Risk Visualization for

“The Pampered Pets”

Executive Summary

This application was designed to support the process of generating threat profiles according to the OCTAVE-S risk assessment framework (Alberts & Dorofee, n.d; Alberts et al., 2005). After developing a risk analysis report, Cathy, the manager of the "Pampered Pets" store, was not convinced by the risk analysis report that compared the risk assessment of the status quo of her store to the one after digitalisation. So there was a need for a visualization tool that would help her decide. There were no Python libraries that can directly generate threat profiles. The available libraries were mostly for attack tree developments. However, it was more convenient to continue this part of the assignment by developing a python application to visualize threat profiles according to different parameters supplied in a JSON file format. For that purpose, I used Graphviz library, an open-source graph visualization library (The Graphviz Authors, n.d). It was used to generate threat profiles using directed graphs. A qualitative assessment will also be possible using this application that generates a PI (Probability-Impact) score for each threat profile and a total PI score for the whole scenario. This number is valuable for comparison and decision-making purposes. It was developed with referent to a risk assessment matrix (P. M. Training, 2022).

Application Description

The application accepts JSON file that contains threat properties according to the OCTAVE-S framework (Alberts & Dorofee, n.d). Each JSON file could include information about threats to several assets in one scenario. The data that needs to be supplied in the JSON file is related to Assets, Access (Optional), Actors, Motives (Optional), and Outcomes, along with probability and impact scores. The software will then generate threat profiles and calculate a PI score for each threat profile and a total PI score for the scenario to allow comparison.

The application will process the JSON files in the 'data' folder, generate threat profiles and PI (Probability-Impact) scores, and export them to the 'output' folder. The 'output' folder will contain other folders with the projects' names. Each project folder will contain other folders with the analyzed scenarios' names.

Each output for each analysis is composed of different threat profiles, and a 'PI Score.txt' file will contain the PI score of each threat profile in addition to the total PI score of the scenario (Figure 1).

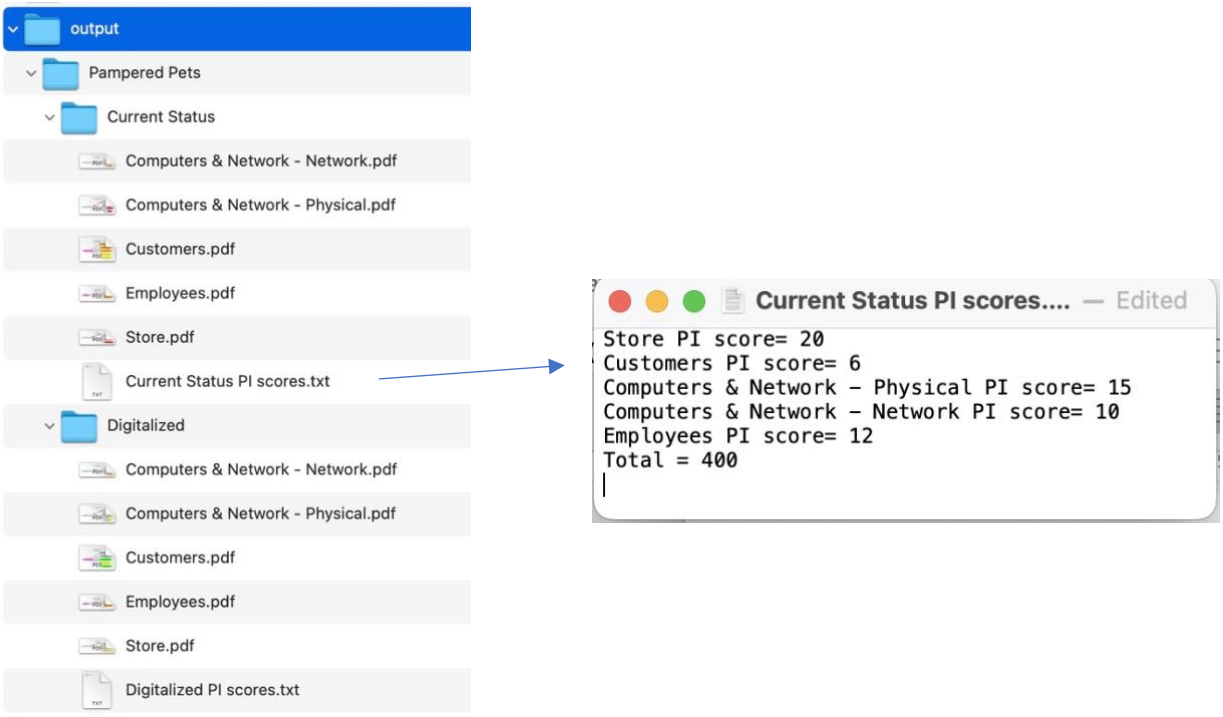


Figure 1 Output Folders

JSON File Structure

The JSON file contains the following main key-vale pairs:

- **Project,** for the project name ("Pampered Pets" in the sample data)
- **Analysis**, for the analysis name ("Current Status" and "Digitalized" in the sample data).
- **Assets** for an array of threat profile properties. Each Asset contains the following key-value pairs:
- **Name,** for the asset name.
- **Access,** for the access. For example, network access or physical access.

- **Actors**, for the actors. For example, inside or outside actors. Each one has a **Probability** attribute that will be used for calculating the PI score.
- **Motives**, for the motives. For example, accidental or deliberate. Each one has a Probability attribute as well.
- **Outcomes**, for the possible outcomes. Each has an **Impact** attribute that will be used for the PI score calculation.
- **Impossible** for an array of the impossible actor-motive and actor-motive-outcome combinations in array format that will be excluded from the processing. For example, ["Outside","Accidental"] means that the application will not connect an outside actor to an accidental motive during the threat profile generation process and PI score calculation.

Threat Profiles

An example of a threat profile generated by the application is shown below (Figure 2).

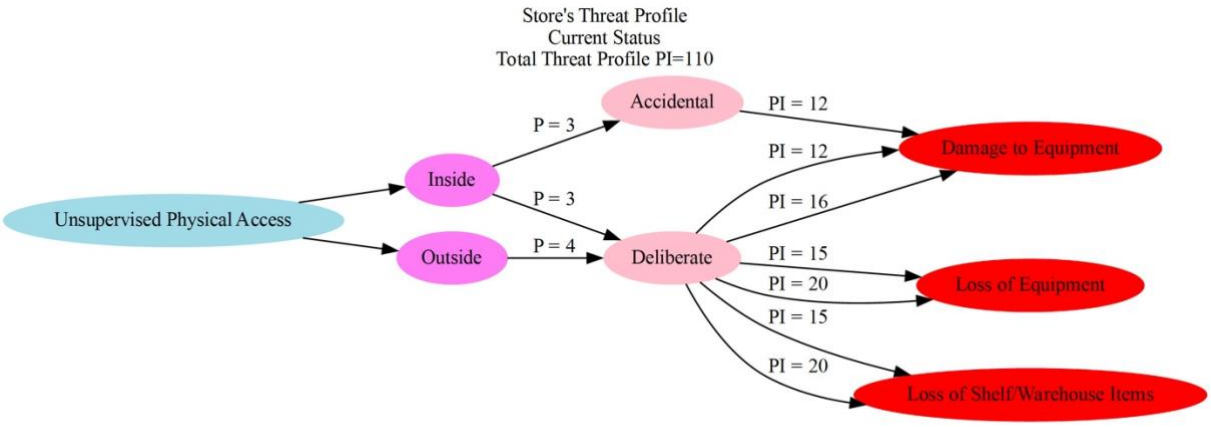


Figure 2 A threat profile for the store asset.

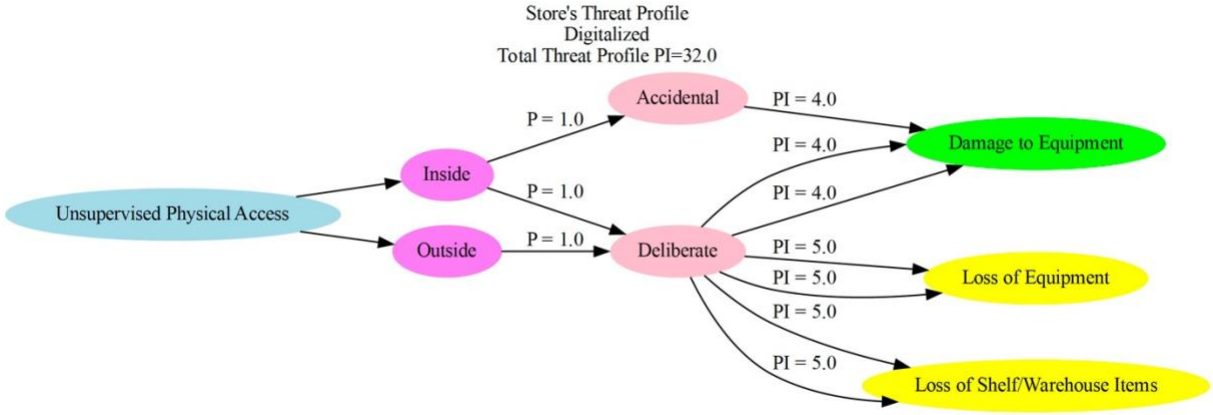
The first category on the left is "Access". The next is "Actors", followed by the "Motives", then “Outcomes”. The first number group on the left represents the probability. Probability is calculated by adding the actor's probability to the motive's probability or representing the actor's probability only if no motives were defined. The second number group on the right represents the PI scores. The PI score is the product of probability multiplied by the impact of the outcome. The total PI score for the threat profile is shown on the third line of the title.

Outcome color coding

The outcome will be color-coded according to the highest PI score. For example, all the outcomes shown in Figure 1 were red because the highest PI score for each outcome was extreme. Please note that each outcome might have more than a PI score due to different threat routes. Figure 2 shows different colors due to different PI scores. This provides feedback to the viewer to enable quick comparison. The color scale according to the PI score is as follows (P. M. Training, 2022):

PI Score range	Meaning	Color
1 - 2	Negligible	Gray
3 - 4	Low	Green
5 - 9	Medium	Yellow
10 - 14	High	Orange
15 - 25	Extreme	Red

Figure 3 A threat profile for the digitalized store asset.



Comparison of the Threat Profiles and PI scores

As was shown in the previous assignment of the analysis report, the digitization of the "Pampered Pets" is recommended due to reduced overall risk that can be inferred from the total PI scores. This can be quickly done by opening the out folder and viewing each analysis's 'PI score.txt'. The file name will contain the analysis's name for easier comparison (Figure 4).

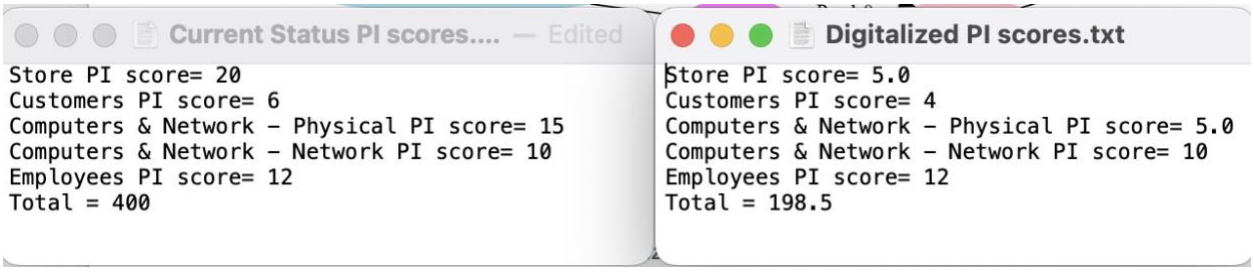


Figure 4 Comparison of the PI scores of the current status with the digitalized business.

Linter Evaluation

Pylint was used for static code analysis. The output was as shown in Figure 5. I had no choice of using too many loops and branches due to the nature of the application. Threat profiles are all about routes and branches, so I had to ignore the Pylint result. The code received a rating of 9.07/10.

```
************* Module threat_profile
threat_profile.py:149:4: W0702: No exception type(s) specified (bare-except)
threat_profile.py:60:4: R1702: Too many nested blocks (6/5) (too-many-nested-blocks)
threat_profile.py:60:4: R1702: Too many nested blocks (9/5) (too-many-nested-blocks)
threat_profile.py:60:4: R1702: Too many nested blocks (8/5) (too-many-nested-blocks)
threat_profile.py:45:0: R0912: Too many branches (18/12) (too-many-branches)
threat_profile.py:45:0: R1710: Either all return statements in a function should return an expression, or none of them
should. (inconsistent-return-statements)
threat_profile.py:160:4: C0103: Constant name "total_pi_score" doesn't conform to UPPER_CASE naming style
(invalid-name)
threat_profile.py:184:8: C0103: Constant name "line" doesn't conform to UPPER_CASE naming style (invalid-name)
```

Figure 5 Pylint output.

GitHub repository

The project GitHub repository can be viewed by clicking on this link.

README

A readme file was provided that contains all the information needed to download the code, setup up the environment, run the code with the sample data and view the output. The readme file can be viewed by clicking on this link.

References

Alberts, C. & Dorofee, A. (n.d) OCTAVE SM* Threat Profiles. Available from: http://130.18.86.27/faculty/warkentin/SecurityPapers/Merrill/AlbertsDorofee_OCTAVEThreatProfiles.pdf [Accessed 27 November 2022].

Alberts, C., Dorofee, A. & Stevens, J. (2005) OCTAVE ® -S Implementation Guide, Version 1.0. Available from: https://resources.sei.cmu.edu/asset_files/Handbook/2005_002_001_14273.pdf [Accessed 27 November 2022].

P. M. Training (2022) Simple Risk Assessment Matrix Template & Excel Example. Available from: https://pm-training.net/risk-assessment-matrix/ [Accessed 28 November 2022].

The Graphviz Authors (n.d) Graphviz. Available from: https://graphviz.org [Accessed 18 December 2022].