TruSTAR Software Engineering Questionnaire

Ecosystem Screener

Thank you for your interest in joining the TruSTAR engineering team! We invite you to complete this screener. Please return your completed questionnaire before the end of the week so we can grade it.
Please submit code as if you intended to ship it to production. The details matter. Documentation and tests are expected, as is well written, simple, idiomatic code. Feel free to use an IDE and different libraries to avoid writing code from scratch. Your code should be ready to package and deploy if needed.

We found that creating a brief github project is the best way to deliver the code.

1. Write a function that allows you to extract valuable information from a string representing a json object. The input of the function is a string that contains a json object, and an array of properties and sub-properties (using dot notation).
The output of the function is a dictionary containing the values of each one of the matching properties. If the property or sub-property is not present, the entry doesn't show up in the result. For example, if the inputs are (written in multiple lines for clarity)

```
a=' {
      "guid": "1234",
      "content": {
            "type": "text/html",
            "title": "Challenge 1",
            "entities": [ "1.2.3.4", "wannacry", "malware.com"]
      },
      "score": 74,
      "time": 1574879179
  }'
```

```
b = ["guid", "content.entities", "score", "score.sign"]
```

then

```
> f(a,b)
> { "guid": "1234", "content.entities": [ "1.2.3.4", "wannacry",
"malware.com"], "score": 74}
```

2- If not done already, modify the function above to accept an arbitrarily nested sequence of properties (e.g `"content.link.href.parent"`)

3- Modify the function above to allow accessing arrayed properties. For example:

```
> f(a,["guid", "content.entities[0]")
> { "guid": "1234", "content.entities[0]": "1.2.3.4"}
```

If the property is not arrayed or the index is not found, there's no need to generate an output for it. Consider that the array access can happen in the middle of the property chain (for example: `"content.entities[0].time"`)

4- The MITRE organization maintains and publishes a catalog of different cyber entities that we use extensively at TruSTAR. Their repository is publicly accessible at https://github.com/mitre/cti.

For this portion of the questionnaire, we ask you to build a full program that, using the function above and other libraries of choice (anything that allows you to browse directories and files on a git repo), extracts and outputs the following fields of interest from all the json files found under the given path.

`Path: https://github.com/mitre/cti/enterprise-attack/attack-pattern`
`Fields: ["id", "objects[0].name", "objects[0].kill_chain_phases"]`

(Optional - only if you have spare time)
5- URLHaus is a project that tracks malicious URLs used for malware distribution. It's possible to obtain different datasets from this URL: https://urlhaus.abuse.ch/api/#retrieve.
Create a new program, or include an option in your existing one (command, parameter, route, depending on how you built it) that allows to obtain a curated list of *unique active malware urls* from URLHaus and store it in different formats: json, csv, xml, or a SQLite table