User Interface and Security

CST-221 : Operating Systems Concepts

Isaac Tucker

07/18/2020

**User Interface and Security**

**User Interface for 3 login types**

Here we see the login screen for entry into a system. The single-factor authentication simply requires a username and password. The two-factor authentication requires username, password and an authentication code provided by a security device (key fob). The multi-factor authentication requires a username, password, authentication code provided by a security token and a photo of one’s face for FaceID authentication.

**A screenshot of a cell phone

Description automatically generated**

**Single-Factor Authentication**

In the first UI we are utilizing single-factor authentication by requesting something a user knows. This is the form that most individuals are familiar with as it is the normal username and password combination; a user enters their username and password to gain access to a system. This has been the standard for systems since conception but as time has progressed the standards for the authentication key, or password, have become more strict. The necessary policies include validation of the username, validation of the password, password entry timeout, unique character processing, password encryption.

Requirements:

Username is unique.

Password must be longer than 8 characters.

Password must contain at least one upper-case and lower-case letter.

Password must include special characters, such as @, #, $

Password must include one or more numerical digits.

Password must not use words found within a dictionary.

Password must not use words found in user’s personal information (ie name).

Password must not be company name or abbreviation.

Password must not match calendar dates, telephone numbers, or other publicly available information.

Passwords must never be stored or transmitted as plaintext.

Passwords must be encrypted with salts for authentication purposes.

When typed, password field is left blank.

Previously used passwords are unacceptable.

**Single-Factor Login Flow Chart:**

A close up of a map

Description automatically generated

In the above flowchart we see the typical flow of single-factor authentication. It is important to note that we are not informing the user who attempts to log in about the password expectations. Because of this most of the password requirements remain hidden at login; they are only disclosed at account creation. As such they are not included within the login flowchart above.

**Two-Factor Authentication**

In the second UI we are building upon the single-factor authentication and adding the use of an authenticator - also known as Time-Based One-Time Passwords. This effectively makes hacking a system far more complex as a user needs to know special information and also possess a special item to gain access to a system. Authenticators are common today in form of key fobs (security token) or special applications one downloads on an external device. There are different approaches on how these devices operate but essentially they have a shared secret, or key, between the device and the company’s server which makes use of the time of the request and a hash algorithm to generate a unique identification code. Many organizations like Blizzard, the Department of Education, Steam, and Square-Enix make use of these devices to help ensure only those actual access can login into the respective systems. At login a user will query their device for a special code that only lasts about 10 seconds. The user must successfully enter in their username, password, and authentication code within that time limit to ensure they can obtain access.

Requirements:

All of the one-factor requirements for Username and Password.

Security token has a shared key that is only known to the device and the server.

When queried, the provided code is only available for 10 seconds.

Device is with the sole possession of the user and device is unique.

**Two-Factor Login Flowchart**

**A close up of a map

Description automatically generated**

The above chart will work a lot like the one-factor authentication except there is extra steps to both input/send and authenticate the one-time password from the security token. I put in the flowchart that the security code is also hashed and salted but this may vary per organization. It is a good idea to hash the code, however, as hackers can figure out the formulas for the one-time password given enough data. If this happens the extra authentication method is essentially useless. While the one-time password is in use, however, it helps avoid brute force and dictionary exploits as knowing a static password alone is insufficient to gain access.

**Multi-Factor Authentication**

In the third UI we are building upon the single and double-factor authentication and adding the use of software known as FaceID to utilize multi-factored authentication. FaceID is a facial recognition program developed by Apple which, when taking a photo of one’s face, projects infrared dots and returns a pattern to the device. This pattern is then securely stored by the CPU for later authentication purposes. This additional factor ties into the current factors, an item a user knows (password) and the item a user has (security token), by utilizing something the user is (their face). This helps because even if a hacker can replicate the math for a one-time password, and even if they decrypted their static password, it is hard to replicate the complexity of the human face.

Requirements:

All of the above one and two factor authentication requirements.

Device requires a camera with enough pixel depth for FaceID to function.

Upon setup, scan face from number of angles to create complete reference map.

FaceID needs to be able to match expected aging, hair growth and other changes.

Infrared needs to be bright enough to work in dark locations.

FaceID will recognize users wearing hats, scarves, glasses, makeup, etc.

FaceID scans are to be encrypted and securely stored.

**Multi-Factor Login Authentication Flowchart**

A close up of a map

Description automatically generated

One aspect different with this authentication is that FaceID is stored on the local device in a secure location (though it does not have to be). There is added benefit to not always sending data on a network to some server for validation; hackers who can only obtain information via outgoing packets will not be able to detect the information stored locally.