Khaula Molapo

Computer Security week 6

1.

A banking application requires strong authentication to ensure customer security. When a user opens the mobile app, they must log in using their username and password. The system first checks the password against the stored hash in the authentication server. If it matches, the server generates a session token that allows the user to proceed. However, because financial data is sensitive, the bank also enables an additional step: a one-time password (OTP) sent to the user's phone via SMS or email. The user must enter this code to complete the login process. This second factor reduces the risk of unauthorized access, even if the password is stolen. Once authenticated, the user can securely check balances, transfer money, and pay bills. The authentication process ensures confidentiality, integrity, and trust between the user and the bank, while balancing ease of use and strong protection against cyber threats.

2.

Multi-Factor Authentication (MFA) is a security process that requires users to verify their identity using more than one factor. The common factors are: something you know (password or PIN), something you have (smartphone, security token), and something you are (fingerprint, face recognition). By combining at least two of these, MFA makes it much harder for attackers to gain access, even if one factor is compromised. MFA is widely used in banking, enterprise systems, and personal accounts to prevent unauthorized logins. Its role is to strengthen authentication, reduce identity theft, and provide greater protection for sensitive data and services.

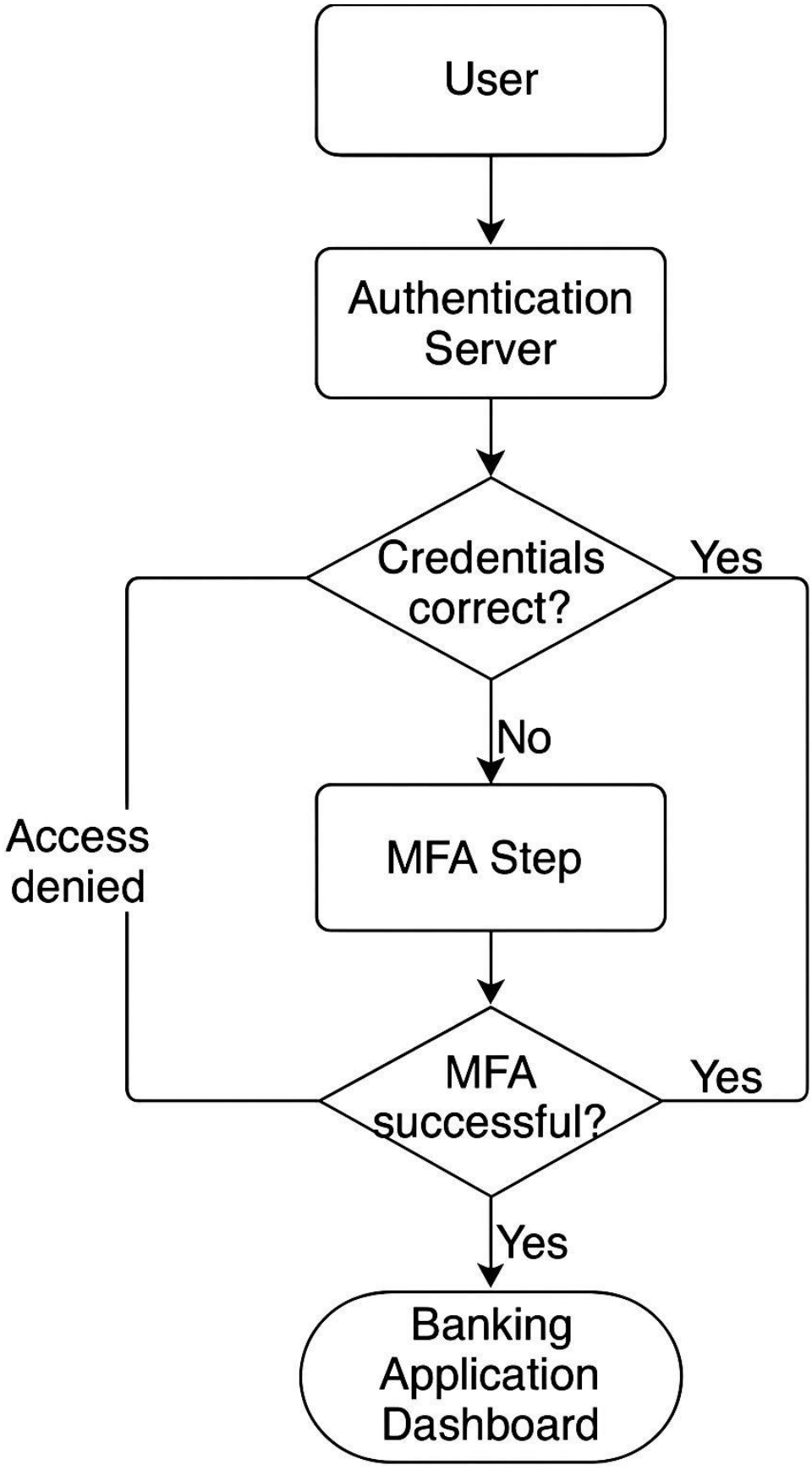
3.

In Auth0, I practiced setting up Multi-Factor Authentication for a sample application. First, I enabled the MFA option in the dashboard and chose OTP through an authenticator app. After configuration, the system required both a password and the generated code before login was successful. I tested the setup by creating a test user and logging in, and I noticed that the second factor worked smoothly. This exercise showed me how MFA can be added

with minimal setup while greatly improving security. It reinforced how important it is for modern applications to use MFA to protect accounts against breaches.

4.

In Draw.io, I designed a simple authentication flowchart for a login process. The flow begins with the User entering their credentials. The request is sent to the Authentication Server, which verifies the username and password against stored data. If the credentials are incorrect, the system denies access. If correct, the flow continues to the MFA Step, where the user enters an OTP or biometric factor. The server validates this second factor. On success, access is granted, and the user reaches the Banking Application Dashboard. This flowchart highlights how authentication and MFA together build layered security and trust.


User
Authentication Server
Credentials correct?
Yes
No
Access denied
MFA Step
MFA successful?
Yes
Yes
Banking Application Dashboard