MIT 6.858 - Computer Systems Security "Lecture 3 - User Authentication"

Lecture 3 - User Authentication

User authentication is an important problem which has a variety of issues that are hard to deal with. Even though the premise of the problem is not complicated, user authentication continues to be challenging because it is not just a technical problem. All schemes we have come up with for user authentication so far are imperfect because the end users are ultimately human and imperfect.

The bigger problem of user authentication can be broken down into three separate sub problems - 1. User registration - The process of registering the identity of a user with a service so that the identity can be verified in the future. This sets up a shared secret between the user and the service (usually a password). 2. Identity verification - The process of authenticating a user and verifying their identity. This process verifies the shared secret. 3. Recovery - The process of recovering the secret if the user loses it. If overlooked, this step can undermine the security of the first two steps, no matter how strong they are.

Identity Verification

The most common way of verifying an identity is a password. The benefit of passwords as a shared secret is that they are easy to remember and very convenient to use. However, the challenge with passwords is that the users of passwords are human, and they tend to use passwords in a way that makes them weaker, such as by reusing passwords, or by using weak or easy to guess passwords.

Passwords are never stored in plaintext. They are combined with a large random number called a salt, and hashed using a cryptographic hash function. The service then stores tuples containing `(username, salt, Hash(salt || password))`. The hash function is also purposefully made to be slow, to make it harder for attackers to brute force passwords.

However, even if passwords can be imperfect, their security can be greatly improved by augmenting them with two-factor authentication. You are already familiar with

common ways of introducing two-factor authentication such as OTPs through SMS or an authentication app. These are all good approaches, but they don't protect against MITM attacks and phishing attacks.

Universal Two Factor Protocol (U2F)

The U2F protocol being developed by Google is a form of two-factor authentication which is resistant to phishing and even MITM attacks. It uses public-key encryption and a dedicated 2FA device to verify a user's identity.

It is a great improvement over OTPs over SMS or authentication apps, but it still makes some assumptions which make it vulnerable in some extremely rare situations.

During authentication, the browser sends a login request to the server. The server sends back a "challenge", which is a long random number, for the U2F device to sign using its private key. The user plugs in the U2F device via USB, which receives the challenge, signs it using its private key, and sends it back to the browser. The browser forwards the signed challenge to the server. The server can now verify if the U2F device's identity using its public key.

This is the basic premise of the protocol. To protect against MITM and phishing attacks, the browser actually forwards not just the server's challenge, but also the origin and the TLS connection ID. The U2F device then verifies the signature for all three instead of just the challenge.