Hunting Certificates & Servers

@erbbysam

DEFCON 27 PACKET HACKING VILLAGE
whoami

@erbbysam

Software Engineer

DC23, DC24 black badge (Badge Challenge, Co9)
The opinions expressed here are my own.

Nothing presented here gives you permission to exploit any host online. Always seek explicit permission.
Part 1 -- Scanning The Internet
Monitoring the Internet… for Black Badges

We noticed @1o57 would use the same TLDs in the DEFCON badge challenge between years…

Before (and after) DEFCON 24, we created monitoring for .codes and a few other TLDs for changes. We searched for:

- Common strings we know @1o57 uses
- Domains hosted on certain hosting providers
- Relevant whois records
Monitoring the Internet… for Black Badges

Our scanners worked!

https://gray.codes/
Monitoring the Internet… for Black Badges

Our scanners worked..... [https://gray.codes/](https://gray.codes/)

Mystery Challenge Registration

Brett B via googlegroups.com to benjamin.franklin, contact

Team Name: Council of 9

OUT OF ALL THE THINGS I HAVE LOST I MISS MY MIND THE MOST
LOST TIME IS NEVER FOUND AGAIN
THIS SHOULD BE A VERY EXCITING YEAR

---

1057 to Brett, benjamin.franklin, contact

Um...

It would appear as though you've been trolled....
Monitoring the Internet… for Black Badges

Our scanners worked.................. https://gray.codes/

(we were all trolled)

Congratulations to first four #mC teams:
1: Council of Nine (first to complete)
2: Vault Dwellers
3: Team Anti-Grifter
4: Psychoholics

https://twitter.com/Gray_Codes/status/816673994858524672
Monitoring the Internet…

For more than Black Badges?

This was my first experience monitoring & searching the public internet. This made me wonder what else could be found online…
TLS Certificates

- TLS Certificates contain hostnames!

Before I talk about scanning the internet for certificates, let's talk about TLS.
TLS Handshake

Client Random

Server Name Indication (SNI)

Handshake leading to master secret

TCP - 68ms

TLS - 136ms

Image via https://commons.wikimedia.org/wiki/File:Full_TLS_1.2_Handshake.svg
TLS Handshake

- Client Random
- Server Name Indication (SNI)

Handshake leading to master secret

I want the hostnames found here

Image via https://commons.wikimedia.org/wiki/File:Full_TLS_1.2_Handshake.svg
Early TLS Termination

Client Random

Server Name Indication (SNI) \{ ClientHello \}

I want the hostnames found here

SSLv3 Record Layer: Alert (Level: Fatal, Description: Handshake Failure)
- Content Type: Alert (21)
- Version: SSL 3.0 (0x0300)
- Length: 2
- Alert Message
  - Level: Fatal (2)
  - Description: Handshake Failure (40)
X509 Certificate SAN Example (google.com)

X509v3 extensions:
X509v3 Extended Key Usage:
   TLS Web Server Authentication
X509v3 Subject Alternative Name:
   DNS:*.google.com, DNS:*.android.com, DNS:*.appengine.google.com, DNS:*.cloud.google.com, DNS:*.crowdsourcing.google.com, DNS:*.g.co, DNS:*.gcp.gvt2.com, DNS:*.gcpcdn.gvt1.com, DNS:*.ggt.cn, DNS:*.google-analytics.com, DNS:*.google.ca, DNS:*.google.cl, DNS:*.google.co.in, DNS:*.google.co.jp, DNS:*.google.co.uk, DNS:*.google.com.ar, DNS:*.google.com.au, DNS:*.google.com.br, DNS:*.google.com.co, DNS:*.google.com.mx, DNS:*.google.com.tr, DNS:*.google.com.vn, DNS:*.google.de, DNS:*.google.es, DNS:*.google.fr, DNS:*.google.hu, DNS:*.google.it, DNS:*.google.nl, DNS:*.google.pl, DNS:*.google.pt, DNS:*.googleapis.com, DNS:*.googleapis.cn, DNS:*.googlecnnaps.cn, DNS:*.googlecommerce.com, DNS:*.googlevideo.com, DNS:*.gstatic.cn, DNS:*.gstatic.com, DNS:*.gstaticcnnaps.cn, DNS:*.gvt1.com, DNS:*.gvt2.com, DNS:*.metric.gstatic.com, DNS:*.urchin.com, DNS:*.url.google.com, DNS:*.youtube-nocookie.com, DNS:*.youtube.com, DNS:*.youtubeeducation.com, DNS:*.youtubekids.com, DNS:*.yt.be, DNS:*.ytiem.com, DNS:*.android.clients.google.com, DNS:*.android.com, DNS:*.developer.android.google.cn, DNS:*.developers.android.google.cn, DNS:*.g.co, DNS:*.ggpt.cn, DNS:*.goo.gl, DNS:*.google-analytics.com, DNS:*.google.com, DNS:*.googlecnnaps.cn, DNS:*.googlecommerce.com, DNS:*.source.android.google.cn, DNS:*.urchin.com, DNS:*.www.goo.gl, DNS:*.youtu.be, DNS:*.youtube.com, DNS:*.youtubeeducation.com, DNS:*.youtubekids.com, DNS:*.yt.be
TLS Scanning the Internet

For each ipv4 host:

- Masscan port 443 -- [https://github.com/robertdavidgraham/masscan](https://github.com/robertdavidgraham/masscan)
- Send a TLS Client Hello, disconnect after the Server Certificate is observed

Used golang, modified the golang TLS stack to terminate at the correct time & return a parsed x509 structure.

pseudocode.sh:
```
./masscan -p443 0.0.0.0/0 -oL masscan_output --excludefile exclude.conf
./golang_scanner masscan_output > scanning_results
```
Am I finding every host on port 443?

<table>
<thead>
<tr>
<th>Scanner</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>This Scanner</td>
<td>51,996,236</td>
</tr>
<tr>
<td>2015 Paper[0]</td>
<td>42,676,912</td>
</tr>
<tr>
<td>Shodan</td>
<td>58,188,083</td>
</tr>
<tr>
<td>Shodan, ipv4 limited “net:0.0.0.0/0”</td>
<td>42,881,125</td>
</tr>
</tbody>
</table>

[0] TLS in the wild: an Internet-wide analysis of TLS-based protocols for electronic communication
Ralph Holz, Johanna Amann, Olivier Mehani, Matthias Wachs, Mohamed Ali Kaafar
Am I finding every certificate?

**SNI**

Servers can use the Server Name Indication (SNI) in the TLS Client Hello to differentiate clients, returning different certificates.

Commonly used by CDN’s. Will not be found here.

**x509 Formatting Issues**

Golang’s x509 parser can be “too strict” when parsing malformed certificates.

This is best described here:

[https://sslmate.com/blog/post/how_certspotter_parses_255_million_certificates](https://sslmate.com/blog/post/how_certspotter_parses_255_million_certificates)
Who else is terminating TLS early?

I ran a TLS server and captured traffic for a few days...

3 clients terminated after the Server Hello:
- Server hosted at hosteurope.de
- Albert-Ludwigs-Universität Freiburg, Germany
- RWTH Aachen University, Germany

This could just be due to the self-signed certificate used during testing. However, no “Certificate Unknown” TLS alert was observed!

Additionally, no SNI was sent.
Part 2 -- Searching DNS Data
How to search large DNS datasets

This started as a problem I had. Rapid7 datasets were 10GB of unsorted, compressed DNS data.

This obviously took a long time to search.

https://blog.erbbysam.com/index.php/2019/02/09/dnsgrep/
How to search large DNS datasets (continued)

I took advantage of the DNS tree structure to sort the data:

com
com.erbbysam
com.erbbysam.blog

To make this a bit more generic/scriptable, I reversed the DNS name, then sorted:

moc.masybbre.golb,521.33.091.4
moc.masybbre,331.021.451.40
moc.masybbre.www,moc.masybbre

With this sorted data, I could now binary search to quickly find the records I wanted.

https://blog.erbbysam.com/index.php/2019/02/09/dnsgrep/
How to search large DNS datasets (continued)

I put this online using a golang webserver:

```json
ubuntu@client:~$ curl 'https://dns.bufferover.run/dns?q=erbbysam.com'
{
  "Meta": {
    "Runtime": "0.000361 seconds",
    "Errors": [
      "rdns error: failed to find exact match via binary search"
    ],
    "FileNames": [
      "2019-01-25-1548417890-fdns_a.json.gz",
      "2019-01-30-1549868121-rdns.json.gz"
    ],
    "TOS": "The source of this data is Rapid7 Labs. Please review the Terms of Service: https://opendata.rapid7.com/about/"
  },
  "FDNS_A": [
    "104.154.120.133,erbbysam.com",
    "54.190.33.125,blog.erbbysam.com",
    "erbbysam.com,www.erbbysam.com"
  ],
  "RDNS": null
}
```

https://blog.erbbysam.com/index.php/2019/02/09/dnsgrep/
https://github.com/erbbysam/DNSGrep
https://github.com/OWASP/Amass/pull/104
Part 3 -- Putting Everything Together
Putting This Together

- Built an efficient TLS scanner that runs once a week
- Built an efficient way to query DNS datasets

Created [https://tls.bufferover.run/dns?q=.defcon.org](https://tls.bufferover.run/dns?q=.defcon.org)

```bash
# curl 'https://tls.bufferover.run/dns?q=.defcon.org' 2>/dev/null | jq .Results
[
  "162.222.171.214,DEF CON Communications Inc.,p2ps0.defcon.org",
  "162.222.171.214,DEF CON Communications Inc.,p2ps1.defcon.org",
  "162.222.171.214,DEF CON Communications Inc.,p2ps2.defcon.org",
  "162.222.171.214,DEF CON Communications Inc.,p2ps3.defcon.org",
  ...
```
Comparing tls.bufferover.run/dns?q=

shodan.io
- Should contain similar results
- Not free

Certificate Transparency monitors (such as crt.sh)
- Only contains publicly trusted certificates
- Does not identify servers which use a given certificate

Rapid7 TLS dataset (https://opendata.rapid7.com/sonar.ssl/)
- I ♥ Rapid7 datasets, but I wish they were easier to use
- This contains only the “new” certificates they encounter on their scans

Many others -- OWASP amass source is a good resource
https://github.com/OWASP/Amass/issues/71
demo

Hack yourself first!

(credit troyhunt.com)
demo

Hack yourself first!

(credit troyhunt.com)

Hack your military first!

https://tls.bufferover.run/dns?q=.mil returns ~473,000 results!

Report what you find → https://hackerone.com/deptofdefense
demo

Hack your military first!

https://tls.bufferover.run/dns?q=.mil returns ~473,000 results!
Questions?

Contact: @erbbysam

Thank you for inspiration/ideas:
https://twitter.com/bbuerhaus
https://twitter.com/smiegles
https://twitter.com/tomnomnom
https://twitter.com/hacker_

Try this out today:
https://tls.bufferover.run/dns?q=.defcon.org