Lec 4

Secure Coding And Mobile Authentication

- **Secure Coding**
- **Secure Coding Practices For Mobile App Development**
- **Mobile Authentication**
- **Types Of Mobile Authentication**
- **Best Practices for Implementing Mobile Authentication**

1. Secure Coding

secure coding is the practice of writing software that's protected from security vulnerabilities.

It involves adhering to coding standards and best practices designed to minimize the risk of cyberattacks, unauthorized access, and data leaks. When you build a mobile app, every piece of code becomes a potential entry point for hackers if it's not secure.

When your code is secure, it's harder for hackers to steal data, take control of the app, and compromise user trust. It's about building a strong foundation so your app stays safe from potential threats.

Risk comes from using the wrong inputs, hardcoding sensitive data, using an insecure API, not handling errors and logs properly, using insecure third parties, or some other code-level issue. Importantly, this is not a piece of code running on the server; it is code running directly on a mobile device.

Mobile application security refers to the measures taken to protect mobile apps from cyber threats, ensuring data privacy, secure communication, and preventing unauthorized access.

1.1. Secure Coding Practices For Mobile App Development

1. **Validate all user input**

One of the simplest yet most effective secure coding practices involves validating all user input. Developers should check and sanitize any data users submit to the app whether through forms or file uploads before processing it.

This practice prevents attacks like SQL injection and cross-site scripting (XSS), where hackers inject malicious code into input fields. It ensures that only legitimate data enters the system, protecting both the app and its users.

2. **Encrypt sensitive data**

Encryption is a must-have when it comes to securing mobile apps. Encryption ensures that sensitive data such as passwords, credit card numbers, and personal information is scrambled and unreadable to anyone who doesn't have the proper decryption key.

For developers, encrypting data both at rest (stored data) and in transit (data being transferred between systems) is essential. This practice makes it much harder for attackers to steal or misuse the data, even if they gain unauthorized access.

3. **Use strong authentication**

Authentication is the process of verifying that a user is who they claim to be. Weak authentication methods make it easy for attackers to gain unauthorized access to your app's data and features. The organization should implement strong authentication methods like multi-factor authentication (MFA) to ensure that only authorized users can access the app.

It's also an important feature that users now expect from secure apps, particularly in finance, healthcare, and e-commerce sectors.

4. **Avoid hardcoding secrets**

Hardcoding sensitive information like API keys, database credentials, or passwords into the source code is a major security risk.

If hackers gain access to the code, they also gain access to this critical information. Developers should store sensitive information in secure, encrypted environments rather than embedding it directly into the app's code.

5. **Conduct regular code reviews**

Even the most skilled developers can overlook potential vulnerabilities. This is why regular code reviews are essential. Code reviews involve having multiple developers review the same code to spot issues that might have been missed.

Automated tools can help detect common issues, but manual reviews are necessary for identifying more complex vulnerabilities.

By embedding a culture of security-conscious coding, companies reduce their risk of deploying vulnerable apps.

6. Test your app for vulnerabilities

No matter how carefully developers write code, vulnerabilities can still exist. Penetration testing and regular security audits help identify these hidden vulnerabilities by simulating real-world attacks.

By testing the app's security before it's launched, developers can fix issues proactively rather than waiting for a breach to occur.

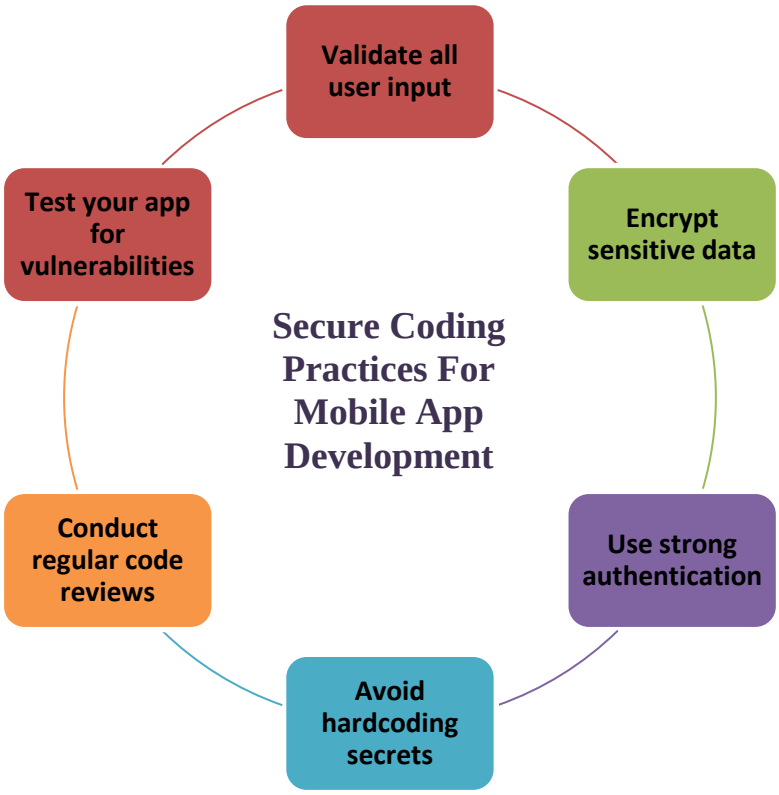

Figure (1) Secure Coding Practices For Mobile App Development

2. Mobile Authentication

Mobile authentication is the process of verifying a customer's identity on their mobile device before granting them access to their accounts and data.

The ability to provide secure mobile access to customer accounts is critical for any organization that conducts business or offers services online.

Enabling customers to access their accounts through whichever device they are using in the moment has become an important part of the overall customer experience that no company can afford to ignore.

Mobile authentication factors fall into three categories:

- Something the user knows, such as a password or PIN (knowledge-based)
- Something the user has, such as a four-number code sent to their phone (possession-based)
- Something the user is, such as a scan of their fingerprint or face (biometric-based)

2.1. Types Of Mobile Authentication

1. Pattern and Digit-Based Authentication

Pattern and digit-based authentication describes two options that can be used to unlock devices and accounts. Pattern-based authentication asks the user to use their finger to draw a pattern on their screen. For digit-based authentication, the user is asked to enter a four- or six-digit PIN.

These factors offer the advantage of speed and ease since either is easier to enter than a password. In practice, though, users typically choose simple patterns, such as an S or an L, or easy pins, such as 1234 or their birthday, which are easily guessed by hackers.

2. Password-Based Authentication

One of the most common factors used for mobile authentication is the password, typically used in conjunction with the customer's email address or username to confirm their identity.

Passwords provide varying degrees of ease of use and security. When users can establish their own passwords without any limitations, they tend to choose things that are easy to remember.

Customer-created passwords are usually simple, with the minimum number of characters required, and contain personal information such as significant dates. Many times, the same passwords are used by a customer for multiple accounts.

3. One-Time Password (OTP) Authentication

A one-time password is a code sent to the customer's mobile device via SMS or email when they try to log into their account.

The code is automatically generated and only valid for a limited period of time, usually for only a few minutes. Depending on how the business has established its security, the OTP can be used in conjunction with or in lieu of a traditional password.

4. Social-Based Login

Social-based login enables customers to gain access to their account using the credentials they've already established with a social networking site such as Google, Facebook, Twitter, or LinkedIn.

5. Mobile Biometric Authentication

Mobile biometric authentication enables customers to use a unique physical feature to access their accounts and data. Commonly used biometric factors include fingerprints, facial scans, and voice recognition.

6. Multi-factor authentication (MFA)

Multi-factor authentication requires at least two authentication factors to grant user access. Such a combination of, for example, a password and a fingerprint highly increases security.

The factors include something the user knows (password, PIN, etc.), something the user has (phone, security key, etc.), and something the user is (biometric authentication).

MFA is much more secure than single-factor authentication. Even if the password is leaked, the second factor is still necessary.

MFA is often used by bank apps.

Some banks require two-factor authentication (2FA) to log into the app or confirm a transaction. So, apart from using a PIN or passcode, the user has to enter a security code sent via text message.

7. Location-based authentication

Location-based authentication uses the device's location to verify one's identity. The device may use GPS, Wi-Fi networks, or trusted locations to determine if user activity is suspicious. This, however, requires the user to share their location details with apps.

This authentication method is a great way to add another layer of security to traditional ones. It may be problematic to verify one's identity from every new location, but this prevents fraud risk.

In the case someone uses your credentials in a new location, you are informed about it and can block the access.

For example, if someone tries to use a stolen credit card in a new location, the bank may flag the transaction and block it.

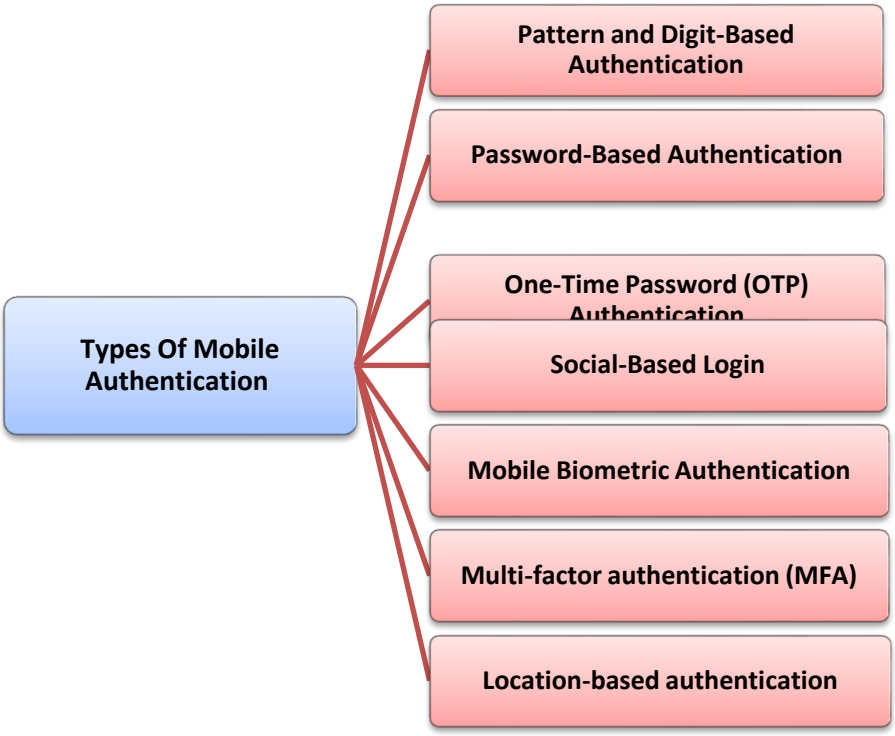

Figure (2) Types Of Mobile Authentication

2.2. Best Practices for Implementing Mobile Authentication

1- Focus on User Experience (UX): Customers should be able to open and access their accounts with minimum of friction. Factors that increase friction can be different on mobile devices vs. on computers.

For example, a password that can be easily typed on a laptop keyboard may become frustrating on a mobile phone keypad. And an OTP that requires the computer user to access their mobile phone or go check their email, could be easily pulled from an SMS on a mobile device.

User experience begins with the process to open an account or register on a site. The longer the registration form, the more frustrating the experience is; the form should only request information that's truly needed.

Another way to simplify UX is by allowing users to remain logged into a mobile app after they have authenticated their identity.

2- Leverage Two-Factor Authentication (2FA): As the name implies, mobile two-factor authentication increases security by requiring a combination of two authentication factors before access is granted.

A common use case: once a customer has entered a password, they are asked to also enter an OTP that has been sent to them via SMS.

Two-factor authentication decreases the likelihood that the attempted entry is being made by a fraudster.

By combining something the customer knows, like a password that's vulnerable to being stolen, with something they have a physical phone or tablet that a fraudster is unlikely to steal or possess 2FA provides an added level of account security.

Two-factor authentication can also be leveraged by use case. For example, instead of being required every time the customer opens the app with the phone they always use, 2FA can be used to authenticate their identity when the customer accesses their account from a new or unrecognized device.

3- Use Passwordless Authentication: It's also possible to take passwords and their vulnerabilities out of the authentication equation completely.

Passwordless authentication, also called passkey authentication, relies on PINs, patterns, and biometrics, with biometrics providing the highest level of security. Passwordless authentication both improves security and offers easy usability for customers.