

Identifying Qakbot Servers with Regex and TLS Certificates

By Matthew

Published: 2023-11-30 · Archived: 2026-04-05 14:01:53 UTC

In this post we will leverage regular expressions and TLS certificates to capture 83 dispersed Qakbot servers.

These servers are well made and there are minimal traditional patterns (ports, service names, ASN's) that can be used for signaturing. Instead we will focus on commonalities within the `subject_dn` and `issuer_dn` fields to identify servers.

This is a relatively advanced technique that will require a basic understanding of regular expressions, and also a paid/researcher license for Censys.

The final query is shown below. [A link can be found here.](#)







```
services.tls.certificates.leaf_data.subject_dn=/C=\w\w, OU=[a-zA-Z0-9 \.]+, CN=[a-z]+\.[a-z]+/ and services.tls
```

Note that this post is primarily a demonstration of technical concepts that can be used to identify malware. We have not 100% validated that all results are Qakbot (although most appear to be), and we are relying on the initial ThreatFox tag being accurate.

Initial Server From ThreatFox

The initial server IP of `74.12.147[.]243:2222` was obtained from ThreatFox. Initially shared by the Twitter user [@drb_ra](#).

Database Entry

IOC ID:	1207439
IOC:	 74.12.147.243:2222
IOC Type 	ip:port
Threat Type 	botnet_cc
Malware:	 QakBot
Malware alias:	Oakboat, Pinkslipbot, Qbot, Quakbot
Confidence Level 	 Confidence level is moderate (50%)
First seen:	2023-11-30 06:54:15 UTC

By searching the IP address on Censys, we can quickly identify a suspicious certificate running on the reported 2222 port.

This certificate contains seemingly random text. With long values and only alphabetical characters.

UNKNOWN 2222/TCP

11/29/2023 19:45 UTC

Details

[VIEW ALL DATA](#)

TLS

Handshake

Version Selected TLSv1_2

Cipher Selected TLS_RSA_WITH_AES_128_GCM_SHA256

Certificate

Fingerprint 7c39dab10cef9856318f1c3e83943b4da91140fd28bb8d2f42eb33efe4aca7f4

Subject C=US, OU=Vzbxanrbu Eivhtmjabe Qjihwitl, CN=motnooz.biz

Issuer C=US, ST=VA, L=Oodrsu, O=Eiuiip Vilo Ubatoea LLC., CN=motnooz.biz

Names motnooz.biz

Fingerprint

JARM 04d02d00004d04d04c04d02d04d04d9674c6b4e623ae36cc2d998e99e2262e

JA3S ccd5709d4a9027ec272e98b9924c36f7

Using "View All Data", we can gather more information about the service running on 2222 .

This reveals an empty service banner that can be later used as a pivot point or as a field to narrow down search results.

2222/UNKNOWN TCP View Definition

Attribute	Value	
services.banner		<input type="button" value="q"/>
services.banner_hashes	sha256:e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855	<input type="button" value="q"/>
services.certificate	7c39dab10cef9856318f1c3e83943b4da91140fd28bb8d2f42eb33efe4aca7f4	<input type="button" value="q"/>
services.discovery_method	IPV4_WALK_FULL_PRIORITY_1	<input type="button" value="q"/>
services.extended_service_name	UNKNOWN	<input type="button" value="q"/>
services.tls.certificate	7c39dab10cef9856318f1c3e83943b4da91140fd28bb8d2f42eb33efe4aca7f4	<input type="button" value="q"/>

The exact structure of the TLS Certificate can be established with this view.

services.tls.certificates.leaf_fp_sha_256	7c39dab10cef9856318f1c3e83943b4da91140fd28bb8d2f42eb33efe4aca7f4	<input type="button" value="q"/>
services.tls.certificates.leaf_data.names	motnooz.biz	<input type="button" value="q"/>
services.tls.certificates.leaf_data.subject_dn	C=US, OU=Vzbxanrbu Eivhtmjabi Qjihwitl, CN=motnooz.biz	<input type="button" value="q"/>
services.tls.certificates.leaf_data.issuer_dn	C=US, ST=VA, L=Oodrsu, O=Eiuij Vilo Ubatoea LLC., CN=motnooz.biz	<input type="button" value="q"/>
services.tls.certificates.leaf_data.pubkey_bit_size	2048	<input type="button" value="q"/>
services.tls.certificates.leaf_data.pubkey_algorithm	RSA	<input type="button" value="q"/>

The search box next to `services.tls.certificates.leaf_data.subject_dn` can be used to pre-build an exact query.

We will use this pre-built query as a base for our regular expression.

🔍 Hosts ▾ ⚙️ ✖ 🚀 >_ Search

```
services.tls.certificates.leaf_data.subject_dn="C=US, OU=Vzbxanrbu Eivhtmjabi Qjihwitl, CN=motnooz.biz"
```

Hosts
Results: 1 Time: 0.23s

📍 **70.27.15.38 (bras-base-ckvlon0127w-grc-13-70-27-15-38.dsl.bell.ca)**

🌐 BACOM (577) 📍 Ontario, Canada

1 Matched Service

⚙️ 2222/UNKNOWN

2 Other Services

🌐 50001/HTTP 🌐 58603/HTTP

< PREVIOUS NEXT >

How To Convert Values Into Regular Expressions

We can go ahead and modify the search parameter to a regular expression.

A summary of the changes can be found below.

- `C=US` -> `C=\w\w` - We will let the `C` field matches on any two characters
- `OU=Vzbxanrbu Eivhtmjabi Qjihwitl` -> `OU=[a-zA-Z0-9]+` - We can let the `OU` field matches any sequence of alphabetical characters, allowing for a space in between.
- `CN=motnooz.biz` -> `CN=[a-z]+\.[a-z]+` - we will let the `CN` field match on any domain containing only lowercase letters.

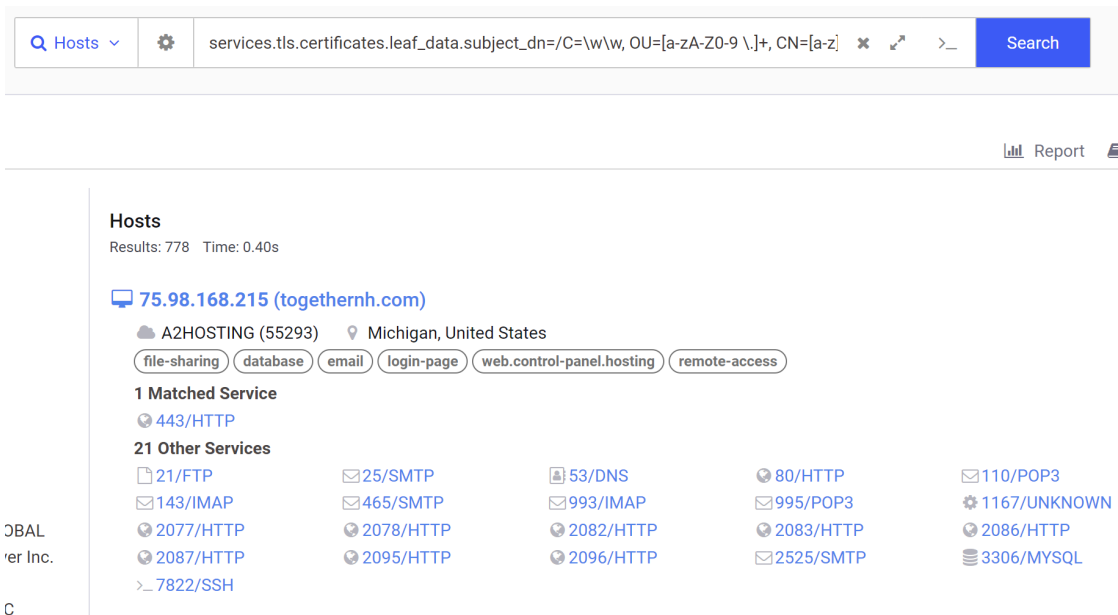
After modifying the query as above, we can also add a filter for our original IP. This ensures that the same IP is matched and hasn't been lost. This is a means of quickly verifying that a regex works as intended.

We can see below that the same Initial IP is matched, meaning that the regex probably works.



With the Regex validated, We can now go ahead and remove the IP Address, leaving only the `subject_dn` field.

This modified search results in 778 servers, many of which don't completely follow the certificate structure we want.



Validating Search Results

If we inspect the first returned result of `75.98.168[.]215`, we can see that the `subject_dn` matches our regular expression structure, but the `issuer_dn` is different to our initial Qakbot.

Below is the first returned result (Which does not match our pattern). Note that it contains the `-` character in the `CN` and `O` fields.

TLS

Handshake

Version Selected TLSv1_3

Cipher Selected TLS_AES_256_GCM_SHA384

Certificate

Fingerprint 733ac217ff0430a3e98c6fd3736dc95f0b25c5b3bfdee7d3ead7ce829777183d

Subject C=US, OU=Domain Control Validated by OneClickSSL, CN=alliantmetals.com

Issuer C=BE, O=GlobalSign nv-sa, CN=AlphaSSL CA - SHA256 - G2

Names alliantmetals.com, www.alliantmetals.com

Fingerprint

Below is the original Qakbot C2. Note the lack of special characters and numerical values.

UNKNOWN 2222/TCP

11/29/2023 19:45 UTC

Details

[VIEW ALL DATA](#)

TLS

Handshake

Version Selected TLSv1_2

Cipher Selected TLS_RSA_WITH_AES_128_GCM_SHA256

Certificate

Fingerprint 7c39dab10cef9856318f1c3e83943b4da91140fd28bb8d2f42eb33efe4aca7f4

Subject C=US, OU=Vzbxanrbu Eivhtmjabe Qjihwitl, CN=motnooz.biz

Issuer C=US, ST=VA, L=Oodrsu, O=Euiup Vilo Ubatoea LLC., CN=motnooz.biz

Names motnooz.biz

Fingerprint

JARM 04d02d00004d04d04c04d02d04d04d9674c6b4e623ae36cc2d998e99e2262e

JA3S ccd5709d4a9027ec272e98b9924c36f7

The initial search returns results that match our `subject_dn` regular expression.

But there are results with a completely different (and not matching) structure on the `issuer_dn`.

We can go back to our initial Qakbot C2 and follow the same process as before to build a regular expression on the `issuer_dn` field.

We can then validate the regular expression by including the initial IP address.

The screenshot shows a search interface with a search bar containing the query: `services.tls.certificates.leaf_data.issuer_dn=/C=\w\w, ST=\w\w, L=[a-zA-Z]+, O=[a-zA-Z0-9\ \.]+, CN=[a-z]+\.[a-z]+/ and ip:70.27.15.38`. Below the search bar, the results section is titled "Hosts" and shows "Results: 1 Time: 0.38s". The single result is `70.27.15.38 (bras-base-ckvlon0127w-grc-13-70-27-15-38.dsl.bell.ca)`. It lists "BACOM (577)" in "Ontario, Canada", "1 Matched Service" (2222/UNKNOWN), and "2 Other Services" (50001/HTTP and 58603/HTTP). Navigation buttons for "PREVIOUS" and "NEXT" are visible at the bottom.

Since the `issuer_dn` field has not been validated; We can now go ahead and add the `issuer_dn` query to the initial `subject_dn` search. We can also include the initial IP for validation.

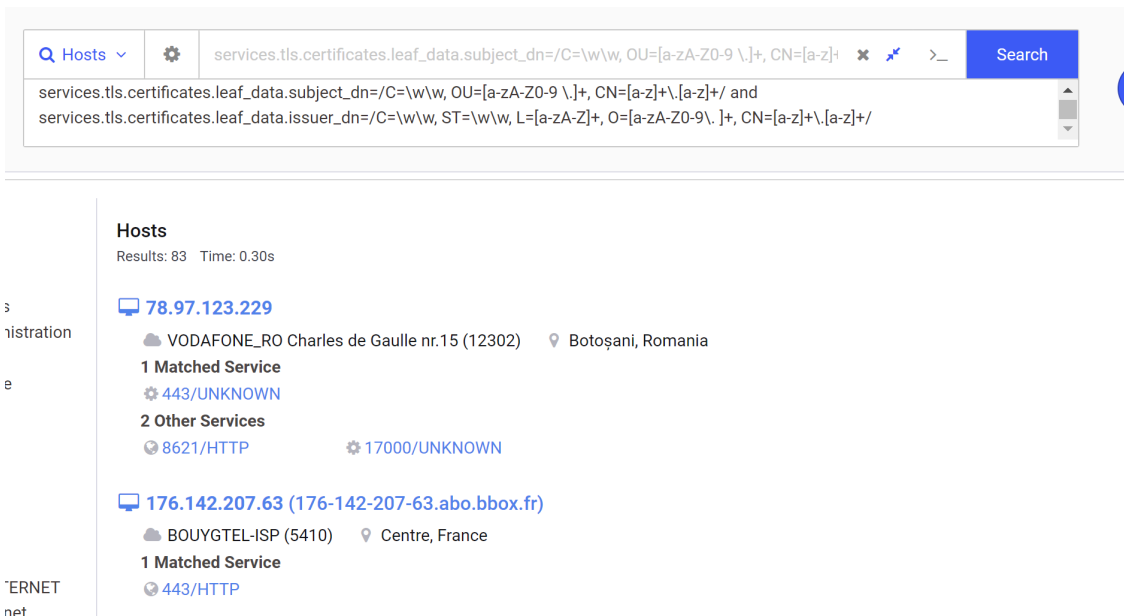
At this point, we have a total query of

```
services.tls.certificates.leaf_data.subject_dn=/C=\w\w, OU=[a-zA-Z0-9 \.]+, CN=[a-z]+\.[a-z]+/ and services.tls
```

The screenshot shows a search interface with a search bar containing the query: `services.tls.certificates.leaf_data.subject_dn=/C=\w\w, OU=[a-zA-Z0-9 \.]+, CN=[a-z]+\.[a-z]+/ and services.tls.certificates.leaf_data.issuer_dn=/C=\w\w, ST=\w\w, L=[a-zA-Z]+, O=[a-zA-Z0-9\ \.]+, CN=[a-z]+\.[a-z]+/ and ip:70.27.15.38`. Below the search bar, the results section is titled "Hosts" and shows "Results: 1 Time: 0.23s". The single result is `70.27.15.38 (bras-base-ckvlon0127w-grc-13-70-27-15-38.dsl.bell.ca)`. It lists "BACOM (577)" in "Ontario, Canada", "1 Matched Service" (2222/UNKNOWN), and "2 Other Services" (50001/HTTP and 58603/HTTP). Navigation buttons for "PREVIOUS" and "NEXT" are visible at the bottom.

The above search confirms that we haven't lost our initial hit, meaning the regex is valid, and the initial IP can be removed.

By removing the Initial IP Address and including only the `subject_dn` and `issuer_dn`, we're now down to a manageable number of 83 results.



Inspecting the first two hits, we can confirm that we have matches on our intended certificate structure.

Certificate

Fingerprint 3a85588536c4f52edb80668f21d982f13fea27d43db047cf3cd47cb8fc506f69

Subject C=AU, OU=Qooued Pkaiw, CN=opzlwf.net

Issuer C=AU, ST=BZ, L=Terioxpfo, O=Xljms Ikuwria Yewzn Inc., CN=opzlwf.net

Names opzlwf.net

Fingerprint

JA3S ccd5709d4a9027ec272e98b9924c36f7

Certificate

Fingerprint 6344a38116a194c8fdab06b9c9d8d915838ee89c136fca262ec9012b0d11605d

Subject C=ES, OU=Owasjeidy Twra Eqhidexoh, CN=ywzxs.biz

Issuer C=ES, ST=LA, L=Jxtvxotc, O=Qwx Itao Inc., CN=ywzxs.biz

Names ywzxs.biz

Fingerprint

Further Validation With Report Building

To save time validating every result individually, we use the "build report" function of Censys to hone in on the `subject_dn` or `issuer_dn` fields.

Report on Hosts

This tool allows you to generate a report on the breakdown of a value present on the Hosts returned by your query. For example, to generate a report on ports seen on Hosts with HTTP services, you could query for `services.service_name: HTTP` and then generate a report on the breakdown of the field `services.port`

Breakdown Field	Number of Buckets	BUILD REPORT
<code>services.tls.certificates.leaf_data.subject_dn</code>	500	

Report for Hosts

This confirms that most of the returned servers match our intended structure.

<code>C=AT, OU=Dgeou, CN=euei.com</code>	1	0.5%
<code>C=AT, OU=Etvub Aodozne Qekab, CN=iene.info</code>	1	0.5%
<code>C=AT, OU=leia, CN=ctxehfdug.net</code>	1	0.5%
<code>C=AT, OU=lpucsti Adsntqtrxp lwqjefo, CN=miyaiwvo.biz</code>	1	0.5%
<code>C=AT, OU=Oxzm Wjwyrtnxoko, CN=utip.biz</code>	1	0.5%
<code>C=AT, OU=Uiexn Wekfbig Umxqaltyot, CN=jaonoi.org</code>	1	0.5%
<code>C=AT, OU=Ukhxtaurzt, CN=txefeta.info</code>	1	0.5%
<code>C=AU, OU=Epclnutaj, CN=oxouy.mobi</code>	1	0.5%
<code>C=AU, OU=Gwoictrevdn, CN=ouxtetbfn.biz</code>	1	0.5%
<code>C=AU, OU=Hkcaxm, CN=ghoatksiwo.net</code>	1	0.5%
<code>C=AU, OU=Itllq, CN=fwoht.org</code>	1	0.5%
<code>C=AU, OU=ixftewbm Ftilepnsa Xtogahm, CN=tqouhdk.mobi</code>	1	0.5%
<code>C=AU, OU=Qoaisr Mbdogjoto Pibj, CN=oamt.mobi</code>	1	0.5%
<code>C=AU, OU=Qooued Pkaiw, CN=opzlvf.net</code>	1	0.5%
<code>C=AU, OU=Wxijqeh, CN=pidewaaetbu.us</code>	1	0.5%
<code>C=CA, OU=Aeha Etee, CN=aihpe.mobi</code>	1	0.5%
<code>C=CA, OU=Etiotjsaowj Eivluokt, CN=ecfvmw.biz</code>	1	0.5%
<code>C=CA, OU=Haltmc Cirh, CN=eivubtno.com</code>	1	0.5%
<code>C=CA, OU=iblsxu Eopocsao, CN=vlquotrgamd.info</code>	1	0.5%
<code>C=CA, OU=lkoaq, CN=oialk.com</code>	1	0.5%
<code>C=CA, OU=llhaenh, CN=ihaknpq.us</code>	1	0.5%

Lots of Qakbot-like Certificates

Honing in on Domain/Host Names

We can also use the "build report" function to hone in on `common_name` fields used in the TLS certificates.

Report on Hosts

This tool allows you to generate a report on the breakdown of a value present on the Hosts returned by your query. For example, to generate a report on ports seen on Hosts with HTTP services, you could query for `services.service_name: HTTP` and then generate a report on the breakdown of the field `services.port`

Breakdown Field	Number of Buckets	BUILD REPORT
<code>services.tls.certificate.parsed.subject.common_name</code>	500	

Report for Hosts

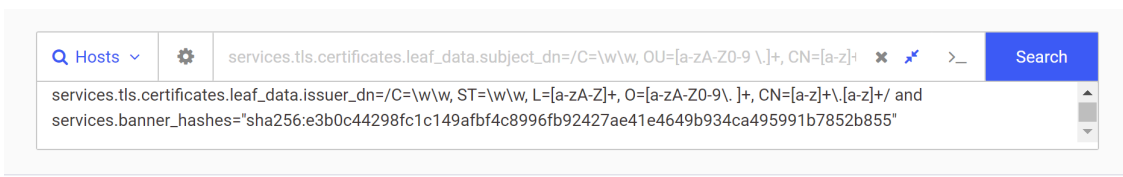
ecfvmw.biz
eeetnu.us
eehpeplhr.us
efpohwf.net
eivubtno.com
euei.com
euydxykaie.org
example.com
fbiafq.info
fwoht.org
ghoaetksiwo.net
gimcyeeoof.org
haeioeee.info
hxin.biz
hzolorcprw.us
iaiea.org
ieaorbuq.net
iene.info
ihaknpq.us
inqoqob.biz
inju.us

Using Build Report to List Domain Names of Returned Servers.

Query Refinement

There are potentially some false positives within the 83 returned results, so if we like, we can go ahead and add the empty banner hash from the initial IP.

This will reduce the hits down to 49. But it's possible that this may remove some malicious results. I did not validate this as it's very time-consuming, and the majority of servers seem to be malicious either way.



Hosts
Results: 49 Time: 0.63s

190.134.148.34
Administration Nacional de Telecomunicaciones (6057) Monteideo Department, Uruguay
1 Matched Service
995/UNKNOWN

149.75.147.46
Microsoft Windows RCN-AS (6079) Illinois, United States
1 Matched Service
443/UNKNOWN

Validating Results With Virustotal

Performing a quick search on some of the returned hits on Virustotal.

Most seem related to Qakbot, although we have not confirmed this 100%.

The image displays two screenshots of the VirusTotal interface for IP address analysis. Both screenshots show a 'Community Score' gauge, a '9 security vendors flagged this IP address as malicious' warning, and a 'Crowdsourced context' section with a warning icon and text: 'Activity related to QAKBOT - according to source Cluster25 - 1 day ago' (top) and 'Activity related to QAKBOT - according to source Cluster25 - 1 month ago' (bottom). The top screenshot is for IP 2.50.137.133 (2.50.128.0/17) with a community score of 9/88, AS 5384 (Emirates Telecommunications Group Company (Etisalat Group) Pjsc), and a last analysis date of 1 hour ago. The bottom screenshot is for IP 24.187.255.114 (24.184.0.0/13) with a community score of 5/88, AS 6128 (CABLE-NET-1), and a last analysis date of 1 month ago. Both pages include a 'Join the VT Community' banner and navigation tabs for 'DETECTION', 'DETAILS', 'RELATIONS', and 'COMMUNITY'.

8 security vendors flagged this IP address as malicious

Similar Graph API

217.165.233.123 (217.164.0.0/15)
AS 5384 (Emirates Telecommunications Group Company (Etisalat Group) Pjsc) | AE | Last Analysis Date 8 days ago

Community Score

DETECTION DETAILS RELATIONS COMMUNITY 3

Security vendors' analysis Do you want to automate checks?

AlphaSOC	Malware	BitDefender	Malware
Cluster25	Malicious	CRDF	Malicious
CyRadar	Malicious	Fortinet	Malware
G-Data	Malware	Lionic	Malicious

Conclusion

We now have a [functioning query](#) that captures 83 servers. We have not had to rely on port numbers, port ranges, or ASN locations to hone in.

Here is another copy of our final query.

```
services.tls.certificates.leaf_data.subject_dn=/C=\w\w, OU=[a-zA-Z0-9 \.]+, CN=[a-z]+\.[a-z]+/ and services.tls
```

If we observe the returned results below, we can see that the ASNs and port numbers vary greatly between the results, meaning that many traditional query styles will not work.

Hosts
Results: 83 Time: 0.30s

78.97.123.229
VODAFONE_RO Charles de Gaulle nr.15 (12302) Botoșani, Romania
1 Matched Service
443/UNKNOWN
2 Other Services
8621/HTTP 17000/UNKNOWN

176.142.207.63 (176-142-207-63.abo.bbox.fr)
BOUYGTEL-ISP (5410) Centre, France
1 Matched Service
443/HTTP

2.50.137.133 (bba-2-50-137-133.alshamil.net.ae)
Microsoft Windows EMIRATES-INTERNET Emirates Internet (5384) Abu Dhabi, United Arab Emirates
email jquery file-sharing network.device remote-access network.device.vpn
1 Matched Service
995/UNKNOWN
9 Other Services
> 22/SSH 110/POP3 143/IMAP 443/HTTP 444/HTTP
500/IKE 554/RTSP 1194/OPENVPN 1194/OPENVPN

96.248.1.183

Catching Qakbot without relying on ASN Numbers.

Related Content

If you found this content useful, check out other related posts in the free [Threat Intelligence](#) Section.

- [Advanced Regex Queries - BianLian](#)
- [Practical Queries for Identifying Malware - Part 3](#)
- [Combining Simple Pivot Points to Identify Infrastructure](#)

Sign up for Embee Research

Malware Analysis and Threat Intelligence Research

No spam. Unsubscribe anytime.

Complete List of Qakbot Infrastructure

IP Addresses

- 2[.]50[.]137[.]133
- 23[.]93[.]65[.]180
- 24[.]187[.]255[.]114
- 24[.]187[.]255[.]116
- 24[.]187[.]255[.]117
- 24[.]255[.]174[.]187
- 31[.]117[.]63[.]201
- 35[.]134[.]202[.]121

37[.]210[.]162[.]30
39[.]40[.]144[.]179
41[.]38[.]97[.]237
41[.]99[.]46[.]66
45[.]65[.]51[.]130
46[.]251[.]130[.]164
47[.]16[.]64[.]215
47[.]149[.]234[.]6
50[.]99[.]8[.]5
60[.]48[.]77[.]48
64[.]46[.]22[.]26
64[.]229[.]117[.]137
67[.]60[.]147[.]240
68[.]160[.]236[.]23
68[.]163[.]65[.]72
70[.]27[.]15[.]38
70[.]29[.]135[.]118
70[.]49[.]34[.]218
70[.]52[.]230[.]48
70[.]121[.]156[.]34
72[.]190[.]100[.]201
74[.]12[.]145[.]202
74[.]12[.]145[.]207
74[.]12[.]147[.]243
76[.]142[.]13[.]8
77[.]124[.]85[.]166
78[.]97[.]123[.]229
79[.]130[.]51[.]242
80[.]192[.]52[.]128
81[.]151[.]251[.]196
82[.]76[.]99[.]171
83[.]110[.]196[.]111
83[.]110[.]223[.]89
84[.]155[.]8[.]44
84[.]215[.]202[.]8
85[.]49[.]243[.]230
85[.]243[.]247[.]137
86[.]97[.]84[.]192
86[.]207[.]26[.]60
86[.]236[.]11[.]235
87[.]223[.]92[.]180
88[.]249[.]231[.]161

90[.]4[.]74[.]222
95[.]76[.]193[.]223
95[.]149[.]166[.]38
96[.]43[.]115[.]158
96[.]248[.]11[.]183
97[.]118[.]24[.]246
100[.]2[.]41[.]26
102[.]157[.]101[.]136
102[.]157[.]244[.]251
104[.]157[.]102[.]161
108[.]4[.]77[.]65
108[.]49[.]159[.]2
109[.]48[.]28[.]129
121[.]121[.]101[.]31
124[.]13[.]232[.]162
125[.]209[.]114[.]181
136[.]232[.]179[.]26
141[.]164[.]249[.]90
149[.]75[.]147[.]46
151[.]48[.]137[.]184
161[.]142[.]99[.]88
168[.]149[.]47[.]164
172[.]77[.]204[.]25
172[.]91[.]3[.]194
173[.]30[.]189[.]100
174[.]164[.]68[.]180
179[.]158[.]101[.]198
186[.]182[.]15[.]91
187[.]147[.]137[.]67
188[.]48[.]72[.]229
189[.]253[.]235[.]140
190[.]134[.]148[.]34
197[.]2[.]11[.]142
201[.]103[.]222[.]151
201[.]244[.]108[.]183
217[.]165[.]233[.]123

Subject Common Names

epyhm[.]net
twmbelz[.]org
iene[.]info
ctxehfdug[.]net

utip[.]biz
jaonioi[.]org
vcivoqeqfh[.]us
ineieutzvt[.]mobi
tuayjhrdwg[.]mobi
oxouy[.]mobi
iemjmedtey[.]biz
ouxtetbtn[.]biz
ghoaetksiwo[.]net
fwoht[.]org
tqouhdk[.]mobi
pidewaetbu[.]us
aihpe[.]mobi
zemureisir[.]info
oialk[.]com
ihaknpq[.]us
jqseote[.]us
gzfjtyr[.]com
aetzfeq[.]net
qbez[.]info
omloeceqiu[.]biz
ztiorhvb[.]net
lfad[.]mobi
egatcwojan[.]us
zcstobno[.]us
faexgkbimwe[.]org
bdae[.]info
xoehdsoeao[.]org
iekztmiw[.]com
oojfkdbgiec[.]info
ioiu[.]us
jaouem[.]info
xocsuioij[.]biz
euydxykaie[.]org
ipzc[.]net
lmatetu[.]mobi
woaitgja[.]info
kmeyih[.]org
bvgfkdinjla[.]net
mrokouejcei[.]mobi
ztmt[.]org
epmsxuv[.]info

vsasikavjed[.]biz
yieziqg[.]biz
zvtilriljat[.]net
vzxei[.]net
fbiafxq[.]info
pmeooxard[.]org
gimcyeeoof[.]org
qocu[.]org
eeapissopx[.]biz
otihelb[.]biz
ewaguarw[.]org
haeioeee[.]info
gokeokaut[.]biz
czqphiwowf[.]biz
ieaorbuq[.]net
tcnzewxk[.]us
lynle[.]biz
hzlfitjo[.]net
alcvl[.]info
wcyoloy[.]mobi
temthdmeo[.]org
zufmpz[.]mobi
lijivtamo[.]mobi
kouxe[.]org
aidoxovuncx[.]mobi
rtouaxye[.]biz
zevjeo[.]mobi
aispzwot[.]biz
paod[.]org
iqtfotoe[.]mobi
twdifusycee[.]biz
frkneeatb[.]info
eehpeplhr[.]us
aodkhtecx[.]net
efpohwf[.]net
oesyahoixic[.]us
motnooz[.]biz

Source: <https://embee-research.ghost.io/advanced-threat-intel-queries-catching-83-qakbot-servers-with-regex-censys-and-tls-certificates/>