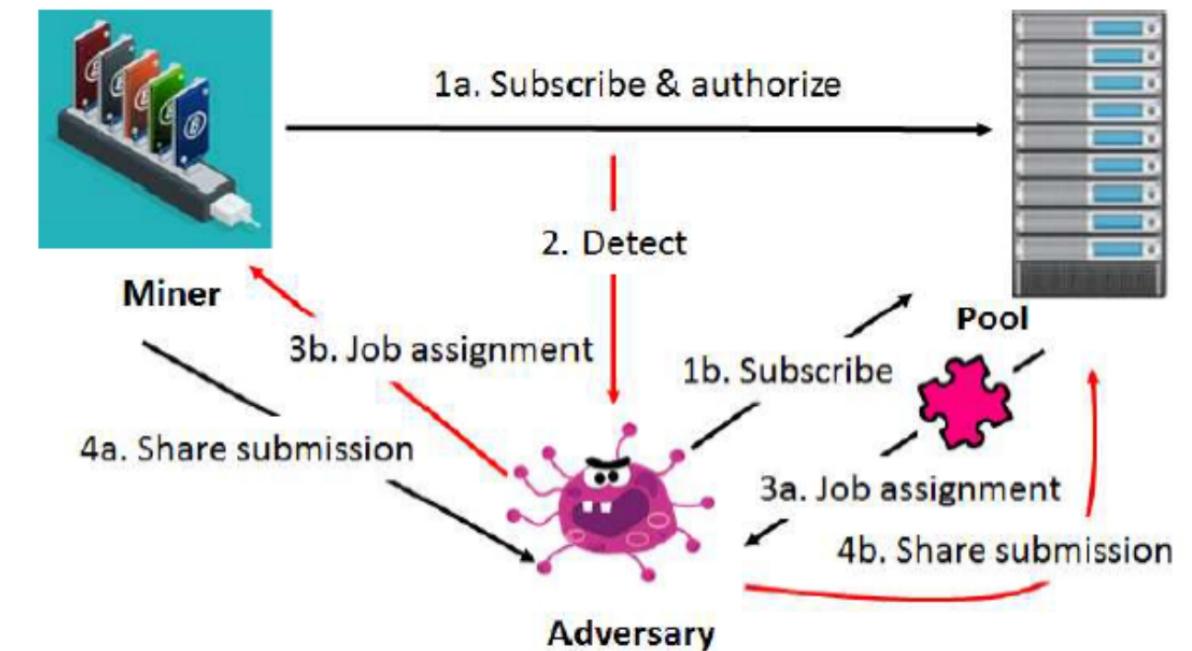


3 Steal Hashrate in Stratum Secretly

3.1 Why direct job insertion is not feasible

Direct job insertion based on TCP hijacking is not feasible in our situation.

The adversary hijacks the communication between the miner and the normal mining pool, and inserts the job of the malicious mining pool built by the adversary directly into the job flow between the normal mining pool and the miner.



Direct job insertion attack model proposed by others before

3 Steal Hashrate in Stratum Secretly

3.1 Why direct job insertion is not feasible

Miner will save extranonce1 specified by normal mining pool during subscription in Stratum. And extranonce1 is used to construct *coinbase* and calculate the *share*.

The extranonce1 specified by normal pool must be different from the one specified by malicious mining pool.

extranonce1_normal != extranonce1_malicious

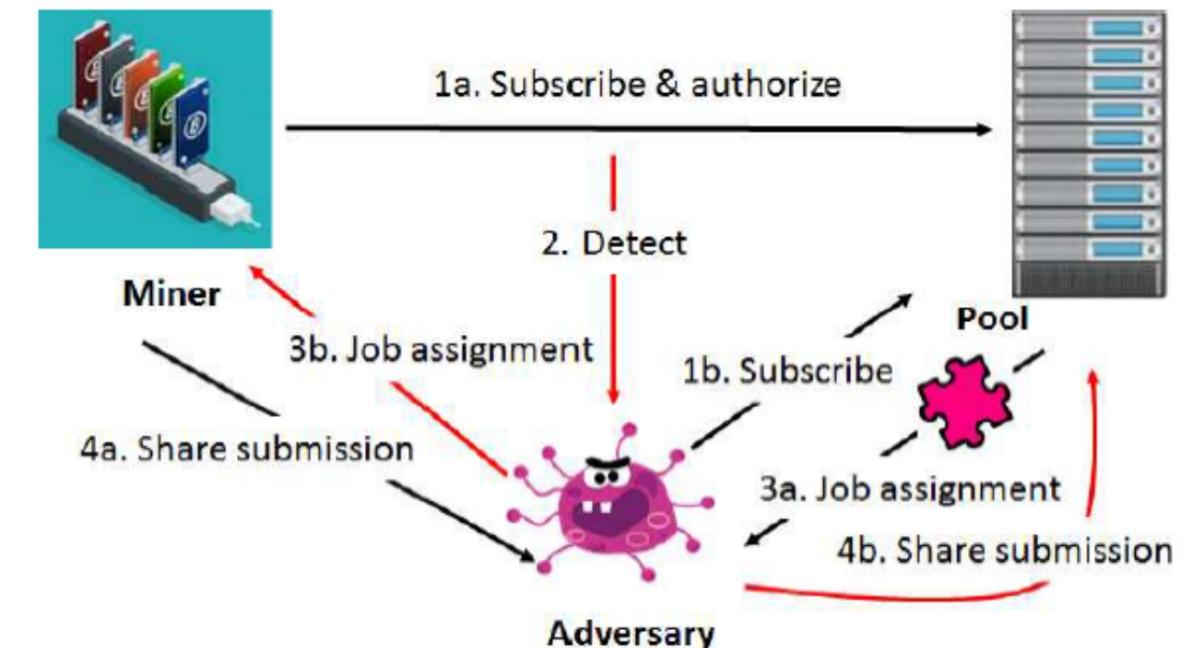
So that the *share* calculated by miner based on extranonce1_normal is different from the one calculated with extranonce1_malicious

3 Steal Hashrate in Stratum Secretly

3.1 Why direct job insertion is not feasible

Therefore, the *share* belonging to the malicious pool however calculated with extrnonce1_normal will be rejected by the malicious pool in verification for *share* in Stratum.

That's the reason why the attack cannot be successfully implemented.



Direct job insertion attack model proposed by others before

3 Steal Hashrate in Stratum Secretly

3.3 Preconditions of attack

The purpose of the two man-in-the-middle attacks we proposed is to steal the hashrate of the miners to work for the malicious mining while the miner and the normal mining pool are barely aware of it.

Key: FIX extranonce1

Malicious pool: the bitcoin mining pool built by the adversary

Normal pool: the bitcoin mining pools maintained by company

3 Steal Hashrate in Stratum Secretly

3.2 Preconditions of attack

Under normal conditions:

the miner sends subscription and authorization requests to the mining pool.

Once subscription and authorization is successful, the mining pool will allocate work to the miner.

When the miner finds out the *share*, it will submit it.

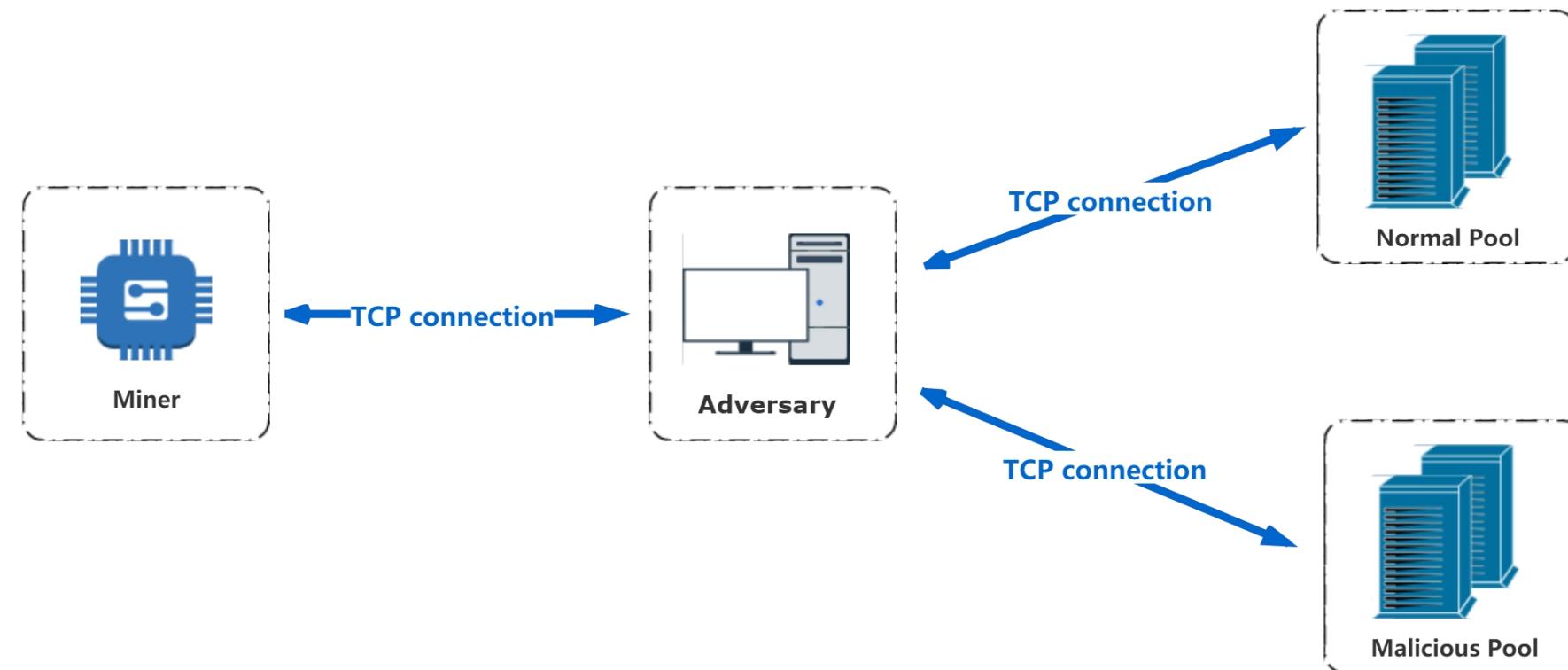


3 Steal Hashrate in Stratum Secretly

3.2 Preconditions of attack

Using man-in-the-middle attack to hijack the communication between miners and mining pools:

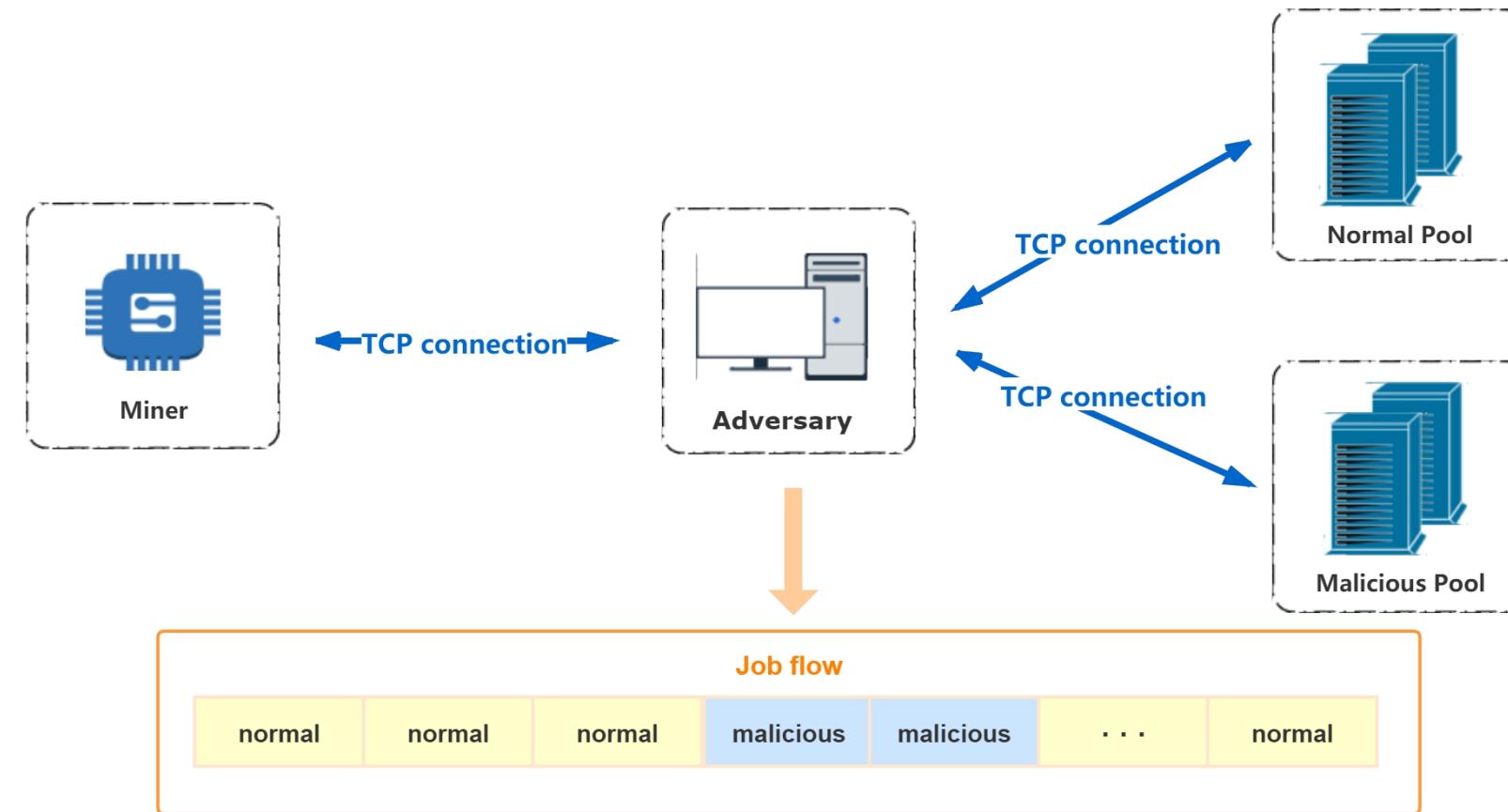
The adversary server hijacks the TCP communication between the miner and the mining pool and then maintains the TCP connections with the miner and the mining pools.



3 Steal Hashrate in Stratum Secretly

3.3 Job injection based on set_extranonce

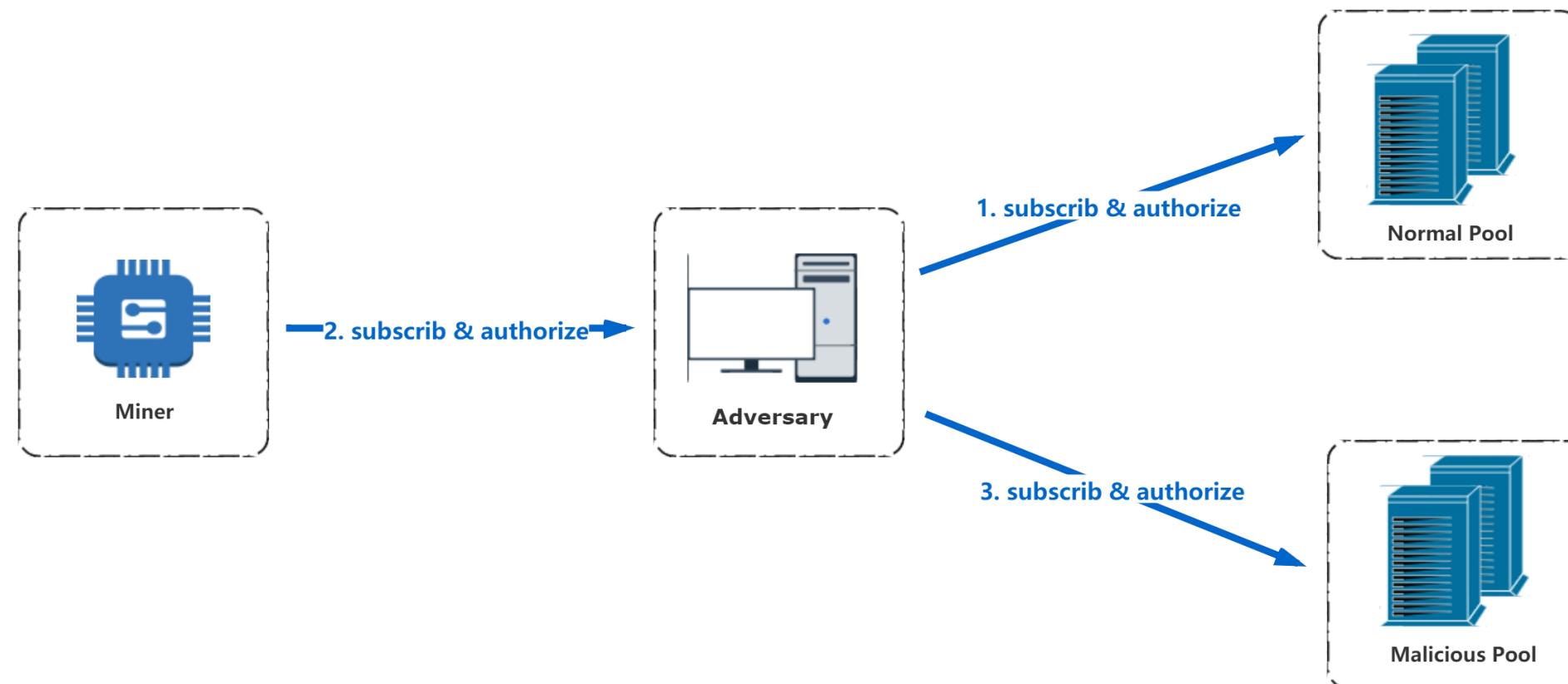
The adversary inserts the job from the malicious pool into the job flow of the normal pool.



3 Steal Hashrate in Stratum Secretly

3.3 Job injection based on set_extranonce

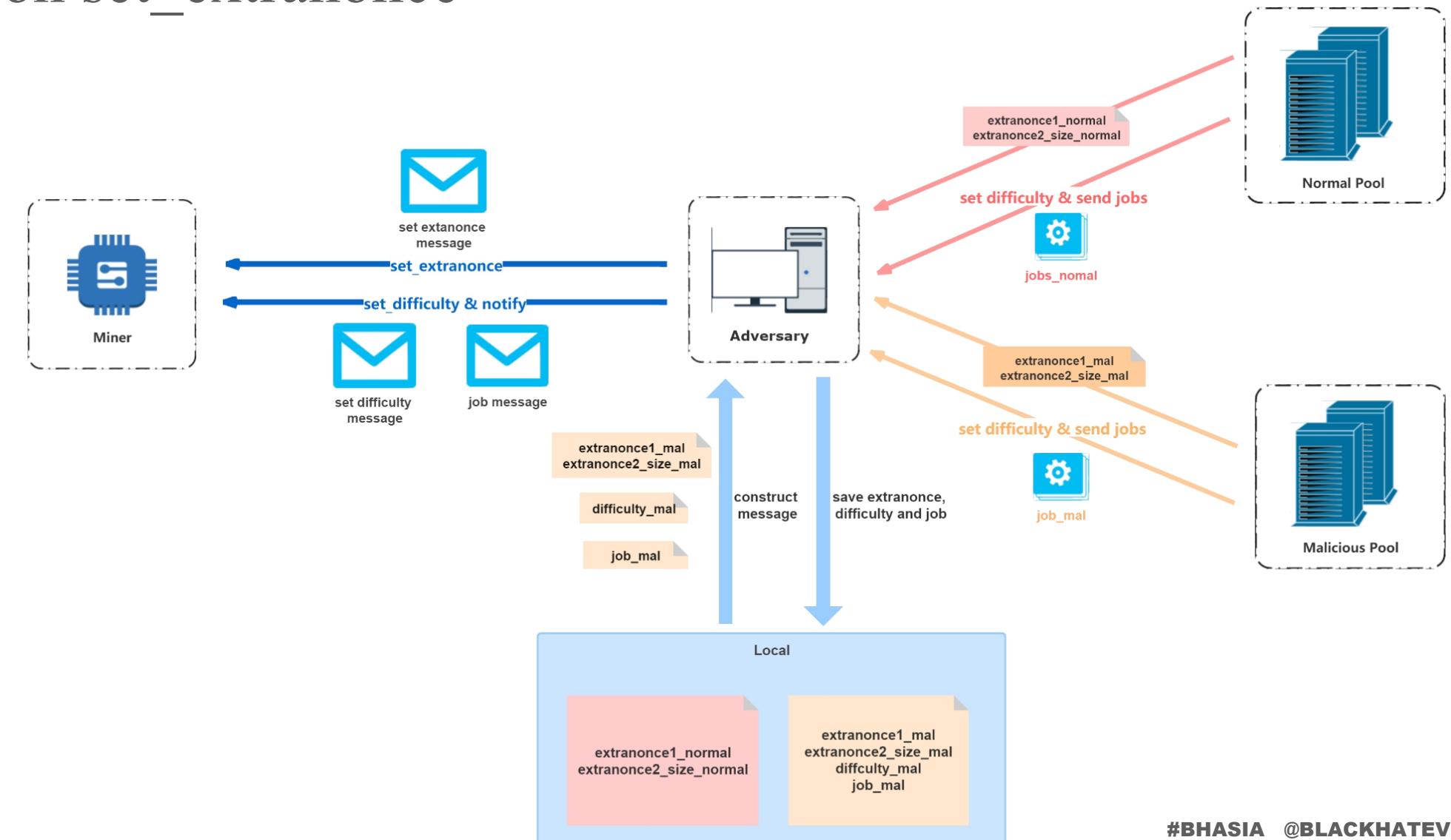
The adversary establishes a connection with the malicious mining pool, and then listens to the subscription and authorization message of the miner and forwards them to the normal mining pool.



3 Steal Hashrate in Stratum Secretly

3.3 Job injection based on set_extranonce

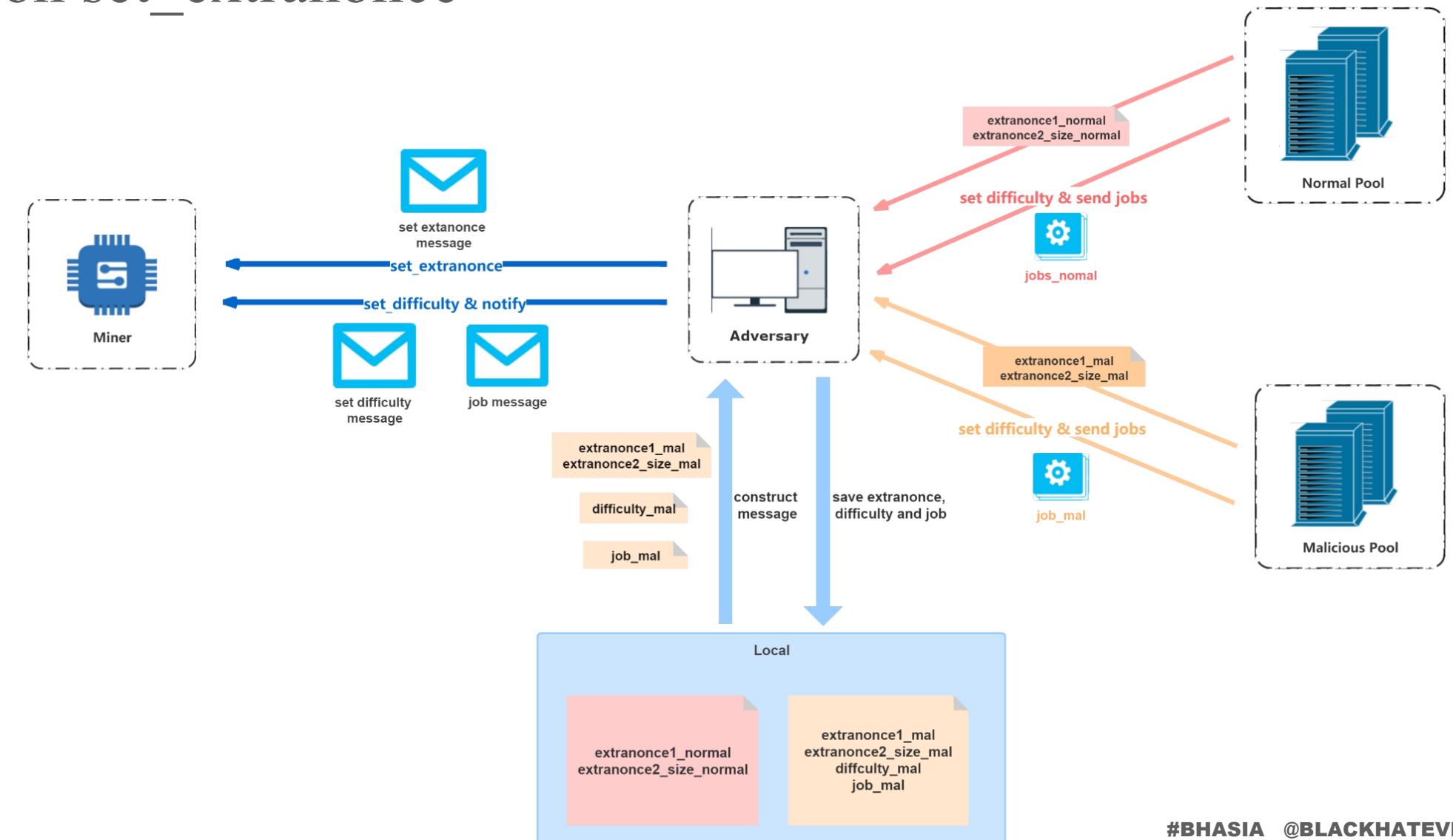
After receiving the subscription message, the mining pools will respond to miner with extranonce1 and extranonce2_size which will be saved by the adversary. What's more, the adversary will save the difficulty and job information sent by the mining pools.



3 Steal Hashrate in Stratum Secretly

3.3 Job injection based on set_extranonce

When the miner works for the normal pool for a period of time, the adversary will reset extranonce with extranonce1 and extranonce2_size of the malicious pool, and construct *set_difficulty* and *notify* message, and then sends them to miner to make miner work for malicious pool.

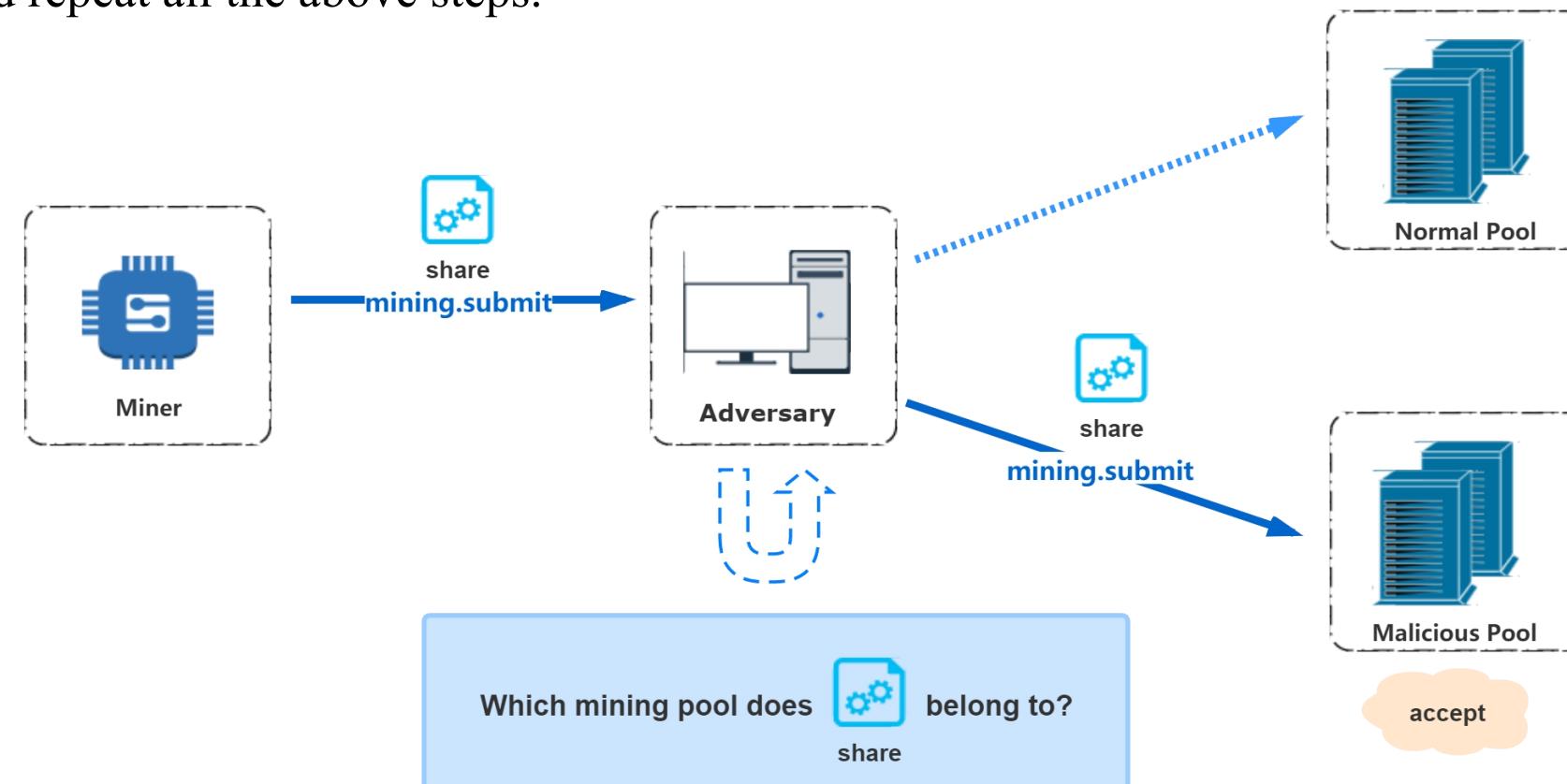


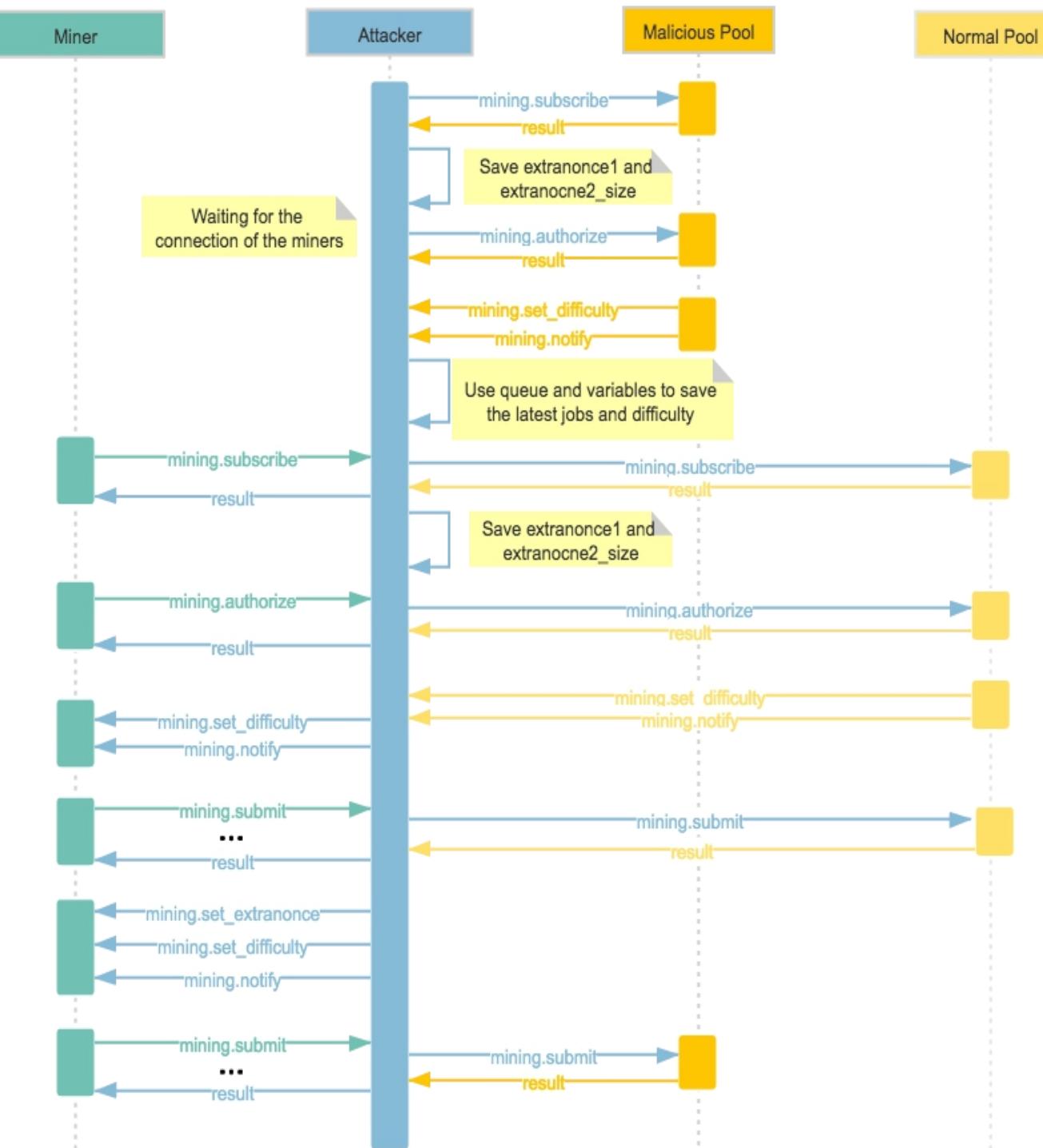
3 Steal Hashrate in Stratum Secretly

3.3 Job injection based on set_extranonce

Then, the adversary forwards *shares* to malicious pool.

After completing all the above steps, the *mining.set_extranonce* method will be used again to switch the miner to work for the normal mining pool, and repeat all the above steps.





Sequence diagram

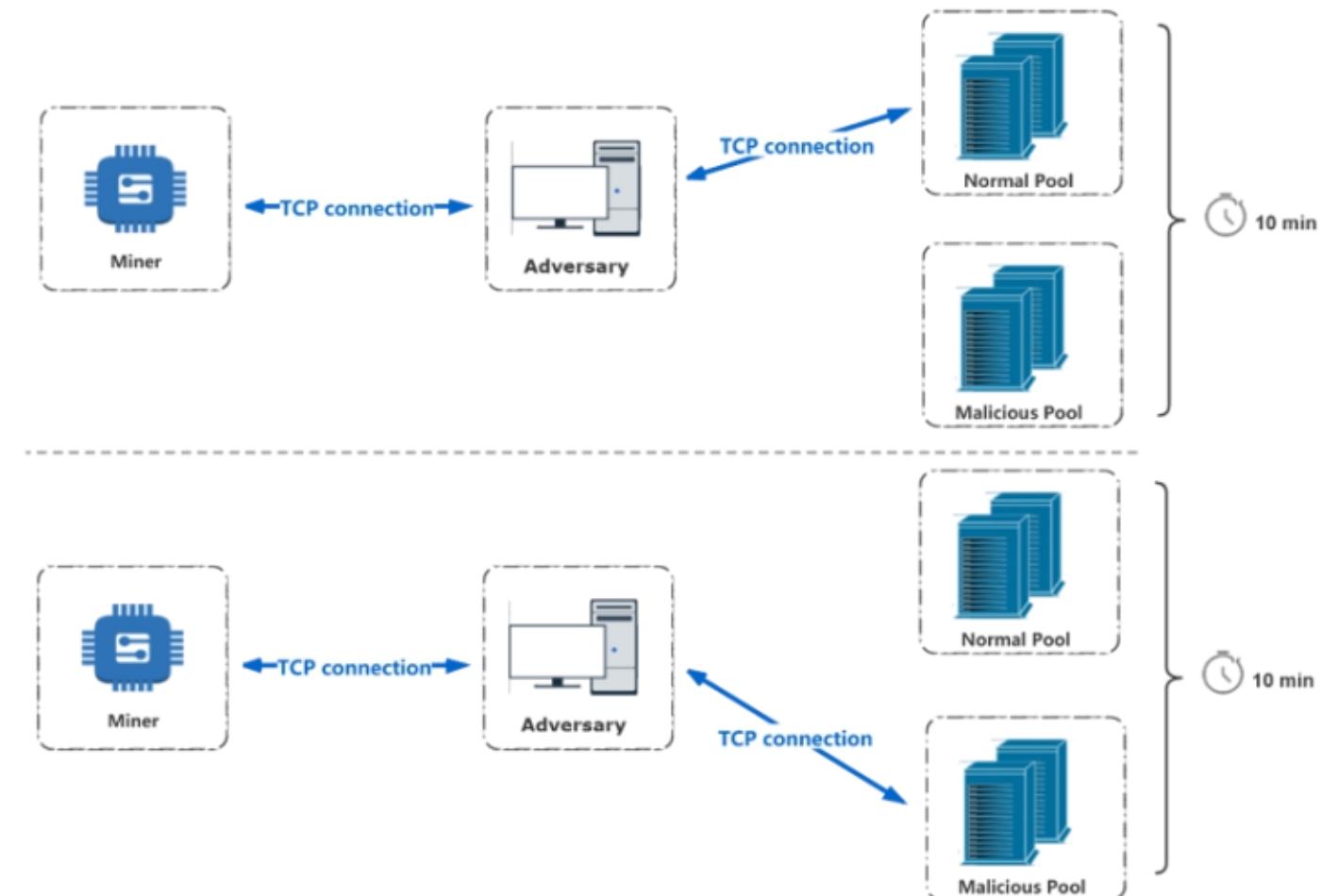
If attackers only steal 5% of the hashrate, it will be very unobvious. The miners and the pool can hardly find these attacks.

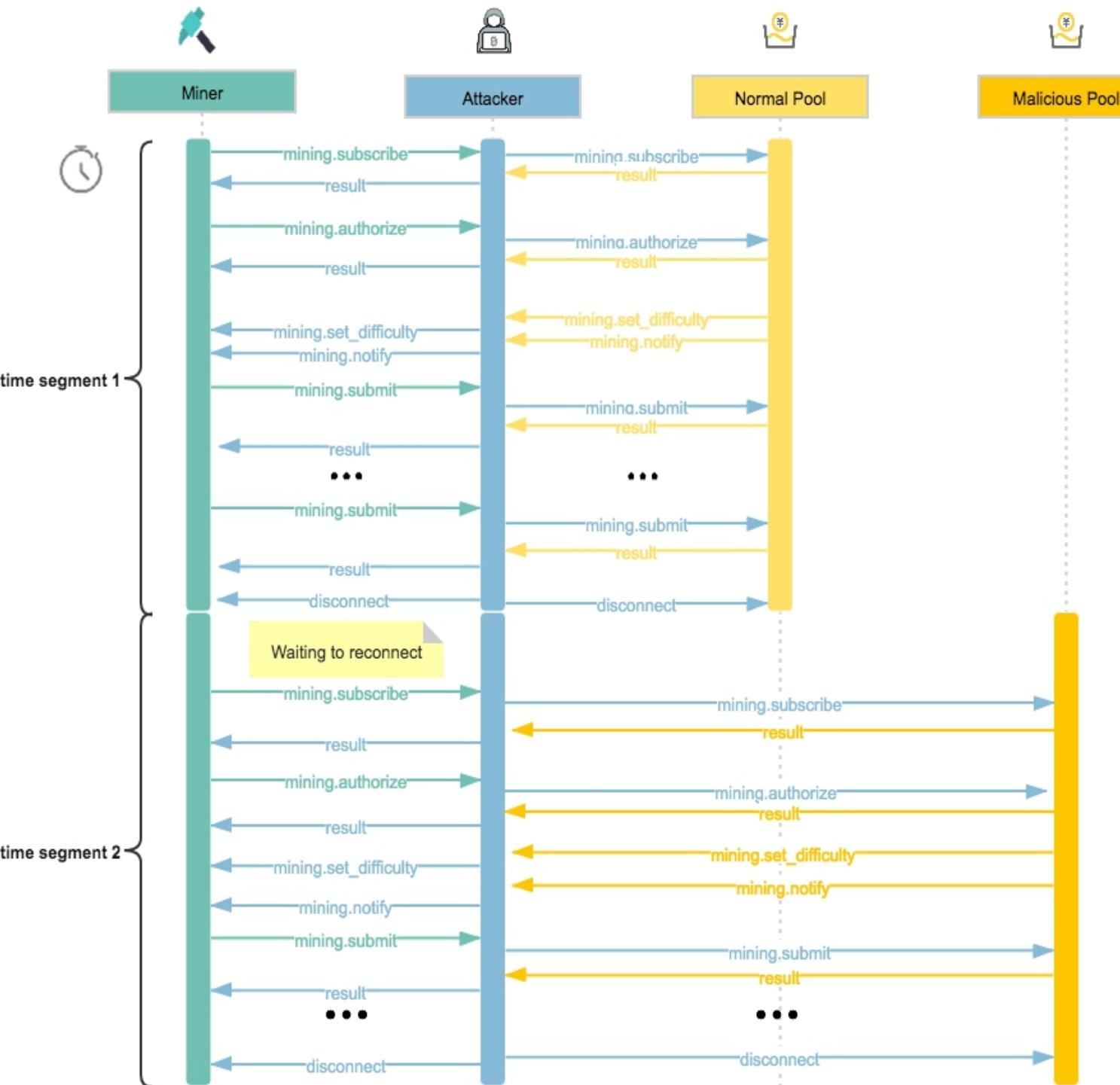
Besides, it has really good performance.

3 Steal Hashrate in Stratum Secretly

3.4 Time segment

The adversary hijacks the TCP communication between the miner and the normal pool, and make it work for two mining pool respectively at different time periods.





Sequence diagram

This method is based on the reconnection scheme.

At the end of the first time segment, the adversary disconnects from the miner and the normal pool.

In the second time segment, the adversary is waiting for reconnection from the miner. And then, make a new connection to the malicious pool.

The extranoce1 is refreshed.

4 Proof of Concept

4.1 BTCPool

BTCPool is backend system of <https://pool.btc.com>.

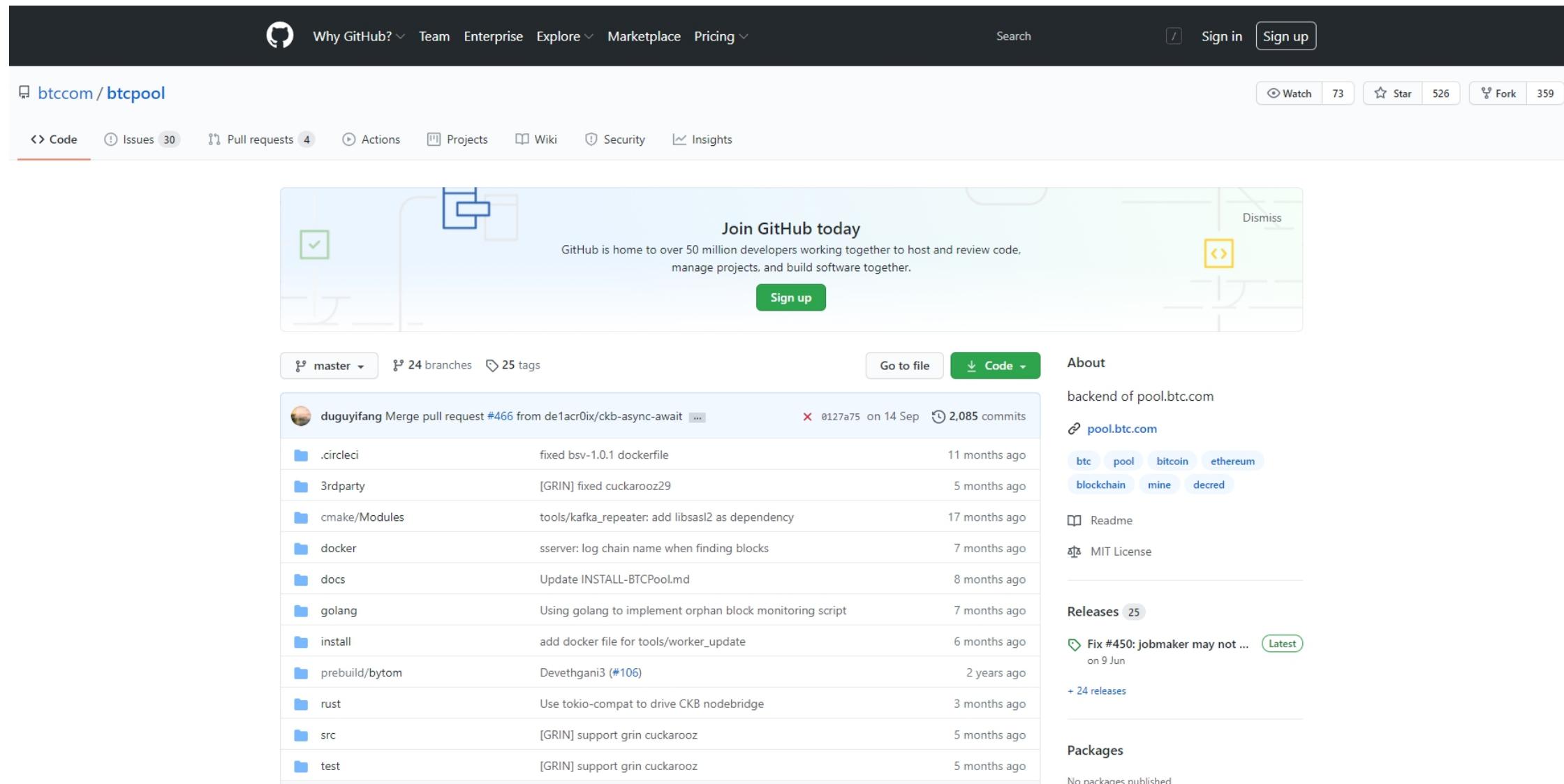
It's an open source project on Github. <https://github.com/btccom/btcpool>

We built two mining pool to simulate the normal pool and the malicious pool using BTCPool on servers.

OS: Ubuntu 16.04 64 bits

4 Proof of Concept

4.1 BTCPool



The screenshot shows the GitHub repository page for `btccom/btcpool`. The repository has 73 stars, 526 forks, and 359 open issues. The code tab is selected, showing the master branch with 24 branches and 25 tags. The repository's purpose is described as the "backend of pool.btc.com". It uses labels like btc, pool, bitcoin, ethereum, blockchain, mine, and decred. The releases section includes a fix for jobmaker and 24 other releases. The packages section indicates no packages have been published.

Join GitHub today

GitHub is home to over 50 million developers working together to host and review code, manage projects, and build software together.

Sign up

Dismiss

master 24 branches 25 tags

duguyifang Merge pull request #466 from de1acr0ix/ckb-async-await ... 0127a75 on 14 Sep 2,085 commits

.circleci fixed bsv-1.0.1 dockerfile 11 months ago

3rdparty [GRIN] fixed cuckaroz29 5 months ago

cmake/Modules tools/kafka_repeater: add libsasl2 as dependency 17 months ago

docker sserver: log chain name when finding blocks 7 months ago

docs Update INSTALL-BTCPool.md 8 months ago

golang Using golang to implement orphan block monitoring script 7 months ago

install add docker file for tools/worker_update 6 months ago

prebuild/bytom Devethgan3 (#106) 2 years ago

rust Use tokio-compat to drive CKB nodebridge 3 months ago

src [GRIN] support grin cuckaroz 5 months ago

test [GRIN] support grin cuckaroz 5 months ago

About

backend of pool.btc.com

pool.btc.com

btc pool bitcoin ethereum blockchain mine decred

Readme

MIT License

Releases 25

Fix #450: jobmaker may not ... Latest on 9 Jun + 24 releases

Packages

No packages published

#BHASIA @BLACKHATEVENTS

4 Proof of Concept

4.2 Ccminer

Ccminer is a kind of mining software that needs to be installed in miner's machine.

It's an open source software on Github. <https://github.com/tpruvot/ccminer/tree/linux>

Version: tpruvot linux

4 Proof of Concept

4.2 Ccminer

[tpruvot / ccminer](#)
forked from cbuchner1/ccminer

Watch 346 Star 1.5k Fork 1.5k

Code Pull requests 2 Actions Projects Wiki Security Insights

Join GitHub today
GitHub is home to over 50 million developers working together to host and review code, manage projects, and build software together.
[Sign up](#)

Dismiss

linux 9 branches 56 tags Go to file Code

This branch is 930 commits ahead of cbuchner1:master. #75 Compare

tpruvot Update makefile for CUDA 11.1 compat. 1eb8dc6 3 days ago 965 commits

Algo256	upgrade BIGNUM class for openssl 1.1	3 years ago
JHA	jha: remove deprecated warning	3 years ago
api	api: update websocket sample (#31)	3 years ago
compat	Handle lyra2v3 algo, for VTC fork	2 years ago
crypto	Linux aarch64 support (ARM64)	17 months ago
equi	Linux aarch64 support (ARM64)	17 months ago
heavy	change defaults to handle cuda 9+, disable heavy and SM 2.x	3 years ago
luby	Add missing real cuda arch checks	4 years ago
lyra2	lyra2v3 changes cleanup	2 years ago
neoscrypt	neoscrypt: add extra space for recent vstudio madness	3 years ago

About
CUDA Open Source miner project, for most nvidia cards
[bitcointalk.org/?topic=770064](#)

Readme
GPL-3.0 License

Releases 56
[v2.3.1 Windows binaries](#) (Latest) on 31 Jan 2019
+ 55 releases

Packages
No packages published

Languages

4 Proof of Concept

4.3 GPU

GeForce RTX 2080 Ti * 3 (ASIC Miners are SOLD OUT >.<)

```
~$ nvidia-smi
Wed Dec 16 16:59:34 2020
+-----+
| NVIDIA-SMI 440.33.01    Driver Version: 440.33.01    CUDA Version: 10.2 |
+-----+-----+-----+
| GPU  Name      Persistence-M | Bus-Id      Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap| Memory-Usage | GPU-Util  Compute M. |
+-----+-----+-----+
|  0  GeForce RTX 208...  On   | 00000000:19:00.0 Off |          N/A |
| 30%   32C   P8     18W / 250W |       1MiB / 11019MiB |     0%   Default |
+-----+-----+-----+
|  1  GeForce RTX 208...  On   | 00000000:1B:00.0 Off |          N/A |
|  0%   43C   P8     1W / 260W |       1MiB / 11019MiB |     0%   Default |
+-----+-----+-----+
|  2  GeForce RTX 208...  On   | 00000000:68:00.0 Off |          N/A |
| 31%   36C   P8     20W / 250W |       59MiB / 11016MiB |     0%   Default |
+-----+-----+-----+
+-----+
| Processes:                               GPU Memory |
| GPU  PID  Type  Process name             Usage     |
| -----+-----+-----+-----|
|  2    1631  G   /usr/lib/xorg/Xorg           39MiB |
|  2    1663  G   /usr/bin/gnome-shell        17MiB |
+-----+
```

```
[2020-12-14 09:31:41] GPU #1: Intensity set to 25, 33554432 cuda threads
[2020-12-14 09:31:41] GPU #2: Intensity set to 25, 33554432 cuda threads
[2020-12-14 09:31:41] GPU #0: Intensity set to 25, 33554432 cuda threads
[2020-12-14 09:31:42] GPU #1: Gigabyte RTX 2080 Ti, 1007.25 MH/s
[2020-12-14 09:31:42] GPU #0: Zotac RTX 2080 Ti, 981.97 MH/s
[2020-12-14 09:31:42] GPU #2: Zotac RTX 2080 Ti, 979.54 MH/s
[2020-12-14 09:31:46] GPU #1: Gigabyte RTX 2080 Ti, 2849.73 MH/s
[2020-12-14 09:31:46] GPU #2: Zotac RTX 2080 Ti, 2869.03 MH/s
[2020-12-14 09:31:46] GPU #0: Zotac RTX 2080 Ti, 2855.05 MH/s
[2020-12-14 09:31:50] GPU #2: Zotac RTX 2080 Ti, 2838.45 MH/s
[2020-12-14 09:31:50] GPU #1: Gigabyte RTX 2080 Ti, 2841.18 MH/s
[2020-12-14 09:31:50] GPU #0: Zotac RTX 2080 Ti, 2829.64 MH/s
[2020-12-14 09:31:54] GPU #1: Gigabyte RTX 2080 Ti, 2561.23 MH/s
[2020-12-14 09:31:54] GPU #0: Zotac RTX 2080 Ti, 2530.67 MH/s
[2020-12-14 09:31:54] GPU #2: Zotac RTX 2080 Ti, 2842.83 MH/s
```

4 Proof of Concept

4.4 Job injection based on set_extranonce

During job injection based on set_extranonce attack, the adversary needs to keep three TCP connections to miner, the normal pool and the malicious pool at the same time. All of the messages between them will be forwarded by adversary.

- In the initial stage of the attack, the adversary will save extranonce1 and extranonce2_size from malicious pool. And the adversary will forward the subscription and authorization message to the normal pool.
- Next, the adversary will forward jobs sent by the normal pool to miner and send *shares* submitted by miner to the normal pool.
- After 10 *shares* being submitted to the normal pool, the adversary constructs fake *set_extranonce* message with extranonce1 and extranonce2_size from malicious pool and send it to miner. And then the adversary inserts the latest difficulty and job of the malicious pool into the job flow, and the miner will complete the job of malicious pool.

4.4 Job injection based on set_extranonce

The right half of the image is the output of the attack script.

The left part is the output of the ccmminer.

The miner received the job message and difficulty negotiation message hijacked and forwarded by the adversary successfully.

```
https://microk8s.io/high-availability
* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
  https://ubuntu.com/livepatch

316 packages can be updated.
59 updates are security updates.

New release '20.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
[miner@miner ~]$ ./ccminer -o stratum+tcp://172.10.2.17:333 --userpass jack:3 -Ph
[miner@miner ~]$ ./ccminer -o stratum+tcp://172.10.2.17:333 --userpass jack:3 -Ph
-a sha256 -i 25
*** ccminer 2.3.1 for nVidia GPUs by tpruvot@github ***
  Built with the nVidia CUDA Toolkit 10.0 64-bits

Originally based on Christian Buchner and Christian H. project
Include some kernels from alexis78, djm34, djEzo, tsiv and krnlx.

BTC donation address: 1AJdfCpLWPNoAMDFHF1wD5y8VgKSSTHxPo (tpruvot)

[2020-12-17 17:48:17] Starting on stratum+tcp://172.16.20.16:3333
* Rebuilt URL to: http://172.16.20.16:3333/
* Trying 172.16.20.1...
* TCP_NODELAY set
* Connected to 172.16.20.1 (172.16.20.10) port 3333 (#0)
* Connection #0 to host: [REDACTED] left intact
[2020-12-17 17:48:17] > {"id": 1, "method": "mining.subscribe", "params": ["ccminer/2.3.1"]}
[2020-12-17 17:48:17] NVML GPU monitoring enabled.
[2020-12-17 17:48:17] 3 miner threads started, using 'sha256d' algorithm.
[2020-12-17 17:48:17] < {"id":1,"result":[{"mining.set_difficulty","01000006"}, {"mining.notify","01000006"}, {""01000006",8}, {"error":null}]
[2020-12-17 17:48:17] > {"id": 2, "method": "mining.authorize", "params": ["jack", "3"]}
[2020-12-17 17:48:17] < {"id":2,"result":true, "error":null}
[2020-12-17 17:48:17] > {"id": 3, "method": "mining.extranonce.subscribe", "params": []}
[2020-12-17 17:48:18] < {"id":null, "method": "mining.set_difficulty", "params": [16384]}
[2020-12-17 17:48:18] < {"id":null, "method": "mining.notify", "params": ["0", "7e13c2153453397973dc59e
lafc9b868aa42b100059a400000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
ffffffffff02becf7fb28000000017a9143edc44cdf9bbf7d5e722345d5bd576de5be72db587000000000000266a24a
a21a9edb8ac82c15cf4a2c84710c0f5efa8df563cd7ffbf14c6f5ce908b95e1f7302a0100000000", "51957eb9b1ec632
191fe9d27655764aee3c3b4cd54297e7765a7e2258e6fc", "83ea1a6b03422003df348522b343e1a9bc9ee74ef9217
a57fab385c850d0fb", "d140c425b156f112b02545a7f782d3d8c686ae0d94c4a4f2643a6495b18899e", "80d4fc7633
9b1c97611ecfc6a5333e3a1a2d7c6f45d494bdd8260171378126ac", "83047415d3be61b00af1b1fd3bcc780fdb5bbfdc
557c1a1c943a7d880bcc53e2", "6b9abc55eabddf4b5faddfb0e476e08d5a460cad12ce265d1a3425ac0d57c98", "32822
7d9137047eb61da81693a95b6857d8e939e62dfcaf400c77e2eb4c3b", "c7af0b0b3ad52a65f02d77d2355b67d9863b
1426151fd320d56e9c8d6948c782", "d54d3c3dea51f4fb281e479a2601f01b3fb8a0d0562c70722299f4d2ef2d843a", "2699fe9a21f41e412d4faabc33a5d429c7b17e5f7794caf2bcc29198ac518f7", "5e028ff9ff05a85f962ddfe825d5fb9
57c504ee914477fe05a98cb80dd94c291", "a4fc39c052ebd597e07cadeff18b2e43ec91dc45c1984c0c0b3e757e8c5b95
79"], "20000000", "170eb156", "5fd3ld5e", true]}
[2020-12-17 17:48:18] Stratum difficulty set to 16384
[2020-12-17 17:48:18] sha256d block 660872, diff 19157154724710.137
[2020-12-17 17:48:19] GPU #0: Intensity set to 25, 33554432 cuda threads
[2020-12-17 17:48:19] GPU #1: Intensity set to 25, 33554432 cuda threads
[2020-12-17 17:48:19] GPU #2: Intensity set to 25, 33554432 cuda threads
[2020-12-17 17:48:19] GPU #0: Zotac RTX 2080 Ti, 1208.28 MH/s
[2020-12-17 17:48:19] GPU #2: Zotac RTX 2080 Ti, 1141.56 MH/s
[2020-12-17 17:48:19] GPU #1: Gigabyte RTX 2080 Ti, 1139.11 MH/s
```

```
Last login: Thu Dec 17 17:14:19 on ttys003
[miner@miner ~]$ nc $ip $port
[miner@miner ~]$ 2666's password:
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-117-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 * Introducing self-healing high availability clusters in MicroK8s.
 Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

https://microk8s.io/high-availability

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
  https://ubuntu.com/livepatch

316 packages can be updated.
59 updates are security updates.

New release '20.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
[miner@miner ~]$ ./ccminer -o stratum+tcp://172.16.20.16:3333 --userpass jack:3 -Ph
[miner@miner ~]$ ./ccminer -o stratum+tcp://172.16.20.16:3333 --userpass jack:3 -Ph
-a sha256 -i 25
*** ccminer 2.3.1 for nVidia GPUs by tpruvot@github ***
  Built with the nVidia CUDA Toolkit 10.0 64-bits

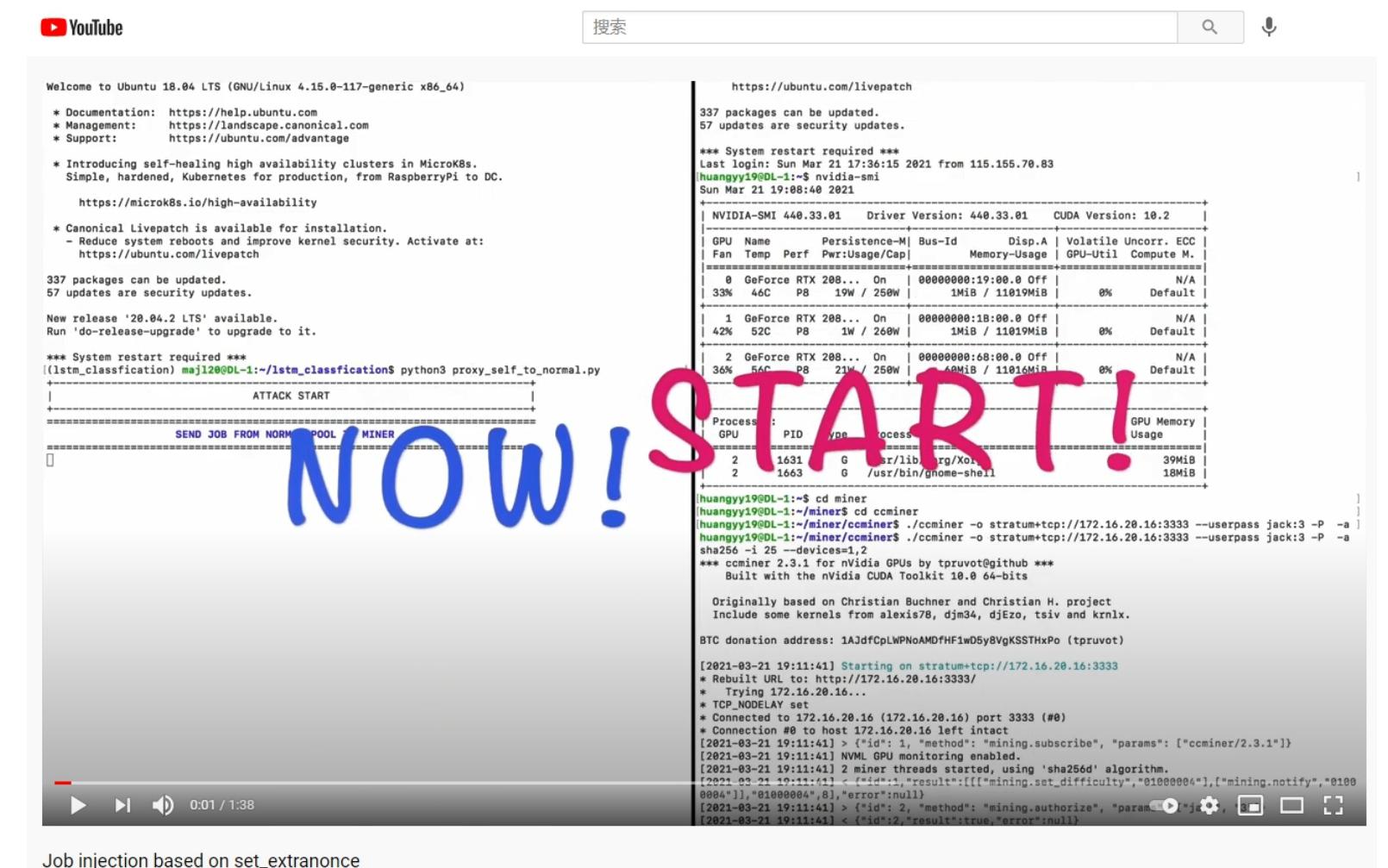
Originally based on Christian Buchner and Christian H. project
Include some kernels from alexis78, djm34, djEzo, tsiv and krnlx.

BTC donation address: 1AJdfCpLWPNoAMDFHF1wD5y8VgKSSTHxPo (tpruvot)

[2020-12-17 17:48:17] Starting on stratum+tcp://172.16.20.16:3333
* Rebuilt URL to: http://172.16.20.16:3333/
* Trying 172.16.20.1...
* TCP_NODELAY set
* Connected to 172.16.20.1 (172.16.20.10) port 3333 (#0)
* Connection #0 to host: [REDACTED] left intact
[2020-12-17 17:48:17] > {"id": 1, "method": "mining.subscribe", "params": ["ccminer/2.3.1"]}
[2020-12-17 17:48:17] NVML GPU monitoring enabled.
[2020-12-17 17:48:17] 3 miner threads started, using 'sha256d' algorithm.
[2020-12-17 17:48:17] < {"id":1,"result":[{"mining.set_difficulty","01000006"}, {"mining.notify","01000006"}, {""01000006",8}, {"error":null}]
[2020-12-17 17:48:17] > {"id": 2, "method": "mining.authorize", "params": ["jack", "3"]}
[2020-12-17 17:48:17] < {"id":2,"result":true, "error":null}
[2020-12-17 17:48:17] > {"id": 3, "method": "mining.extranonce.subscribe", "params": []}
[2020-12-17 17:48:18] < {"id":null, "method": "mining.set_difficulty", "params": [16384]}
[2020-12-17 17:48:18] < {"id":null, "method": "mining.notify", "params": ["0", "7e13c2153453397973dc59e
lafc9b868aa42b100059a4000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
ffffffffff02becf7fb28000000017a9143edc44cdf9bbf7d5e722345d5bd576de5be72db58700000000000266a24a
a21a9edb8ac82c15cf4a2c84710c0f5efa8df563cd7ffbf14c6f5ce908b95e1f7302a0100000000", "51957eb9b1ec632
191fe9d27655764aee3c3b4cd54297e7765a7e2258e6fc", "83ea1a6b03422003df348522b343e1a9bc9ee74ef9217
a57fab385c850d0fb", "d140c425b156f112b02545a7f782d3d8c686ae0d94c4a4f2643a6495b18899e", "80d4fc7633
9b1c97611ecfc6a5333e3a1a2d7c6f45d494bdd8260171378126ac", "83047415d3be61b00af1b1fd3bcc780fdb5bbfdc
557c1a1c943a7d880bcc53e2", "6b9abc55eabddf4b5faddfb0e476e08d5a460cad12ce265d1a3425ac0d57c98", "32822
7d9137047eb61da81693a95b6857d8e939e62dfcaf400c77e2eb4c3b", "c7af0b0b3ad52a65f02d77d2355b67d9863b
1426151fd320d56e9c8d6948c782", "d54d3c3dea51f4fb281e479a2601f01b3fb8a0d0562c70722299f4d2ef2d843a", "2699fe9a21f41e412d4faabc33a5d429c7b17e5f7794caf2bcc29198ac518f7", "5e028ff9ff05a85f962ddfe825d5fb9
57c504ee914477fe05a98cb80dd94c291", "a4fc39c052ebd597e07cadeff18b2e43ec91dc45c1984c0c0b3e757e8c5b95
79"], "20000000", "170eb156", "5fd3ld5e", true]}
[2020-12-17 17:48:18] ATTACK START
[2020-12-17 17:48:18] SEND JOB FROM NORMAL_POOL TO MINER
[2020-12-17 17:48:18]
```


4.4 Job injection based on set_extranonce

<https://www.youtube.com/watch?v=ZvpdOj6U0vM>



4 Proof of Concept

4.5 Time segment

During the time segment attack, the adversary needs to switch the connection to different pools at a fixed time segment. We set the time segment to 10 minutes.

- In the first 10 minutes, the adversary will forward the subscription and authorization message to the normal pool. The miner will only work for the normal pool in the remaining time. At the end of the ten minutes, the adversary will disconnect from the normal pool.
- In the next 10 minutes, the miner will reconnect to the pool, and the adversary will hijack the TCP connection to the malicious pool. The miner will only work for the malicious pool in the remaining time until the time segment ends.

4.5 Time segment

The adversary successfully forwarded the *shares* submitted by the miner to the normal mining pool.

```
c8c10e242b998c4ca8d766f58b023dd5caf354861fd2621779436c", "e2ac2151c29a37c71f88170cb7d03ca6d8446748aa
37cb5a81e27e9d39ad4988", "5cb81f104123c3fa1b73d69c18886678f6b310c5f9c8806242c25dd9166a5f37"], "200000
00", "170f1372", "5fdc73dc", false]
[2020-12-20 18:11:57] GPU #0: Zotac RTX 2080 Ti, 2429.15 MH/s
[2020-12-20 18:11:57] GPU #1: Gigabyte RTX 2080 Ti, 2497.16 MH/s
[2020-12-20 18:11:59] GPU #2: Zotac RTX 2080 Ti, 1007.52 MH/s
[2020-12-20 18:12:01] GPU #0: Zotac RTX 2080 Ti, 2419.58 MH/s
[2020-12-20 18:12:01] GPU #1: Gigabyte RTX 2080 Ti, 2510.13 MH/s
[2020-12-20 18:12:03] GPU #2: Zotac RTX 2080 Ti, 1006.68 MH/s
[2020-12-20 18:12:04] > {"method": "mining.submit", "params": [{"jack", "13", "20000000000000000000", "5
fdc73dc", "616dc14d"}, "id":10}
[2020-12-20 18:12:04] < {"id":10,"result":true,"error":null}
[2020-12-20 18:12:04] accepted: 4/4 (diff 64.538), 5921.71 MH/s yes!
[2020-12-20 18:12:05] GPU #1: Gigabyte RTX 2080 Ti, 2525.02 MH/s
[2020-12-20 18:12:05] GPU #0: Zotac RTX 2080 Ti, 2425.88 MH/s
[2020-12-20 18:12:06] > {"method": "mining.submit", "params": [{"jack", "13", "26000000000000000000", "5
fdc73dc", "f9a3d011"}, "id":11}
[2020-12-20 18:12:06] < {"id":11,"result":true,"error":null}
[2020-12-20 18:12:06] accepted: 5/5 (diff 108.894), 5922.53 MH/s yes!
[2020-12-20 18:12:08] GPU #2: Zotac RTX 2080 Ti, 1009.53 MH/s
[2020-12-20 18:12:09] GPU #1: Gigabyte RTX 2080 Ti, 2500.26 MH/s
[2020-12-20 18:12:09] GPU #0: Zotac RTX 2080 Ti, 2415.32 MH/s
[2020-12-20 18:12:12] GPU #2: Zotac RTX 2080 Ti, 1008.21 MH/s
[2020-12-20 18:12:13] GPU #0: Zotac RTX 2080 Ti, 2426.38 MH/s
[2020-12-20 18:12:13] GPU #1: Gigabyte RTX 2080 Ti, 2493.01 MH/s
[2020-12-20 18:12:16] GPU #2: Zotac RTX 2080 Ti, 1006.54 MH/s
[2020-12-20 18:12:17] GPU #0: Zotac RTX 2080 Ti, 2415.21 MH/s
[2020-12-20 18:12:17] GPU #1: Gigabyte RTX 2080 Ti, 2525.26 MH/s
[2020-12-20 18:12:20] > {"method": "mining.submit", "params": [{"jack", "13", "62000000000000000000", "5
fdc73dc", "91a3d058"}, "id":12}
[2020-12-20 18:12:20] < {"id":12,"result":true,"error":null}
[2020-12-20 18:12:20] accepted: 6/6 (diff 71.134), 5934.60 MH/s yes!
[2020-12-20 18:12:20] GPU #2: Zotac RTX 2080 Ti, 1006.19 MH/s
[2020-12-20 18:12:21] GPU #1: Gigabyte RTX 2080 Ti, 2522.46 MH/s
[2020-12-20 18:12:21] GPU #0: Zotac RTX 2080 Ti, 2423.98 MH/s
[2020-12-20 18:12:25] GPU #2: Zotac RTX 2080 Ti, 1007.23 MH/s
[2020-12-20 18:12:25] GPU #1: Gigabyte RTX 2080 Ti, 2499.49 MH/s
[2020-12-20 18:12:25] GPU #0: Zotac RTX 2080 Ti, 2428.29 MH/s
[2020-12-20 18:12:26] < {"id":null,"method": "mining.submit", "params": [{"14", "c3625bf105f1a760dc4179c
b9394ac0b2d75875f0006ee1f00000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000fffffff2d036d190a048623df5f726567696f6e312f50726f6a6563742042543506f6c2f", "f
fffffff02d5324d2e0000000017a9143edc44cdf9bbf7d5e722345d5bd576de5be72db58700000000000000000000000000000000000000
a9ed27f4a00f9453d307b9e655a52895c58937641dce59e8040d4102bd000000000000000000000000000000000000000000000000000
f420cefef99118840b397655afbf7b0d3f5a4fbdc0a2ab", "2b0a0e3e8bf5b9adbd3cccf2b0446cc848858f7a729eb27a3a7d
866b2262a1e9f", "a97057f5f09f84c5c49c8500a37fd348191df57b54f095b9c4ac9e851ae14727", "aa1d803966b3214f
23451a2bf9ebc7841915528057832117b", "da1c257dbc31fce3c87956b10505f9fd2f264a45f7d7595
742d45d4ef14c6e5", "918fbfdfb840ae2e0e3a0c0e0c7c10e244f8632fe9ef9e005399048d1402ceec0", "c3e87179a655a
00b98940d5c6e0edd7957c476c027d9aa0715522f747b7b5724", "bdbda7bfa02b635e5e6392e2d01f8fd23fec75e7066824
2da388a4a072213ff69", "cal1acd4cbd824f81d1f656a8fce864f70131f9111bbd6fe0a029665c9814d6", "ab716be1b3
c8c10e242b998c4ca8d766f58b023dd5caf354861fd2621779436c", "e2ac2151c29a37c71f88170cb7d03ca6d8446748aa
37cb5a81e27e9d39ad4988", "5cb81f104123c3fa1b73d69c18886678f6b310c5f9c8806242c25dd9166a5f37"], "200000
00", "170f1372", "5fdc73dc", false}]
[2020-12-20 18:12:29] GPU #0: Zotac RTX 2080 Ti, 2409.62 MH/s
[2020-12-20 18:12:29] GPU #1: Gigabyte RTX 2080 Ti, 2502.63 MH/s
[2020-12-20 18:12:29] GPU #2: Zotac RTX 2080 Ti, 1003.10 MH/s
[2020-12-20 18:12:33] GPU #0: Zotac RTX 2080 Ti, 2429.10 MH/s
[2020-12-20 18:12:33] GPU #1: Gigabyte RTX 2080 Ti, 2497.83 MH/s
[2020-12-20 18:12:33] GPU #2: Zotac RTX 2080 Ti, 1004.05 MH/s
[2020-12-20 18:12:37] GPU #1: Gigabyte RTX 2080 Ti, 2502.19 MH/s
[2020-12-20 18:12:37] GPU #0: Zotac RTX 2080 Ti, 2422.44 MH/s
[2020-12-20 18:12:38] GPU #2: Zotac RTX 2080 Ti, 1007.34 MH/s
[2020-12-20 18:12:41] GPU #1: Gigabyte RTX 2080 Ti, 2526.08 MH/s
[2020-12-20 18:12:41] GPU #0: Zotac RTX 2080 Ti, 2416.28 MH/s
```

```
Last login: Sun Dec 20 18:04:57 on ttys001
[shenqijiadexiaolilideMacBook-Pro:~ nct$ ssh majl20@172.16.20.16
majl20@172.16.20.16's password:
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-117-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

https://microk8s.io/high-availability

* Canonical Livepatch is available for installation.
  - Reduce system reboots and improve kernel security. Activate at:
    https://ubuntu.com/livepatch

325 packages can be updated.
59 updates are security updates.

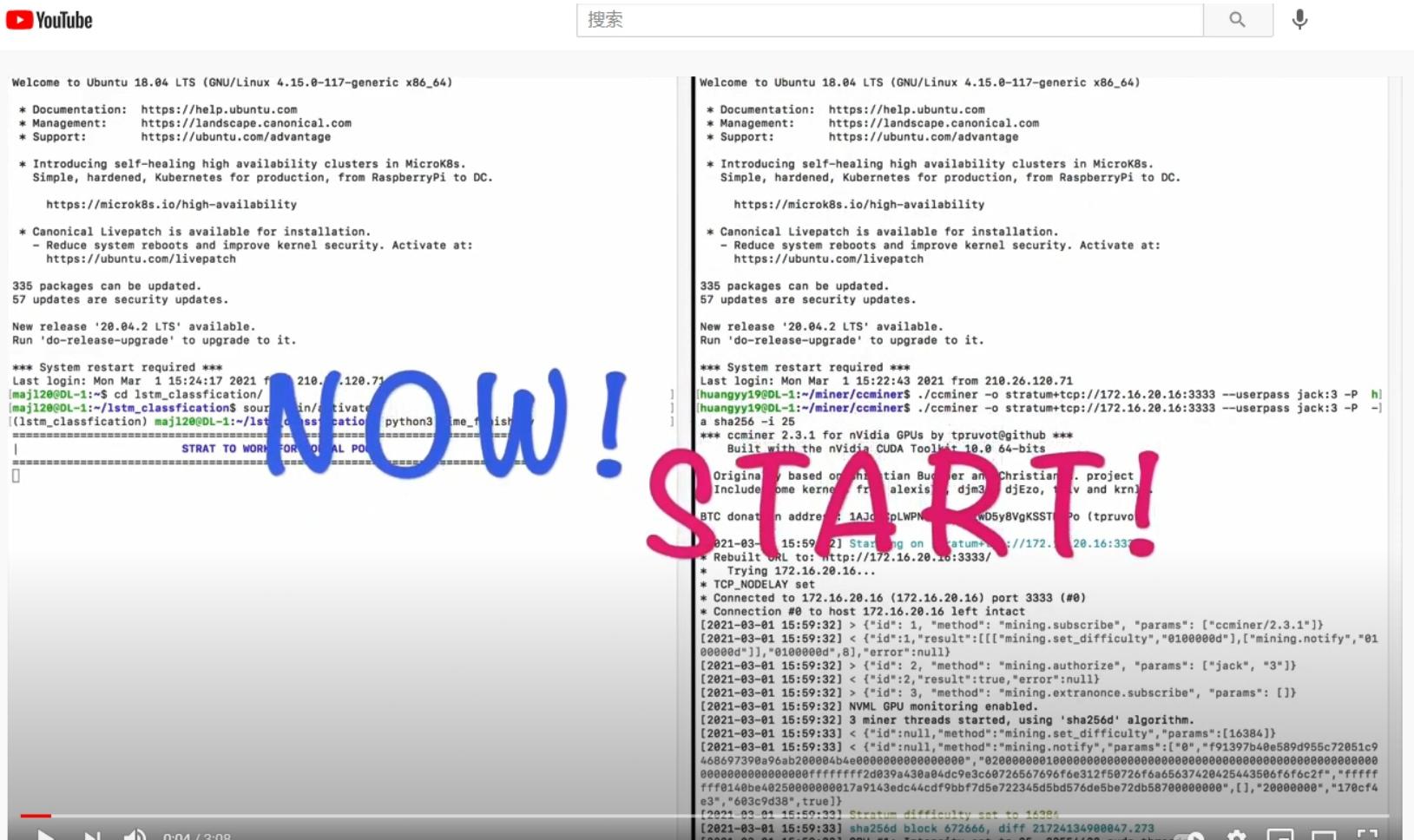
New release '20.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Sun Dec 20 15:56:31 2020 from 115.155.66.142
[majl20@DL-1:~$ cd lstm_classification/
[majl20@DL-1:~/lstm_classification$ source bin/activate
([lstm_classification] majl20@DL-1:~/lstm_classification$ python3 time_finish.py
=====
| STRAT TO WORK FOR NORMAL POOL | =====
+-----+
| | TEN MINUTES PASSED |
+-----+
=====

| STRAT TO WORK FOR MALICIOUS POOL | =====
```


4.5 Time segment

<https://www.youtube.com/watch?v=OIS4TRMgAJs>



Time segment

#BHASIA @BLACKHATEVENTS

4 Proof of Concept

4.6 Summary

Job injection based on set_extranonce attack model has better concealment, because it inserts a small amount of malicious pool job to miner in the job flow at a low frequency, making it difficult for the pool administrator to detect.

The second attack model is to switch the connection between the normal pool and the malicious pool in a fixed time segment. Therefore, the mining pool administrator may observe the fluctuation of the computing power.

Both of our proposed attack schemes can achieve the purpose of stealing hashrate for malicious mining pool.

4 Proof of Concept

4.6 Summary



The screenshot shows a blog post from MarkMonitor. At the top, there's a navigation bar with the Clarivate logo, the MarkMonitor™ logo, and links for 'Why Domain Management' and 'Products'. Below the navigation is a large block of redacted JavaScript code. Underneath the code, the post title is displayed in large, bold, black font: 'China cybersecurity update: DNS hijacking and IoT crimes'. Below the title is a horizontal line separating the header from the main content. To the left of this line is the author's name, 'BRIAN KING', followed by their title, 'Director of Internet Policy and Industry Affairs' and the company, 'Clarivate'. To the right of the line is the publication date, 'JANUARY 25, 2019', and the estimated reading time, '3 MINUTE READ'. At the bottom of the post, there's a section titled 'Share this article' with icons for LinkedIn, Facebook, and Twitter.

China cybersecurity update: DNS hijacking and IoT crimes

BRIAN KING
Director of Internet Policy and Industry Affairs
Clarivate

Share this article

LinkedIn  Facebook  Twitter 

Many ISPs are using traffic hijacking for illegal profit-making activities such as pop-up advertisements.

If they change the target of hijacking to Stratum, it will greatly harm the interests of miners.

It's dangerous to blockchain community.



Thanks