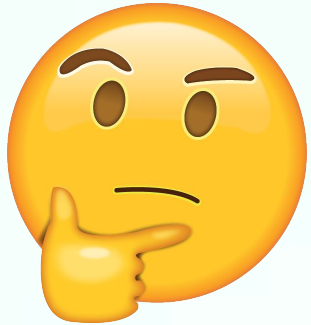
In Certificates We Trust?

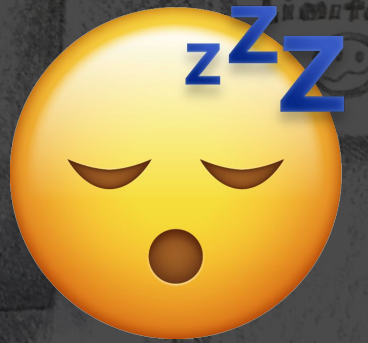
Mark Piper
Insomnia Security

- Append-only, immu
- Can be monitored
- Enables early detec
- https://www.cer

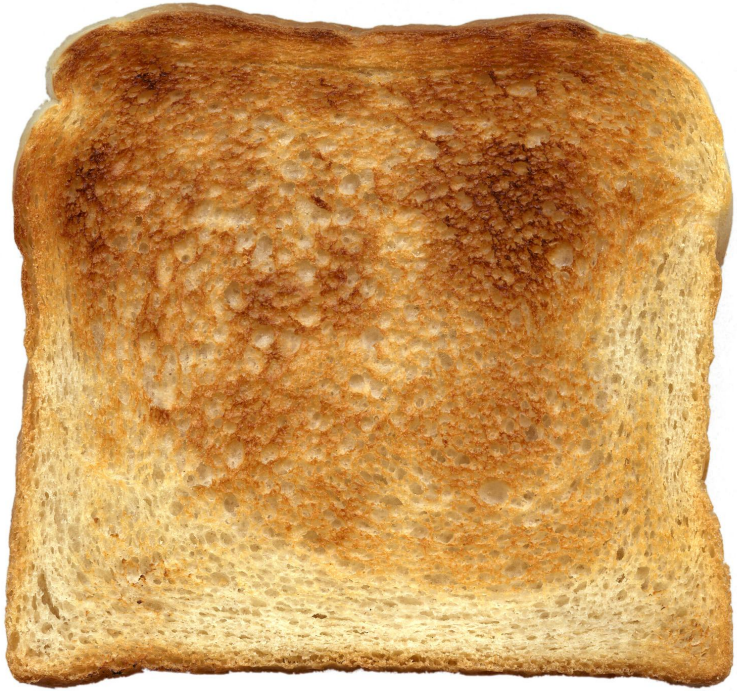


Certificate Authorities

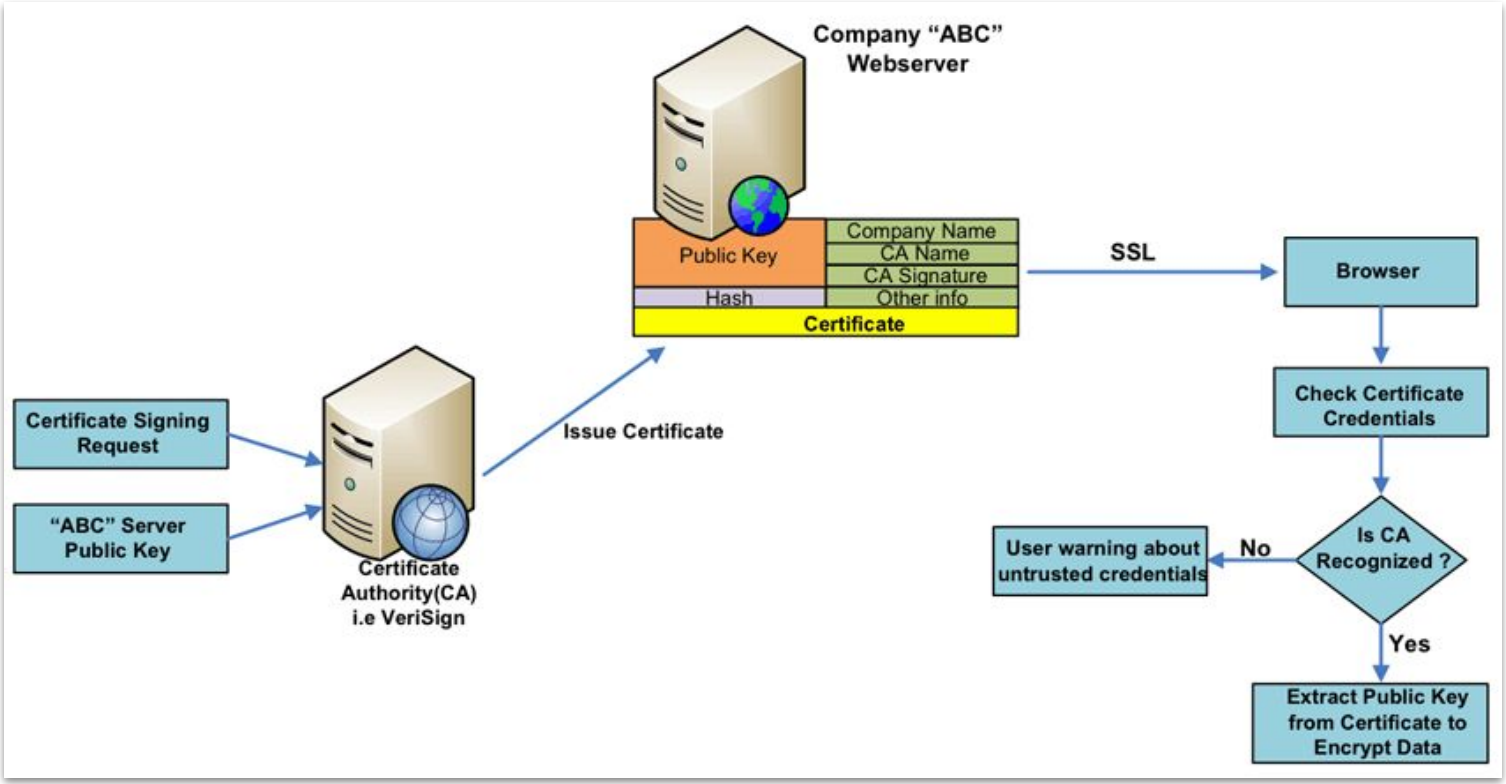
“A certificate authority (CA) is a trusted entity that issues electronic documents that verify a digital entity's identity on the Internet. The electronic documents, which are called digital certificates, are an essential part of secure communication and play an important part in the public key infrastructure (PKI)…”

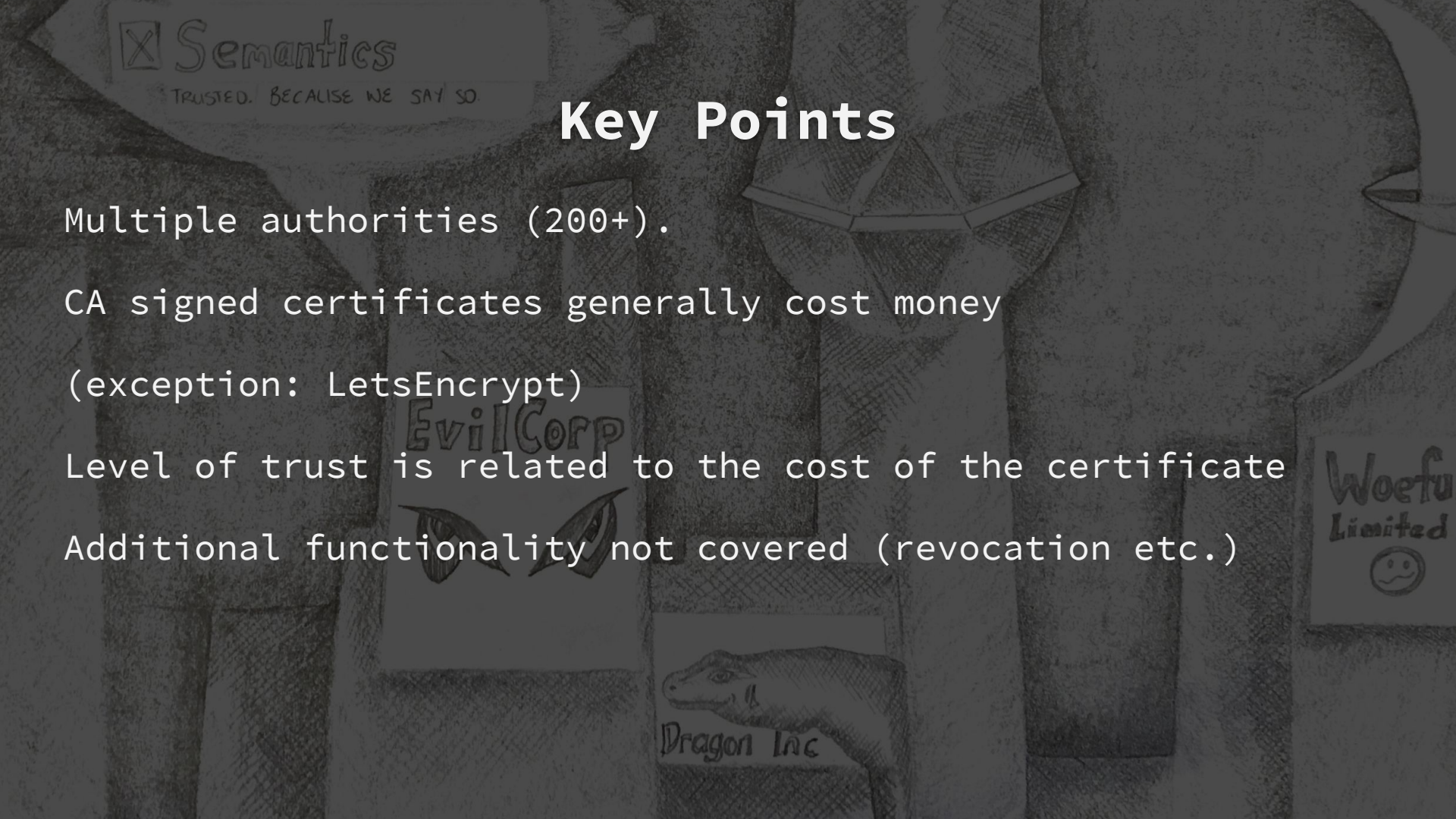


“Certificate Authorities are like toast” -- Metlstorm



Certificate Authorities





Key Points

Multiple authorities (200+).

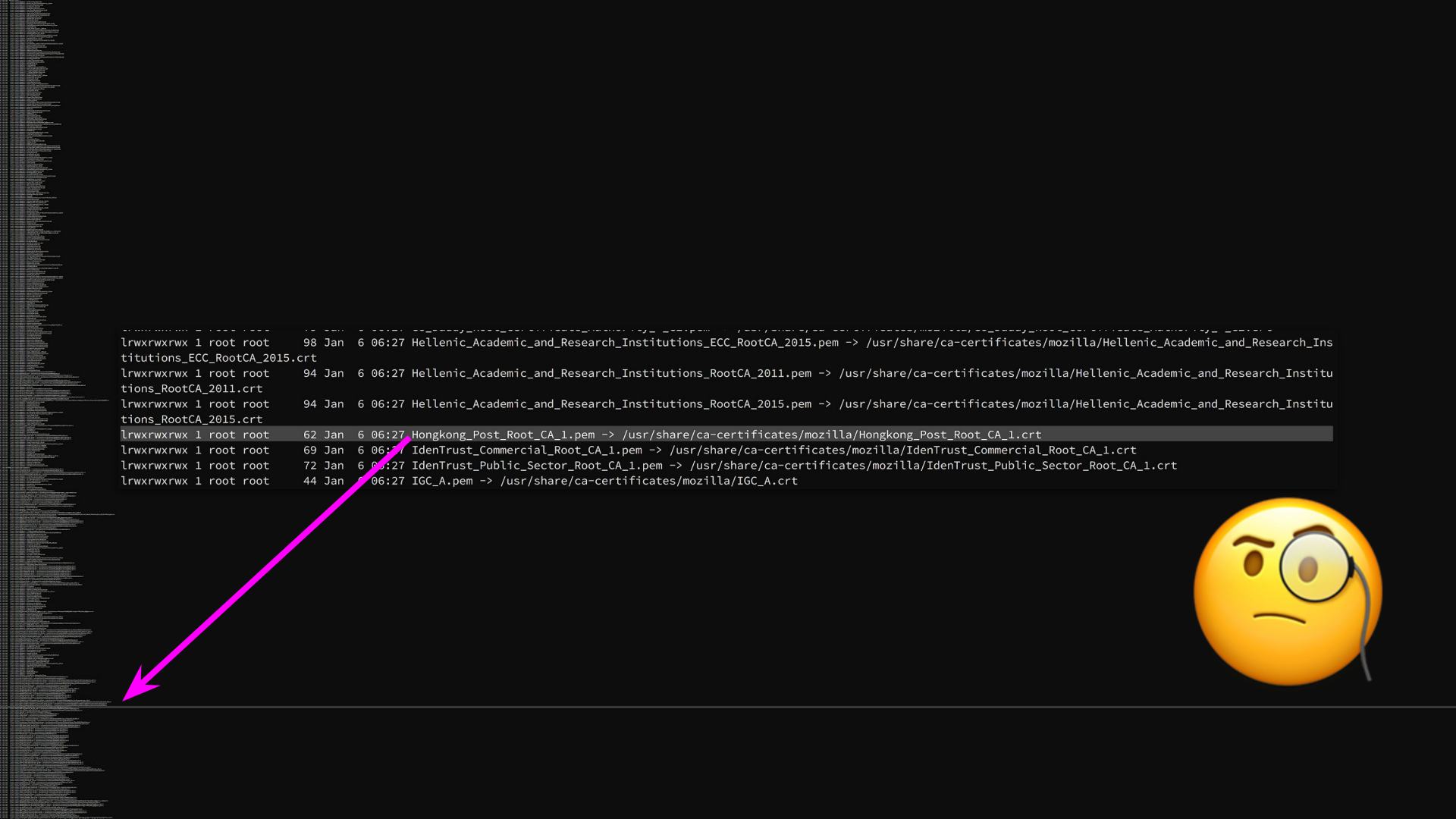
CA signed certificates generally cost money

(exception: LetsEncrypt)

Level of trust is related to the cost of the certificate

Additional functionality not covered (revocation etc.)

```
lrwxrwxrwx 1 root root     98 Jan  6 06:27 Hellenic_Academic_and_Research_Institutions_ECC_RootCA_2015.pem -> /usr/share/ca-certificates/mozilla/Hellenic_Academic_and_Research_Institutions_ECC_RootCA_2015.crt
lrwxrwxrwx 1 root root     94 Jan  6 06:27 Hellenic_Academic_and_Research_Institutions_RootCA_2011.pem -> /usr/share/ca-certificates/mozilla/Hellenic_Academic_and_Research_Institutions_RootCA_2011.crt
lrwxrwxrwx 1 root root     94 Jan  6 06:27 Hellenic_Academic_and_Research_Institutions_RootCA_2015.pem -> /usr/share/ca-certificates/mozilla/Hellenic_Academic_and_Research_Institutions_RootCA_2015.crt
lrwxrwxrwx 1 root root     62 Jan  6 06:27 Hongkong_Post_Root_CA_1.pem -> /usr/share/ca-certificates/mozilla/Hongkong_Post_Root_CA_1.crt
lrwxrwxrwx 1 root root     69 Jan  6 06:27 IdenTrust_Commercial_Root_CA_1.pem -> /usr/share/ca-certificates/mozilla/IdenTrust_Commercial_Root_CA_1.crt
lrwxrwxrwx 1 root root     72 Jan  6 06:27 IdenTrust_Public_Sector_Root_CA_1.pem -> /usr/share/ca-certificates/mozilla/IdenTrust_Public_Sector_Root_CA_1.crt
lrwxrwxrwx 1 root root     44 Jan  6 06:27 IGC_A.pem -> /usr/share/ca-certificates/mozilla/IGC_A.crt
```




Certificate Utopia?

Certificate Utopia?

Trust issues.

Multiple jurisdictions, poor standards

Complicated with “reseller” and intermediate cert issuing capabilities

Virtually no auditing of actual value (as demonstrated by Symantec)

Dozens of incidents over the last 10 years

Starter list: https://git.cryto.net/joepie91/ca-incidents

Observed Issues (So Far)

Issuing certificates to known bad sites / actors

Lack of request validation (issuing on behalf of another org)

Issuing wildcard certificates (MiTM / Interception Appliances)

Issuing test certificates certs for orgs (Google, Facebook, Yahoo, etc.)

Compromise of the CA itself

Example: Malware Issuance

2011: Comodo caught issuing certs for malware

Iranians owned one of their reseller account and issued certs

9 certificates issued in total

Yahoo.com cert was used in the wild

> Login.{yahoo,skype,live}.com

> {www,mail}.google.com

> addons.mozilla.org

Example: CA Compromise

2011: Diginotar

Issued wildcard certificates for Google and a few others

Certificates were used in the wild on attacks

Access to the HSM was online via unpatched

Windows Servers hosted attackers for months (years?)

Diginotar no longer exists

Example: Trust Chain Compromise

Symantec issued a intermediate CA to Blue Coat

Legitimate issue they hoped would go unnoticed

Allows for the creation of valid certificates for any domain within bluecoat appliances

“No no! We’re using them to FIND MiTM attempts!!!”

Example: Backdating

2015 WoSign issues two duplicate certificates

Included identical serial numbers, x509 extension values

Exception: Not Before field

Meant that the sites in question did not have to conform to SHA1 deprecation deadlines

Example: Test Certificates

2015 / 2016 Multiple “test” certificates issued by Symantec for a number of domains

Google got a bit grumpy, demanded an audit

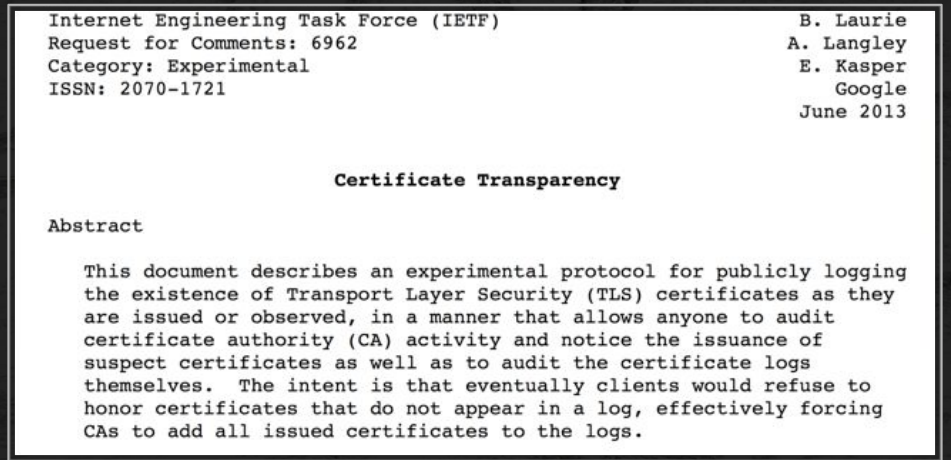
Found 30,000 certificates issued for legitimate domains

Google Justice

2013: IETF announces RFC 6962

Primarily drafted by Google security engineers

Proposes a new standard for transparency relating to certificate issuance by CA’s



Internet Engineering Task Force (IETF)
Request for Comments: 6962
Category: Experimental
ISSN: 2070-1721

B. Laurie
A. Langley
E. Kasper
Google
June 2013

Certificate Transparency

Abstract

This document describes an experimental protocol for publicly logging the existence of Transport Layer Security (TLS) certificates as they are issued or observed, in a manner that allows anyone to audit certificate authority (CA) activity and notice the issuance of suspect certificates as well as to audit the certificate logs themselves. The intent is that eventually clients would refuse to honor certificates that do not appear in a log, effectively forcing CAs to add all issued certificates to the logs.

RFC: Sixty Nine Sixty Two

“This document describes an experimental protocol for publicly logging the existence of Transport Layer Security (TLS) certificates as they are issued or observed, in a manner that allows anyone to audit certificate authority (CA) activity and notice the issuance of suspect certificates as well as to audit the certificate logs themselves.

The intent is that eventually clients would refuse to honor certificates that do not appear in a log, effectively forcing CAs to add all issued certificates to the logs…”

TL; DR?

Every time a certificate is issued from a CA, append basic details to a shared, immutable log

Have multiple logging instances

Log is public

Auditable!

Specifications for log storage, access, duplication etc

Q: If there is a public log of every certificate issued via a CA, does this mean we can query it?

Yes!

Several log interfaces exist:

* Google CT report (https://transparencyreport.google.com/https/certificates)
* Crt.sh (https://crt.sh/)
* Facebook Developer Tools (https://developers.facebook.com/tools/ct)

Graph API Explorer

Access Token: EAACEdEose0cBAIAq9T2ZCNDzksw3MAGqPLvwAHeil8B3FWZAJeeRE3jHKc9ZBakVqteyU

GET ▾ → /v2.12 ▾ / certificates?query=facebook.com

Node: certificates


```
{
  "data": [
    {
      "certificate_pem": "-----BEGIN CERTIFICATE-----
MIIGsjCCBZqgAwIBAgIQCzw7YBoY9Z7itrsFYF7ywDANBgkqhkiG9w0BA
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDV
d3cuZGlnaWNlcnQuY29tMS8wLQYDVQQDEyZEaWdpQ2VydCBTSEEyIEhpZ
dXJhbmNlIFNlcnZlciBDQTAeFw0xNzEyMTUwMDAwMDBaFw0xOTAzMjIxM
MGkxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpOYWxpZm9ybmlhMRMwEQYDV
ZW5sbyBQYXJrMRcwFQYDVQQKEw5GYWNlYm9vaywgSW5jLjEXMBUGA1UEA
YWNlYm9vay5jb20wWTATBgcqhkjOPQIBBggqhkjOPQMBBwNCAASIA87Ij
puN20BXCVsDjoP9wnF2rSV60qC130oLTrgfOQ3Uk1dv1R6LFCx4gs2pJU
/b+BXOUbo4IEGDCCBBQwHwYDVR0jBBgwFoAUUWj/kK8CB3U8zNllZGKiE
HQYDVR0OBBYEFMD9dPV9y8Yn8QPTYqJF14QcFSEIMIHHBgNVHREEgb8wg
ZmFjZWJvb2suY29tgg4qLnh4LmZiY2RuLm5ldIILKi5mYnNieC5jb22CD
ZmJjZG4ubmV0gg4qLmZhY2Vib29rLm5ldIIOKi54eS5mYmNkbi5uZXSCD
c2VuZ2VyLmNvbYIGZmIuY29tggsqLmZiY2RuLm5ldIIKi5mYi5jb22CE
YWNlYm9vay5jb22CDW1lc3Nlbmdlci5jb22CDGZhY2Vib29rLmNvbTAOB
Af8EBAMCB4AwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCMHUGA
MGwwNKAyoDCGLmh0dHA6Ly9jcmwzLmRpZ2ljZXJ0LmNvbS9zaGEyLWhhL
ci1nNi5jcmwwNKAyoDCGLmh0dHA6Ly9jcmw0LmRpZ2ljZXJ0LmNvbS9za
LXNlcnZlci1nNi5jcmwwTAYDVR0gBEUwQzA3BglghkgBhv1sAQEwKjAoB
BQcCARYcaHR0cHM6Ly93d3cuZGlnaWNlcnQuY29tL0NQUzAIBgZngQwBA
CCsGAQUFBwEBBHcwdTAkBggrBgEFBQcwAYYYaHR0cDovL29jc3AuZGlna
Y29tME0GCCsGAQUFBzAChkFodHRwOi8vY2FjZXJ0cy5kaWdpY2VydC5jb
aUNlcnRTSEEySGlnaEFzc3VyYW5jZVNlcnZlckNBLmNydDAMBgNVHRMBA
MIIBfgYKKwYBBAHWeQIEAgSCAW4EggFqAWgAdgCkuQmQtBhYFIe7E6LMZ
BPkb3jjd80OyA3cEAAAAWBXnEHoAAAEAwBHMEUCIBC3Rn4i2bhLyR344
vxoi+WPJBhGT+j1gJmg5AiEAwQ3rzH1mmMSYNYKtVNDZMo+l6e8Z35t+X
gWAAdwCHdb/nWXz4jEOZX73zbv9WjUdWNv9KtWDBtOr/XqCDDwAAAWBXn
AwBIMEYCIQCRjvvPARW3J1ENmo2Nz1cxisa1BcbDuqvSrfuXkz8btAIhA
8JjlVHUChiFzghsKVBeTxRagi55tgsAciaoZAHUAu9nfvB+KcbWTlCOXq
lQqrUugakJZkNo4e0YUAAAFgV5xCUgAABAMARjBEAiBY6qdNgMoQAqVTl
+X/1f/esBUczsb3MWdZ1ZgIgXdxZNTrDBgyTzxgbVROkbkqU3tZZdaiwsw4WI0xI0
BtQwDQYJKoZIhvcNAQELBQADggEBAGu0uxZD+IRXXlFWLPvknRkXA7J08NyVKG70
M2vDi2xF2YB8qlZgoxW8YiiV86IpwtOhYLZinSO0iCBDQmTf627LTPfuDcF6qOuO
WFTvj1IbplPvGWIu5tNBiFWNQxFAIL2Rf+5vmIe+YezUHTLGGqwRtFa2ImS17IMk
YjZ90LYXXO5qb1RKkFJtAvEBTbJsv8kr+J6Rx+YNJy17LnBX+MbWiyBbvUQoM3sY
MmcWmcaQmECz9ZHWYjZeufSHbHKG6KDYLU8x6DyhgtxK2rsoIMlNnJkNHaLjw+b8
7VCYa+EMWppvVuNyXOk9Jkbx7Q3SEoodT77kkHUX0bF2OkZy6cc=
-----END CERTIFICATE-----
".
```


Response received in 306 ms

Copy Debug Information | Get Code | Save Session

Certificate Transparency Monitoring

Certificate Transparency is an open framework to log, audit and monitor all publicly-trusted TLS certificates on the Internet. This tool lets you search for certificates issued for a given domain. Subscribe to email updates to be alerted when new certificates are issued.

Certificates | Subscriptions

facebook.com | Search for Domain

Send updates to: mark@aggre... | Subscribe

Domains	Subject	Issued by	Validity	Certificate
*.m.internmc.facebook.com *.internmc.facebook.com... More	/C=US/ST=California/L=Menlo Park/O=Facebook, Inc./CN=*.internmc.facebook.com	/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert SHA2 High Assurance Server CA	Feb 12, 2018 - Nov 02, 2018	Show Details (CT Precertificate)
*.fb.beta.messenger.com *.m.beta.facebook.com beta... More	/C=US/ST=CA/L=Menlo Park/O=Facebook, Inc./CN=*.beta.facebook.com	/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert SHA2 High Assurance Server CA	Feb 12, 2018 - Feb 20, 2019	Show Details (CT Precertificate)
m.intern.facebook.com *.m.intern.facebook.com *.in... More	/C=US/ST=CA/L=Menlo Park/O=Facebook, Inc./CN=*.intern.facebook.com	/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert SHA2 High Assurance Server CA	Feb 12, 2018 - Feb 20, 2019	Show Details (CT Precertificate)
*.fb.beta.messenger.com *.m.beta.facebook.com beta... More	/C=US/ST=CA/L=Menlo Park/O=Facebook, Inc./CN=*.beta.facebook.com	/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert SHA2 High Assurance Server CA	Feb 12, 2018 - Feb 20, 2019	Show Details (CT Precertificate)
*.alpha.facebook.com *.fb.m.alpha.facebook.com fb.... More	/C=US/ST=CA/L=Menlo Park/O=Facebook, Inc./CN=*.alpha.facebook.com	/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert SHA2 High Assurance Server CA	Feb 12, 2018 - Feb 20, 2019	Show Details (CT Precertificate)

Issues

Slow and buggy

Limited coverage

“Right side” search limitations (*i.e: *.insomnia**)

Still buggy

API Rate limiting

Grand LHKF Approach

“Fuck it. Let’s query the whole log set!”

Write python to download some logs

Write python to query the logs

For a match on our filter (.nz)

Save the certificate details

$$$ Profit $$$

Attempt #1: Manual Parsing

32gig / 2tb / 8 procs (4 cores / proc)

Download a series of individual logs

Use all the procs, filter for ‘.nz’, save raw PEM

Hacked up PoC code

Took ~7 days to execute in full

Issues

Parsing x509 is dumb and stupid and hates you

Turns out the logs are really big (actual big data)

Sample dataset was 900+gig (< 4 logs)

Data is stored within Merkle Hash Trees (MHTs)

Basically, it’s a massive MHT in 900gig+ of zip files

Disk is cheap. Ram is cheap. CPU isn’t.

ZIP processing is CPU intensive

```
atastore/ct-data/20170313-Pilot
ls
0  0a  14  1e  28  32  3c  46  50  5a  64  6e  78  82  8c  96  a0  aa  b4  be  c8  d2  dc  e6  f0  fa
1  0b  15  1f  29  33  3d  47  51  5b  65  6f  79  83  8d  97  a1  ab  b5  bf  c9  d3  dd  e7  f1  fb
2  0c  16  20  2a  34  3e  48  52  5c  66  70  7a  84  8e  98  a2  ac  b6  c0  ca  d4  de  e8  f2  fc
3  0d  17  21  2b  35  3f  49  53  5d  67  71  7b  85  8f  99  a3  ad  b7  c1  cb  d5  df  e9  f3  fd
4  0e  18  22  2c  36  40  4a  54  5e  68  72  7c  86  90  9a  a4  ae  b8  c2  cc  d6  e0  ea  f4  fe
5  0f  19  23  2d  37  41  4b  55  5f  69  73  7d  87  91  9b  a5  af  b9  c3  cd  d7  e1  eb  f5  ff
6  10  1a  24  2e  38  42  4c  56  60  6a  74  7e  88  92  9c  a6  b0  ba  c4  ce  d8  e2  ec  f6  index.csv
7  11  1b  25  2f  39  43  4d  57  61  6b  75  7f  89  93  9d  a7  b1  bb  c5  cf  d9  e3  ed  f7
8  12  1c  26  30  3a  44  4e  58  62  6c  76  80  8a  94  9e  a8  b2  bc  c6  d0  da  e4  ee  f8
9  13  1d  27  31  3b  45  4f  59  63  6d  77  81  8b  95  9f  a9  b3  bd  c7  d1  db  e5  ef  f9

atastore/ct-data/20170313-Pilot
wc -l index.csv
79027 index.csv

atastore/ct-data/20170313-Pilot
head index.csv
00042406bea95b00cc6d5633eb51ccd33fb5a739a8f5061a399ee72f3e84b96,thefinishingsuite.co.nz
00042406bea95b00cc6d5633eb51ccd33fb5a739a8f5061a399ee72f3e84b96,www.thefinishingsuite.co.nz
000e399f940f4069a2bf6e915ec0dc3ffcb18690e312363a97ea53fa49def41,*.happymeal.co.nz
000e399f940f4069a2bf6e915ec0dc3ffcb18690e312363a97ea53fa49def41,happymeal.co.nz
0011611c91f4d966ee6b704c5f530a0c23fabea832435e03ffdb7624e75cddc,wt.engage.ubiquity.co.nz
0017be983b42de24465488bd8153bfac942a91d33f24300279d9eb122b25d48,secure.bayofmanycoves.co.nz
0019634463c0012967cb78c5a066b59382b2b857afeccfee1ee75ecccbc08b1,*.spektrabright.co.nz
0019634463c0012967cb78c5a066b59382b2b857afeccfee1ee75ecccbc08b1,spektrabright.co.nz
001ce4709d038b01ec5481a2c11a8295970ce8754945e9ef3226056cd7a0baf,elitedoors.net.nz
001ce4709d038b01ec5481a2c11a8295970ce8754945e9ef3226056cd7a0baf,www.elitedoors.net.nz

atastore/ct-data/20170313-Pilot
head 00/000042406bea95b00cc6d5633eb51ccd33fb5a739a8f5061a399ee72f3e84b96
----BEGIN CERTIFICATE-----
IIFLzCCBBegAwIBAgISA0ZBET32e91woJ1BWHWTlHJjMA0GCSqGSIb3DQEBCwUAMEoxCzAJBgNV
AYTAlVTMRYwFAYDVQQKEw1MZXQncyBFbmNyeXB0MSMwIQYDVQQDExpMZXQncyBFbmNyeXB0IEF1
Ghvcml0eSBYMzAeFw0xNjExMDEyMjM5MDBaFw0xNzAxMzAyMjM5MDBaMCIxIDAeBgNVBAMTF3Ro
WZpbmlzaGluZ3N1aXRlLmNvLm56MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAngtG
1d2oZumNeDDxzJyYCNE4YpON02XaBLOZdi6RJhir/Zhxy/lnzRkon0UC25dYxge5ViffdOgUywE
IJ/VWv5CmJtM+hxGAkh3vt0I35Pd3HMXDPvsiD6zYSsZWJFyV+/NAeAGUJwAChyWCwokhfcleED
68n7X64vT/4GR+elWfbxbY59Nq1oX6LlYXpskwjsZE9E9dRtQG7+Cj6QxMah35iPcTs/vThHAUx
1iEkur0Z2EgU90NYeA4MagKo3LnJbMIPdxTqAuqooMtRGqflE2aRfBmCgRIm5CkgiQzp0gH+Az4
Mku1EuMAhkaNvizOdJkN54Jqn+W+6UdbQIDAQABo4ICNTCCAjEwDgYDVR0PAQH/BAQDAgWgMB0G
atastore/ct-data/20170313-Pilot
openssl x509 -text -in 00/000042406bea95b00cc6d5633eb51ccd33fb5a739a8f5061a399ee72f3e84b96
ertificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            03:46:41:11:3d:f6:7b:dd:70:a0:9d:41:58:75:93:94:72:63
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = US, O = Let's Encrypt, CN = Let's Encrypt Authority X3
        Validity
```

Attempt #2: Certstream

Certstream from Ryan Sears (@fitblip)!

‘Streams’ logs in ‘real-time’ into a consumable API

API clients for python, go etc.

Includes Axeman for log parsing

https://github.com/CaliDog

Attempt #2: Certstream

1) Download all the certificates
2) Send certificates to Google Storage
3) Import all the certificates into BigQuery
4) Query and export dataset via BigQuery
5) Process updated results (openssl + python)

```
atastore/ct-data/20171200-NZ-TEST
> ls -l
otal 3792560
rw-r--r--. 1 pipes pipes 486195764 Dec 15 07:51 000000000000.csv
rw-r--r--. 1 pipes pipes 483780954 Dec 15 07:50 000000000001.csv
rw-r--r--. 1 pipes pipes 485873185 Dec 15 07:51 000000000002.csv
rw-r--r--. 1 pipes pipes 484861732 Dec 15 07:50 000000000003.csv
rw-r--r--. 1 pipes pipes 486112832 Dec 15 07:51 000000000004.csv
rw-r--r--. 1 pipes pipes 485828180 Dec 15 07:51 000000000005.csv
rw-r--r--. 1 pipes pipes 484387856 Dec 15 07:50 000000000006.csv
rw-r--r--. 1 pipes pipes 486502966 Dec 15 07:50 000000000007.csv

atastore/ct-data/20171200-NZ-TEST
> wc -l 000000000000.csv
76514 000000000000.csv

atastore/ct-data/20171200-NZ-TEST
> head $_
ll_dns_names,chain_hash,cert_der,not_before,not_after
ail.thelittleredfox.co.nz cpanel.thelittleredfox.co.nz webdisk.thelittleredfox.co.nz webmail.thelittleredfox.co.nz,7b560706d5d4fe5fc69bf55483fb13388efb0645b22a4c9fb044f60746979530,MIIFdDCCBFygAwIBAgISAz0fR9BCR968bko
BCwUAMEoxCzAJBgNVBAYTAlVTMRYwFAYDVQQKEw1MZXQncyBFbmNyeXB0MSMwIQYDVQQDExpMZXQncyBFbmNyeXB0IEF1dGhvcml0eSBYMzAeFw0xNzExMTUxNDUzNDNaFw0xODAyMTMxNDUzNDNaMCUxIzAhBgNVBAMTGm1haWwudGhlbGl0dGxlcmVkZm94LmNvLm56MIIBIjANBgkqhk
KCAQEAtl6JOdmrXBdWjGvd/1qoe3pv0R+M/RUieY7cQGKIY+pKsSG5Ob+6u7lTiFpgPaYeVakFAqBK8Et/noIFD67nJLfFgav/3h+maPp5O9FO8D93QvuvjkmWocz1H7jILE9dhx/6CwhnlxHUi3VGeXrqoqtcidG6Xt5HSOh1fzHwEdOLJCn2TmSJNgX05ORVoCMPYcdSDCgXJCHb5zcEt
3o8u6d6lkqw2wONmF5rFjkCjKNMV5v5EvMEpVnMrMG+rlzF5S3mFp5rwkMhL/l1bQDDdQvlNP8vXbZZm34mjmvwjZgmtXN0JpNMRRdRzFS7VQIDAQABo4ICdzCCAnMwDgYDVR0PAQH/BAQDAgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjAMBgNVHRMBAf8EAjAAMB0GA1Ud
vGz/C3mTPDAfBgNVHSMEGDAWgBSoSmpjBH3duubRObemRWXv86jsoTBvBggrBgEFBQcBAQRjMGEwLgYIKwYBBQUHMAGGImh0dHA6Ly9vY3NwLmludC14My5sZXRzZW5jcnlwdC5vcmcwLwYIKwYBBQUHMAKGI2h0dHA6Ly9jZXJ0LmludC14My5sZXRzZW5jcnlwdC5vcmcvMIGBBgNVHRE
0dGxlcmVkZm94LmNvLm56ghptYWlsLnRoZWxpdHRsZXJlZGZveC5jby5ueoIdd2ViZGlzay50aGVsaXR0bGVyZWRmb3guY28uem6MIH+BgNVHSAEgfYwgfMwCAYGZ4EMAQIBMIHmBgsrBgEEAYLfEwEBATCB1jAmBggrBgEFBQcCAR
5jcnlwdC5vcmcwgasGCCsGAQUFBwICMIGeDIGbVGhpcyBDZXJ0aWZpY2F0ZSBtYXkgb25seSBiZSByZWxpZWQgdXBvbiBieSBSZW5aW5nIFBhcnRpZXMgYW5kIG9ubHkgaW4gYWNjb3JkYW5jZSB3aXRoIHRoZSBDZXJ0aWZpY2F0ZSBQb2xpY3kgZm9lbmQgYXQgaHR0cHM6Ly9sZXRzZ
G9yeS8wDQYJKoZIhvcNAQELBQADggEBAJMnfsiWCwtmsb9YyWgjLwW650G4tWsxkypyZPFjij4RShMmllcX3MsfGj4HUGNhxz7GNp2eYhi7h5A24J67a8WMmfQ/Adwmi4ERw29fKPUlLhQlwonJonv13GeGDrdmwaCPB0F9tSAuJAr0sY3WRPK2GYvJjmT6T3WEKp3rrRh6Lk7veT7XN8VR
BhafMZpfE8WRJZBEKfCDfHR+SKfKbvfae8bc1uzcQ+SmUbpfeh96tXgFb6yILP6t2QVqkFrxgXYqsr8pY9QjuQVcrj//0vKCH8iySJ14pUkxo/nWt/T+3JVuIzfeHqERyMxrDbE=,1510757623,1518533623
ni104988.cloudflaressl.com *.babet-modelleri.net *.callynnlynell.com *.clasburpieperte.gq *.cravoecanelaartedesign.com.br *.electrician-auckland.net.nz *.esportchannel.pl *.jackofalltrades.co.nz *.jntktpfb.cf *.lcs.
st.com *.monaskincare.xyz *.panelbeaters-northshore.co.nz *.proteinbaro.hu *.rewindmassage.co.nz *.safeworx.co.nz *.upsmarketinggoals.xyz *.whitetshirts.net babet-modelleri.net callynnlynell.com clasburpieperte.gq c
m.br electrician-auckland.net.nz esportchannel.pl jackofalltrades.co.nz jntktpfb.cf lcs.pl merrickvillecanalfest.com monaskincare.xyz panelbeaters-northshore.co.nz proteinbaro.hu rewindmassage.co.nz safeworx.co.nz u
tetshirts.net,01fa7521693914e50ee7460ec58273c90714e4f0ef02f5f68ba0418564de6c32,MIIGxTCCBmqgAwIBAgIQER6Z9awlcgM41mofUvBztDAKBggqhkjOPQQDAjCBkjELMAkGA1UEBhMCR0IxGzAZBgNVBAgTEkdyZWF0ZXIgTWFuY2hlc3RlcjEQMA4GA1UEBxMHU2Fs
T0RPIENBIExpbWl0ZWQxODA2BgNVBAMTL0NPTU9ETyBFQ0MgRG9tYWluIFZhbGlkYXRpb24gU2VjdXJlIFNlcnZlciBDQSAyMB4XDTE2MTExODAwMDAwMFoXDTE3MDUyMTIzNTk1OVowbDEhMB8GA1UECxMYRG9tYWluIENvbnRyb2wgVmFsaWRhdGVkMSEwHwYDVQQLExhQb3NpdGl2ZVN
iBgNVBAMTG3NuaTEwNDk4OC5jbG91ZGZsYXJlc3NsLmNvbTBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABDcEr+lfkNEPF86t+tS4fCLB1Lep/eUQNmgIq4gq+hd2mrkMhP594tL9VKlDj8/dUI9SQvQvEKPFwJNOQcWyqXSjggTFMIIEwTAfBgNVHSMEGDAWgBRACWFn8LyDcU/eEggsb9
xhFNCO84DDBQbCG7mlp7yLJgAwDgYDVR0PAQH/BAQDAgeAMAwGA1UdEwEB/wQCMAAwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCME8GA1UdIARIMEYwOgYLKwYBBAGyMQECAgcwKzApBggrBgEFBQcCARYdaHR0cHM6Ly9zZWN1cmUuY29tb2RvLmNvbS9DUFMwCAYGZ4EMAQIBM
Wh0dHA6Ly9jcmwuY29tb2RvY2E0LmNvbS9DT01PRE9FQ0NEb21haW5WYWxpZGF0aW9uU2VjdXJlU2VydmVyQ0EyLmNybDCBiAYIKwYBBQUHAQEEfDB6MFEGCCsGAQUFBzAChkVodHRwOi8vY3J0LmNvbW9kb2NhNC5jb20vQ09NT0RPRUNDRG9tYWluVmFsaWRhdGlvblNlY3VyZVNlcnZl
MAGGGWh0dHA6Ly9vY3NwLmNvbW9kb2NhNC5jb20wggMMBgNVHREEggMDMIIC/4Ibc25pMTA0OTg4LmNsb3VkZmxhcmVzc2wuY29tghUqLmJhYmV0LW1vZGVsbGVyaS5uZXSCEouY2FsbHlubmx5bmVsbC5jb22CFCouY2xhc2J1cnBpZXBlcnRlLmdxgh4qLmNyYXZvZWNhbmVsYWFydGV
szWN0cmljaWFuLWF1Y2tsYW5kLm5ldC5ueoISKi5lc3BvcnRjaGFubmVsLnBsghcqLmphY2tvZmFsbHRyYWRlcy5jby5ueoINKi5qbnRdHBmYi5jZoIIKi5sY3MucGyCGyoubWVycmlja3ZpbGxlY2FuYWxmZXN0LmNvbVISKi5tb25hc2tpbmNhcmUueHl6gh4qLnBhbmVsYmVhdGVycy
AqLnByb3RlaW5iYXJvLmh1ghUqLnJld2luZG1hc3NhZ2UuY28ubnqCECouc2FmZXdvcnguY28ubnqCFyoudXBzbWFya2V0aW5nZ29hbHMueHl6ghIqLndoaXRldHNoaXJ0cy5uZXSCE2WhbGxvbmVycmlja3ZpbGxlY2FuYWxmZXN0LmNvbVIQbWJuYXNraW5jYXJlLnh5eoIdcGFuZWxiZWF0
GVkZXNpZ24uY29tLmJyghtlbGVjdHJpY2l0eS5uZXQubnqCEGVzcG9yGNoYW5uZWwucGyCFWphY2tvZmFsbHRyYWRlcy5jby5ueoILam50a3RwZmIuY2aCBmxjcy5wbIIZbWVycmlja3ZpbGxlY2FuYWxmZXN0LmNvbVIQbWJuYXNraW5jYXJlLnh5eoIdcGFuZWxiZWF0
bnqCDnByb3RlaW5iYXJvLmh1ghNyZXdpbmRtYXNzYWdlLmNvLm56gg5zYWZld29yeC5jby5ueoIVdXBzbWFya2V0aW5nZ29hbHMueHl6ghB3aGl0ZXRzaGlydHMubmV0MAoGCCqGSM49BAMCA0kAMEYCIQCwfftSEVW7SHvIoBky4na2u7rDjsqL9A/Zx/VG0aMSbwIhAK9uGiN7JJp3nrR
4HG15,1479427200,1495411199
ww.sewfrisco.co.nz,7b560706d5d4fe5fc69bf55483fb13388efb0645b22a4c9fb044f60746979530,MIIFCjCCA/KgAwIBAgISA0KvjJBtq9tSiVU0HsvmR4TsMA0GCSqGSIb3DQEBCwUAMEoxCzAJBgNVBAYTAlVTMRYwFAYDVQQKEw1MZXQncyBFbmNyeXB0MSMwIQYDVQQDExp
vcml0eSBYMzAeFw0xNjA2MDIwMDA3MDBaFw0xNjA4MzEwMDA3MDBaMB4xHDAaBgNVBAMTE3d3dy5zZXdmcmlzY28uY28ubnowggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQD77JIMFsesM4GXs5whNzxvQVp1r5XOw9w3enMIOLOTLW1UE64rjFZOcSSDDA+VGbi/OXYTcjy3MF
HsBT/PiNAEKtVgH1ZupzW8IlBrVnEJnM0RvxE6h9Lekmv1BO9BiX0Sm3wEX9426BFYtm6dD3gAmV+JhSHa8X6GJioANWEQZQMwafz0SxbO1L7uPCT04nt4MKRquxa4s8h9/eJQ+eyzN2uaaAbPhP/JunVUPOiPrYO1GB3sk6EWHvGHsnbuffh8exYR03eT7+fciFkX9rMqkjac++L6PNmlP
o6BAgMBAAGjggIUMIICEDAOBgNVHQ8BAf8EBAMCBaAwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCMAwGA1UdEwEB/wQCMAAwHQYDVR0OBBYEFJivNrtRpqXCOPSDOJL9TinzCUFnMB8GA1UdIwQYMBaAFKhKamMEfd265tE5t6ZFZe/zqOyhMHAGCCsGAQUFBwEBBGQwYjAvBggr
c3AuaW50LXgzLmxldHNlbmNyeXB0Lm9yZy8wLwYIKwYBBQUHMAKGI2h0dHA6Ly9jZXJ0LmludC14My5sZXRzZW5jcnlwdC5vcmcvMB4GA1UdEQQXMBWCE3d3dy5zZXdmcmlzY28uY28ubnowgf4GA1UdIASB9jCB8zAIBgZngQwBAgEwgeYGCysGAQQBgt8TAQEBMIHWMCYGCCsGAQUFBwIB
lbmNyeXB0Lm9yZzCBqwYIKwYBBQUHAgIwgZ4MgZtUaGlzIENlcnRpZmljYXRlIG1heSBvbmx5IGJlIHJlbGllZCBicmGUGFydGllcyBhbmQgb25seSBpbiBhY2NvcmRhbmNlIHdpGggdGhlIENlcnRpZmljYXRlIFBvbGljeSBmb3VuZCBhdCBodHRwczovL2xldHNl
l0b3J5LzANBgkqhkiG9w0BAQsFAAOCAQEAJGFkwbmysWORh85Vo4R39SUWMr14Jp2LZwehlyIfjPWTNMDi4qDEO/RLZss83Dx3oE89c9eo0R0docCH1YXqPzqzod9ndWJfeOaYIUpaH036YqV8GkvTiHLTMwtwNlt40Q56bYtuWNuIJAQBdtxyXYlvsjhR4HHJQZmmWCuZHajZ0dsY7WFvN
4+GrrD5fqWBGVebyTAUkOHCaOaQScnUWMjtKdl+xZjTuPtfh02/5c7eSdOvpU91tTPbQ+sdCQYR6xn5pLYsVBEJpIakiKOXgbRCxMwWu3hrzO5F/ovu2Ow62Yex4eV6jqmtBdu9wg==,1464826020,1472602020
hesimplecoffincompany.co.nz www.thesimplecoffincompany.co.nz,7b560706d5d4fe5fc69bf55483fb13388efb0645b22a4c9fb044f60746979530,MIIFPTCCBCWgAwIBAgISA5FT00mCPmGEGo7Vqc6BYjuoMA0GCSqGSIb3DQEBCwUAMEoxCzAJBgNVBAYTAlVTMRYwF
XB0MSMwIQYDVQQDExpMZXQncyBFbmNyeXB0IEF1dGhvcml0eSBYMzAeFw0xNzA5MjQxMjEyMDBaFw0xNzA5MjQxMjEyMDBaMCcxJTAjBgNVBAMTHHRoZXNpbXBsZWNvZmZpbmNvb2Fubnowggg EiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC63mIUIjKEgby0juD2Y8Zw
IVjTUlxFGj9ZRNRo88Q2IuLScaFm/RrtLNFGHHu+watvq2sM1JHWpNyGnv/deRFKDMd4zRvKhXwrVC4L6TIFNHcmFJYHvm4+3kg8jVahH09pcWoxnrISt5KrG2SdMSUL7s9WHkMG1FoLi0PW8TIyIr4nu4Q6Mkj8ajGivukhraWlzrr7IIfChcU/bnj2ubvzDGyHj7bw0HipFVOtm6LxY9W
rGg/iOMvZIuxx8OUNkYU28JFYioD39c2bEjmqpcG1YXtgw42J5ndrS8JAgMBAAGjggI+MIICOjAOBgNVHQ8BAf8EBAMCBaAwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCMAwGA1UdEwEB/wQCMAAwHQYDVR0OBBYEFLBSm+uiaqkHmJf0lov4AgfK4XotMB8GA1UdIwQYMBaAFK
yhMG8GCCsGAQUFBwEBBGMwYTAuBggrBgEFBQcwAYYiaHR0cDovL29jc3AuaW50LXgzLmxldHNlbmNyeXB0Lm9yZzAvBggrBgEFBQcwAoYjaHR0cDovL2NlcnQuaW50LXgzLmxldHNlbmNyeXB0Lm9yZy8wSQYDVR0RBEIwQIIcdGhlc2ltcGxlY29mZmluY29tcGFueS5jby5ueoIgd3d3L
XB0bnkuY28ubnowgf4GA1UdIASB9jCB8zAIBgZngQwBAgEwgeYGCysGAQQBgt8TAQEBMIHWMCYGCCsGAQUFBwIBFhpodHRwOi8vY3BzLmxldHNlbmNyeXB0Lm9yZy8wgasGCCsGAQUFBwICMIGeDIGbVGhpcyBDZXJ0aWZpY2F0ZSBtYXkgb25seSBiZSByZWxpZWQgdXBvbiBieSBSZW5aW5nIFBhcnRpZXMgYW5kIG9ubHkgaW4gYWNjb3JkYW5jZSB3aXRoIHRoZSBDZXJ0aWZpY2F0ZSBQb2xpY3kgZm9lbmQgYXQgaHR0cHM6Ly9sZXRzZ
```

Ryan Sears
@fitblip

Following

Wow, so after moving #certstream to @googlecloud from @heroku to save on costs (I was spending ~$100/mo), I'm slapped in the face with a $1,000 bill for bandwidth charges?! #certstream is going to be offline until I can get this sorted, sorry for the downtime. WTF @Google.

Feb 1 – 16, 2018	Compute Engine Network Internet Egress from Americas to Americas: 8811.434 Gibibytes (Source: [redacted]	$979.38

1:57 PM - 17 Feb 2018

1 Retweet **6** Likes

4 1 6

Ryan Sears
@fitblip

Following

Finally bit the bullet and moved #certstream back to @heroku from @googlecloud. Waiting on the DNS changes to propagate, but the service should be live within the hour for everyone. Here's hoping for no more un-expected massive hosting costs out of the blue :(

1:04 PM - 19 Feb 2018

1 Retweet **4** Likes

1 4

Red Team Use

Review data for certificates relating to $target

Often includes internal alternative subject names (.lan)

Review data for an entire geographic region (.AU)

Understand extensions in use within certificates

Assist with generating corpus for x509 fuzzing

Blue Team Use

Visability!

Understand who is issuing certificates within your trusted environments

Determine if phishing is occurring against $target term

Understand certificates in use within malware (historic and current data)

Discovery of unexpected fb.com certificates

Earlier this year, our Certificate Transparency monitoring service alerted us to an important opportunity to better align internal certificate policies. Specifically, we learned that the Let's Encrypt CA issued two TLS certificates for multiple `fb.com` subdomains.

These two certificates raised red flags for our team because they:

- were not issued by our primary CA vendor
- were not authorized by our security team
- were shared with multiple domains that we do not own or control

Blue Team Use

Test Project:

1) Tap the certstream:

Write python to tap the certstream for all new events.

2) Filter by subject:

Filter for anything to do with ‘paypal’, ‘appleid’, ‘.govt.nz’ etc.

3) Determine if the certificate status (new or not):

Certs end up in multiple logs, maybe reissued etc.

4) Look up information about the domain (http, dns etc):

Gather some intel on the target domain.

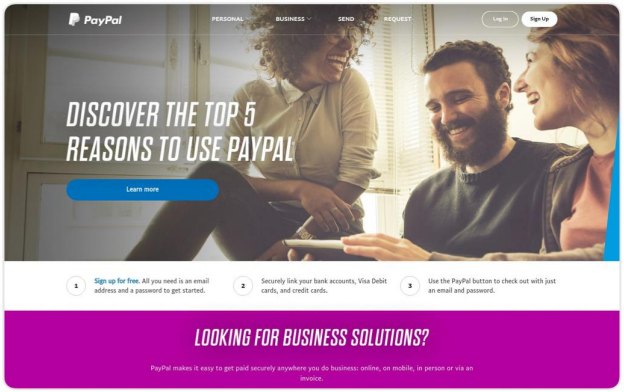
5) Generate monitor event into Slack for monitoring:

Fire a webhook and notify us.

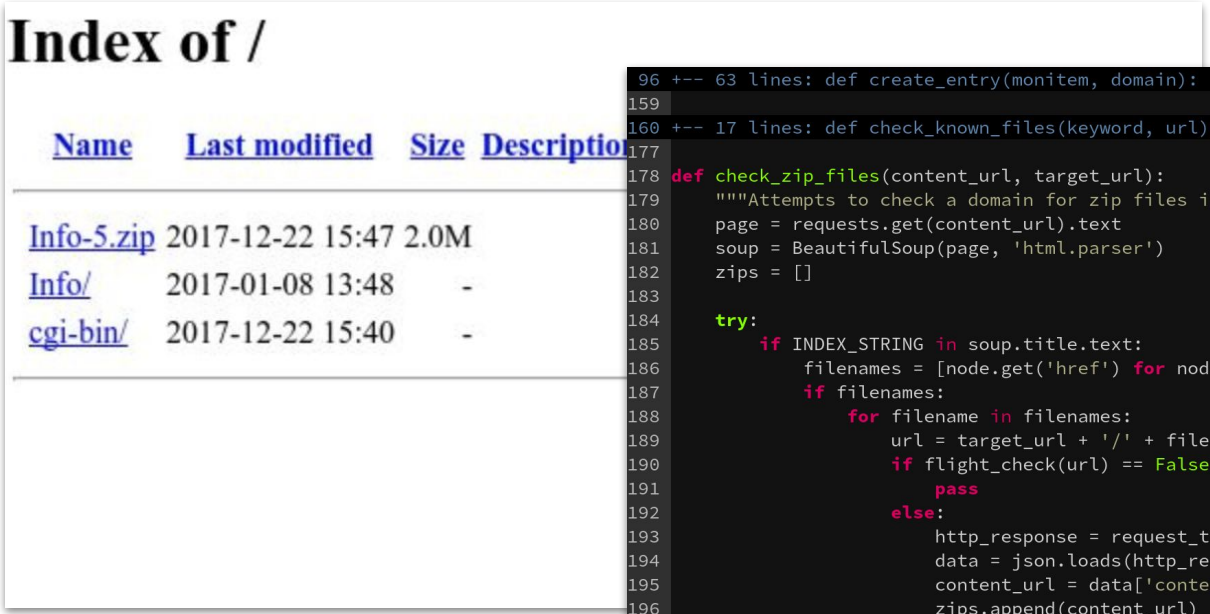
CT BOT APP 12:03 PM

🔥 New certificate spotted:

Keyword: paypal
URL: hxxps://verify-account-paypal.cf
IP: 217.61.98.14
VirusTotal: https://www.virustotal.com/#/search/217.61.98.14
HTTP Request: https://storage.googleapis.com/sig6-requests/4344e994db13f65aca064c6883e670cd.txt
Screenshot: https://storage.googleapis.com/sig6-screenshots/4344e994db13f65aca064c6883e670cd.png



Blue Team Bonus: Kit Stealing




```
 96 +-- 63 lines: def create_entry(monitem, domain):  Attempt to create an entry for the domain in the db---------
159
160 +-- 17 lines: def check_known_files(keyword, url):  Checks a list of known bad files defined in config.yaml---
177
178 def check_zip_files(content_url, target_url):
179     """Attempts to check a domain for zip files in a file index"""
180     page = requests.get(content_url).text
181     soup = BeautifulSoup(page, 'html.parser')
182     zips = []
183
184     try:
185         if INDEX_STRING in soup.title.text:
186             filenames = [node.get('href') for node in soup.find_all('a') if node.get('href').endswith('zip')]
187             if filenames:
188                 for filename in filenames:
189                     url = target_url + '/' + filename
190                     if flight_check(url) == False:
191                         pass
192                     else:
193                         http_response = request_text(url, filename).decode('utf-8')
194                         data = json.loads(http_response)
195                         content_url = data['content_url']
196                         zips.append(content_url)
197                 return zips
198     except:
199         traceback.print_exc(file=sys.stdout)
200         return zips
201
202 +--  7 lines: def update_entry(domain):  Updates the last_seen for a given domain-----------------------------
209
210 +--  9 lines: def flight_check(url): Run a pre-flight check on the domin before we do anything else-----------
```

Getting Started

Dozens of projects out there:

Bounty Monitor: https://github.com/nashcontrol/bounty-monitor

S3 Monitoring: https://github.com/eth0izzle/bucket-stream

Phish Catcher: https://github.com/x0rz/phishing_catcher

Certstream Slack: https://github.com/heptiolabs/certstream-slack

…

Conclusion

CA’s can’t be trusted, so we now log all certificates issued are logged publicly

This log is freely accessible (if somewhat large)

It is possible to process the log in bulk and filter for useful domains

It is also possible to incrementally monitor the log

There are awesome starter projects / ideas to explore