FAF.CS16.1 Fall 2021

Lab 1: Importing Compliance Audit Policies

Handed out: Tuesday, September 3, 2021
Due: Tuesday, September 10, 2021 (20:15)

Introduction

In this course you will do a sequence of labs, the end goal of which is to develop a Security Benchmarking Tool (SBT). These labs will familiarize you with instruments used in IT security departments of different companies as well as give you practical experience in developing them. Concretely, you will be working with cyber security audit policies - benchmarks that allow a security officer to assess a system's vulnerability to different attacks and enforce certain security configurations based on community best practices.

Getting started

The SBT you will develop should be a local desktop application. As such, you are not constrained in choosing the platform / framework you will be using for this lab. It is okay to choose a framework that you don't know as long as you are ready to spend the time learning it. There are plenty of choices for many programming languages, so spend the time reading about some of them. In the end, the tool that you've chosen must help you with creating a modular and easily deployable application. You have to avoid developing an application that requires complex framework dependencies to run.

Importing Audit Policies

Some security companies (like CIS or Tenable) provide a set of audit policies that can be downloaded and used locally to test whether your system is set up properly to fight against common attacks and vulnerabilities. In this lab you will focus on downloading and persisting said policies. Below are some links to get you started. You'll need to familiarize yourself with the types of policies each company provides and choose one for your further work.

- https://www.tenable.com/downloads/download-all-compliance-audit-files
- https://www.cisecurity.org/benchmark/microsoft_windows_desktop/

Your task is to find a set of policies suitable for the environment you will be programming in (i.e. Win, Linux, MacOS etc.) and download them. Then, you'll need to provide means of persisting the imported policies i.e. parsing and saving them locally into a structured form (ex: database). It should be possible to upload the same policies multiple times and save them locally under different custom names.

To summarize, your application must be able to:

- Import the manually downloaded policies from a predefined trusted location;
- Parse and understand the format of data within the imported policy;
- Save the same set of policies under a different name within a structured form (ex: database).

Reporting

At the end of this lab, you will need to present the source code and a video screen recording of the functionality that you have implemented. The video recording and source code must be uploaded on Moodle, in the Submit Lab 1 assignment activity. Don't forget to make your code public on any hosting service of your choosing (e.g. Github, Bitbucket etc.). Any code on Github must contain a *readme* file (here's a tutorial on how to make a good one).

Grading

At the end of this lab you are expected to provide a minimum viable product (MVP) that would contain the features described in the previous chapter, alongside a simple Graphical User Interface (GUI) to use those features. Showing the working features in a terminal or a GUI with placeholder buttons is also acceptable, for a medium penalty. However, be aware that you'll still need to implement whatever you skipped for the following labs. What is not acceptable is not providing anything on the day of the deadline, so don't do that.

Future labs

Next lab will cover the implementation of a feature that would assess the local workstation based on the imported audit policy, outputting on screen the audit scan result.

Good Luck!