Authentication and Authorization

Code security protects the normal, day-to-day operations of an app, tool, or daemon. But what happens when your code is under siege? It is often essential to know not only what the user is doing but also who the user is and whether the user is allowed to do that. This is where authentication and authorization come into play.

Authentication

"If you know yourself but not your enemy, for every victory gained you will also suffer a defeat."
—Sun Tzu, *The Art of War*

When securing software, the first thing you must do is find a way to distinguish friend from foe. This process is called *authentication*.

In computer security, authentication verifies the identity of a user or service. Authentication usually serves one of two purposes:

- As a precursor to authorization, identifying the requesting entity to determine whether that entity should have permission to perform an operation
- For producing an audit trail by logging who performed an operation so that blame can be cast when something breaks

Three types of authentication are most common:

- **Local user authentication.** Verifying a user's identity is usually performed by the operating system as the first step in authorization. If your code is running as a normal user, the operating system limits what your code can do based on that user's permissions. Your code can also ask the operating system for the identity of the user for auditing purposes.
- **Network host authentication.** Verifying the authenticity of a remote server is often necessary—for example, to determine whether it is safe to send credit card information to a specific website. (Digital certificates, described in the next chapter, are a common way to achieve this.)
- **Remote user authentication.** Users are often authenticated by remote servers when performing certain tasks. Authenticating a user remotely requires that your code send credentials in some form, such as a password, a cookie, or a digital certificate.

Authorization

Authorization is the process by which an entity such as a user or a server gets permission to perform a restricted operation. The term is also often used to refer to the right itself, as in "The soldier has authorization to enter the command bunker."

The difference between authentication and authorization is somewhat subtle. Often, the mere fact that a user has an account means that the user is authorized to do something, in which case authentication and authorization are the same thing. However, in more complex systems, the difference becomes more obvious.

Consider a computer with two users. Each user is known to the system. Therefore, both users can each log in to the computer, and it authenticates them. However, neither user is authorized to modify the other's files, and as a result, neither user can do so.

The details of authorization depend on whether you are using iOS or macOS.

In iOS, the user can set a passcode (which by default is a four-digit personal identification number) to prevent unauthorized use of the device. After entering this passcode, the user of the device is presumed to be authorized to use the device. In addition, each app is digitally signed and can therefore be authenticated by the operating system. Therefore, there are no user authentication or authorization APIs in iOS.

In macOS, there are several layers of authorization:

- If FileVault 2 (full-disk encryption) is enabled, the computer requires a password to decrypt the boot volume.
- If automatic login is disabled, macOS displays a login screen after booting.
- macOS also displays a login screen when the user logs out.

- If the appropriate checkbox in the Security system preferences pane is checked, macOS displays a login screen when waking from sleep or when leaving a screen saver.
- When an app or tool requests access to a locked keychain, a password is required.
- If an app or tool needs elevated privileges, an administrator password is required.
- Some apps may restrict access to parts of their functionality through the Authorization Services API.

In addition, on both macOS and iOS, some apps may require you to log in to a remote server, which in turn performs authentication and authorization.

To Learn More

For a more detailed conceptual overview of authentication and authorization in macOS, read *Authentication, Authorization, and Permissions Guide*.

You can also learn about other Apple and third-party security books in Other Security Resources.