Assignment: Tracking Certificate Misissuance in the Wild

Anuraag Baishya

Implementation: 14 points

For the implementation part of the assignment, you will implement a simple 'certificate' linter called `certlint.py` in `Python 3`. The 'certificate' is a `JSON` file. An example is included in the zip file provided. The zip file is password protected. I hope you would have found the password from my presentation. To lint the certificate, follow the following guidelines:

- Subject Name:
  - Must include Organisation, Country and Common Name
  - Common name must be a valid domain name
  - Should include Locality and State/Province
- Issuer Name:
  - Must include Organisation and Country
  - Should include Locality and State/Province
- Should be valid for the year 2019
- Valid To and Valid From fields must have valid dates.
- Valid To date must be later than Valid From date
- Should include signing algorithm

Website names and Top Level domains (TLD) for Common Name field can be fictitious, however the format should be correct. The general format for a valid Common name is either `<subdomain.domain.tld>` or `<domain.tld>`. Wildcard character (*) can only be present in subdomains. Similarly CAs, signing algorithms, countries, states and localities can be fictitious.

Some valid Common Name field information:

- `www.domain.com`
- `domain.com`
- `*.domain.bla`

Some invalid Common Name field information:

- `www.`
- `domain.*`
- `*.domain`

NOTE: Multiple TLDs will not be present in test cases. (For eg: `.gov.uk`, `.co.in`, etc will not be present)

The program must output all errors and warnings for the 'certificate'. Errors in fields designated as 'must' should be output as an Error and errors in fields designated as 'should' should be output as warnings. The output format is `ERROR: description` for errors and `WARNING: description` for warnings.

You are free to use any libraries that you may wish for JSON parsing, however you must code the checks. Please include any external dependencies in a `requirements.txt` file.

The division of points is as follows:

- Opening the zip file: 2 points
- Implementing the certificate linter: 12 points

The code will be tested in a `Python 3.7` environment. The program would be run against a few test cases and points will be awarded on basis of accuracy in printing errors and warnings. The program would be run as `python3 certlint.py certificate.json`

If you are absolutely unable to find the password for the zip file, send me an email and I will send you a zip file which is not password protected at the cost of the 2 points for opening the zip file.

Analysis and Writing: 6 points (3 + 3)

Question 1:

Assume that for you next project you have to choose between **Comodo Positive SSL** certificate and **DigiCert Standard SSL** certificate. Which would you choose and why? Answer in 2-3 statements.

Question 2:

What harm can a rogue CA cause in a SSL context? Give 2-3 examples.

[In lieu of **any one** question, feel free to answer one of your two thought provoking questions from the summary in 2-3 statements. Please mention the question before answering.]

Submission

Please submit a `.tar.gz` file, named `advsec_jhed.tar.gz`, containing a PDF file with the answers to the two questions and the `certlint.py` file via email to `anuraag@jhu.edu`. The subject line should be 'Advanced Topics Assignment'. The assignment is due **Monday, March 18, 11:59 pm**. Feel free to email me if you have any doubts, comments or concerns.