TLS Metadata Analysis of Cobalt Strike Certs (Bad Sites) vs Good websites Certs:

We have observed the use of Cobalt strike by a lot of malware, and the use of certificates (encryption) makes payload analysis difficult. Still, the metadata of the certificate used by cobalt strike (malware) gives you a detection opportunity.

I have analysed cobalt strike certificates metadata from PCAP provided by Brad Duncan @malware_traffic vs an extensive list of top genuine website certificates.

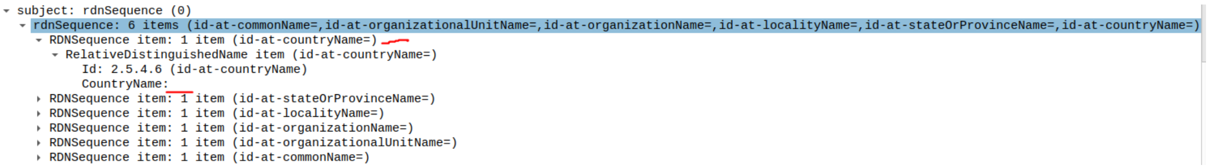
Metadata use ->

Certificate Subject, Issuer, Country, URI, Length, DNS name and Validity.

Upon analysis of metadata of certificates used by malware, shows a lot of difference compared with certificates used by good websites.

1) Length (tls.handshake.certificate_length):- As certificates used by malware lacks information such as Subject/Issuer/country details, their overall length gets reduce, though some uses free certificates provided by Let's encrypt (Certificate Authority) still lacks some information like subject, a country name which reduces the length (less than 1000 bytes)

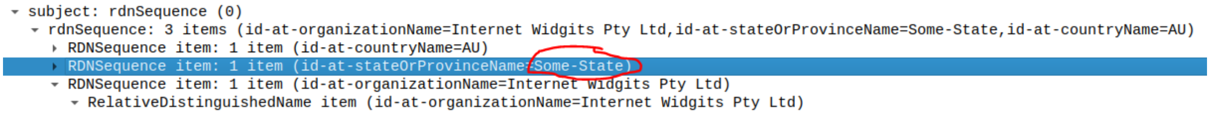
2) Country Name (x509sat.CountryName):- In certificates used by malware, you will see missing country details. Subject and Issuer both the fields show up this information, but if any of the subject or issuer details are missing, you will see this field blank; in some good certs as well, you can see these details missing (but very rare). You can see below all the details are missing.



```
▾ subject: rdnSequence (0)
  ▾ rdnSequence: 6 items (id-at-commonName=,id-at-organizationalUnitName=,id-at-organizationName=,id-at-localityName=,id-at-stateOrProvinceName=,id-at-countryName=)
    ▾ RDNSequence item: 1 item (id-at-countryName=)
      ▾ RelativeDistinguishedName item (id-at-countryName=)
          Id: 2.5.4.6 (id-at-countryName)
          CountryName:
    ▸ RDNSequence item: 1 item (id-at-stateOrProvinceName=)
    ▸ RDNSequence item: 1 item (id-at-localityName=)
    ▸ RDNSequence item: 1 item (id-at-organizationName=)
    ▸ RDNSequence item: 1 item (id-at-organizationalUnitName=)
    ▸ RDNSequence item: 1 item (id-at-commonName=)
```

3) Subject (x509sat.uTF8String) : The subject is nothing but the owner of the certificate here, you can see the details of the certificate owner.

when checked for the details of the good certificate, we can see here domain name, but for a fake certificate, this information was missing, and for some, it had fake details, e.g. Some-State



```
▾ subject: rdnSequence (0)
  ▾ rdnSequence: 3 items (id-at-organizationName=Internet Widgits Pty Ltd,id-at-stateOrProvinceName=Some-State,id-at-countryName=AU)
    ▸ RDNSequence item: 1 item (id-at-countryName=AU)
    ▸ RDNSequence item: 1 item (id-at-stateOrProvinceName=Some-State)
    ▾ RDNSequence item: 1 item (id-at-organizationName=Internet Widgits Pty Ltd)
      ▾ RelativeDistinguishedName item (id-at-organizationName=Internet Widgits Pty Ltd)
```

So we should observe these details.

Note: I am struggling to get which filter to use to fetch details specifically related to subject and issuer, but I have used x509sat.uTF8String and x509sat.printableString to fetch all the details from subject and Issuer fields.

4) Issuer (x509sat.printableString):- Here, I have observed some data is missing like OrgName, Locality, State, and many malware use certificates from Lets Encrypt (Certificate Authority). So here as well, some details can be missing or fake.

5) Validity (x509af.utcTime) :- so the Majority of certificates use of malware found to be valid for one year only. Still, it’s not a good property to detect the malicious intention of the certificate as good sites also have a validity of one year.

In this case, we can combine other details like no domain associate with IP or missing additional information.

6) DNSname (x509ce.dNSName):- If dnsName field is missing means it’s just IP based communication (no domain); you can combine other fields like missing fields or wrong information, short length.

If the domain is associated, good domains will issue certificates for subdomains, but the same was not observed for malware-related domains.

6) CRL (x509ce.uniformResourceIdentifier) :- For IP based communication this information was missing, but not good candidate for detection.

Above mentioned ones are easy to detect, like missing information or some random words in certificate details.

We can explore areas like the geographical focus of certificate distribution organisations; here, you can find root and intermediate CA lists.

https://wiki.mozilla.org/CA

Certificate extensions (x509af.extensions) also has some fields which can be explored too.

So that’s it for this analysis; I have kept metadata of certs below.

https://github.com/sankyhack/CobaltStrike-TLS-Metadata-Analysis

Thank You!!!