Listen ▶

Module Five

Learning Objective

By the end of this module, you will meet this learning objective:

☑ Describe measures used to protect operating platforms from security threats

Module Overview

From an application developer's perspective, while using higher-level languages such as Java, C#, or Python, an operating system provides services that the application needs. These services include memory management, storage management, process execution, exception handling, security, networking services, and so on. Today, technology is moving from the physical world where companies purchase hardware servers and licensed platform software like operating systems, databases, and web servers, to cloud services where they pay for what they actually need and use. This shift can provide significant cost savings, increasingly faster time to market, and access to a greater variety of prebuilt services including several kinds of relational databases and caching services. The possibilities are expanding rapidly, but so is the complexity of weaving together all of these technologies.

The work you have done so far and the information gained regarding creating classes with attributes (data) and behaviors (methods) can also be applied at a higher level of scope to design the interactions between services exposed and running in a distributed environment. For example, your applications could send notifications of their status as tweets, load images from imgur, display a Spotify playlist, control smart devices around the home, office, and vehicle, or use two-factor authentication with Google Authenticate.

When thinking about how to design services to represent the behaviors your application wishes to make available to a web service, think about it as you would for creating methods on a class. Just like using design patterns to design classes, there are architectural patterns to design and organize services so they can interact with one another. The principles you have learned making classes and methods interact together are the same principles needed to make applications work together. This module will explore adding capabilities to allow a login request to be handled - a necessary requirement in nearly all applications that you will develop and work on in your career.

Login authentication is only one aspect of security. Permission to perform actions, known as authorization, is equally important. Project Two will give you exposure to securing REST endpoints, known as resources in security terms. Menu options in a user interface are another form of resource to secure. From an application security perspective, any action, whether an API call or a user-initiated one, needs to be protected.

Module at a Glance

This is the recommended plan for completing the reading assignments and activities within the module. Additional information can be found in the module Resources section and on the module table of contents page.

1. Review the Module Five resources.
2. Post your initial response to this week’s discussion.
3. Submit Project Two.

4 Post peer responses to the discussion.