FAF.CS16.1 Fall 2021

Lab 2: Creating Custom Audit Policies

Handed out: Friday, September 17, 2021
Due: Friday, September 24, 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.

Creating Custom Audit Policies

The audit policies provided by CIS or Tenable usually have hundreds of settings that can be checked and enforced. At times however, a user might choose not to enforce all options, because some of them were already tailored to the systems' needs. In this lab you will implement features that would allow a user to select a subset of options that they would like to use, save that subset in a local database or any other structured form, and finally, export them as an audit file. Since scrolling through hundreds of options can be cumbersome, a search bar with which the user could filter the list of options could greatly improve the usability of the system (e.g. searching for the term "password" should return all options linked to passwords). When the user is happy with their choice, they must be able to save the chosen options within the application (i.e. a "Save" option), create a new audit policy (i.e. a "Save As" option) or export their selection into a new .audit file (i.e. a "Export" option). To summarize, your application must enable the user to:

- Choose which options they would like to run (by selecting or deselecting options);
- Search by name for an option (via a search bar);
- Select or deselect all options in one click;
- Create and save a policy that contains only the selected options under the same name or a different one.

Reporting

At the end of this lab, you will need to present your source code and a screen recording of the functionality that you have implemented. The links to your video and source code must

be uploaded on Moodle, in the Submit Lab 2 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 an application that would contain the features described in the previous chapter, alongside all features described in previous labs. Showing working features in a terminal or a GUI with only a subset of features and placeholder buttons is also acceptable, for a 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

In the following labs you will implement features that would allow the user, given a policy, to perform an audit of a workstation's settings, show the audit scan results and enforce the policy on the settings that are not set properly.

Good Luck!