SEG 3125 (Analysis and Design of User Interfaces)
Winter 2019
Lab 2 - APT Tracking Platform

February 12th to March 4th
Work in groups of two or individually

Introduction

For this lab, you will create an APT (advanced persistent threat) Tracking website prototype. The website must allow threat analysts to:

1. Display information about APTs. This is called an attack matrix.
2. Add information to an existing APT
3. Create a new APT including common names, alternate names, TTPs, targets, toolsets, modus operandi, references.
4. Add comments (threat analyst comments) for any of the APTs currently tracked.
5. Search for any type information (by APT, by technique, by modus operandi, among other).
6. (**Bonus**) Create Visualizations to correlate information. Ex. How many APTs use PowerShell exploits in their arsenal.

Website Pages Description

The number and structure of the pages that constitute your website is left to your discretion. Nonetheless, to give you some ideas on how to start, the following is a description of the main pages of your website.

Page	**Description**
Matrix Page	This page displays the attack matrix. See the following for more details: https://attack.mitre.org/matrices/enterprise/ It should allow the user to modify the information displayed. A grid view should be suitable for this purpose.
Add a new APT Page	This page allows to enter the following information about an APT: • Common Names • Alternate Names • Target Sector • Attack Vector • Associated Malware • TTPs (tactics, techniques, and procedures) • Modus Operandi • References • Toolsets

Search Page	This page allows the threat analyst to search for any desired piece of information (APT name, technique, tool, etc.).

The descriptions above ***possibly*** do not summarize all the pages on the website. Nor should they restrict your creativity in designing your website prototype. They are merely suggestions. You are free to structure the website in any way you want as long as you include all the functions described in the **Introduction**.

Implementation

Obviously, since this is a prototype and you are not expected to implement a fully functional website (running on a web server with a database to store the patient's information and their follow-ups). This is a user interface course and therefore, the focus is on the design of the user interface. Consequently, at the very least, this is what you have to do:

- You must build an HTML website that showcases your user interface design.
- You must support dynamically generated **error messages**. You should handle all the possible error scenarios, especially when it comes to entering information into a form (e.g. incorrect format, missing mandatory field…). It is your task to identify these error scenarios and handle them gracefully. Javascript (or any Javascript based APIs) can be used to generate these messages.

This is what you do **not** have to do:

- Any content that is supposed to be pulled from a database can be coded statically into the website. Therefore, when the user searches for something, the Search Page will always show you the same information regardless of your search query. This is a mockup of the search function.

Evaluation

This lab is not intended as an exercise in website building. That is why we have drastically simplified the task of implementing the website prototype. Nonetheless, your focus should be on your user interface design. You are expected to apply the design principles we have learned in class. Pay particular attention to consistency (e.g. fonts and colour schemes), icon design, use of metaphors, general page layout (maybe you can apply some of the patterns seen in class), error prevention mechanisms and user feedback. It is strongly recommended that you produce paper sketches for your first design ideas. Evolve these sketches into the eventual website using the procedure studied in class. For the demo, you do not have to show your paper sketches or storyboards. Simply present the completed prototype.

This is the marking scheme:

- Website supports all the functions specified in the Introduction (40 points)
- User interface is well designed and follows the Heuristic principles seen in class (30 points)
- TA questions are answered correctly (30 points)

Frequently Asked Questions

What data should be used?

- You can use the JSON file provided (in the Brightspace).

What to submit?

You will simply show a demo of your application to the TA. The TA will also ask you questions about your design decisions. You still need to submit your application but it will not be graded (this is just in case you are not satisfied with the grade given by the TA during the demo).

When can I demo?

You have four lab sessions to complete and demo your application. Demos are performed on a first come first serve basis. Therefore, the TA might have lots of demos to go through on the very last session. Consequently, if you finish early, it is advisable to demo your work immediately and not way until the end.