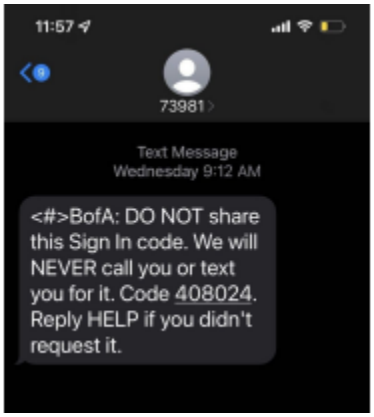
CAS CS 357

InClass Note 16

1. Why do we need an MFA(multi factor authentication)?
    a. Servers provide certificate to clients
    b. Clients provide password to servers
    c. Passwords become leaked/stolen
    d. Ex) Facebook passwords compromised: Meta warns 1M users may have login information stolen
    e. Use multi factor authentication to protect adversaries from stealing password
2. Authentication based MFA
    a. 

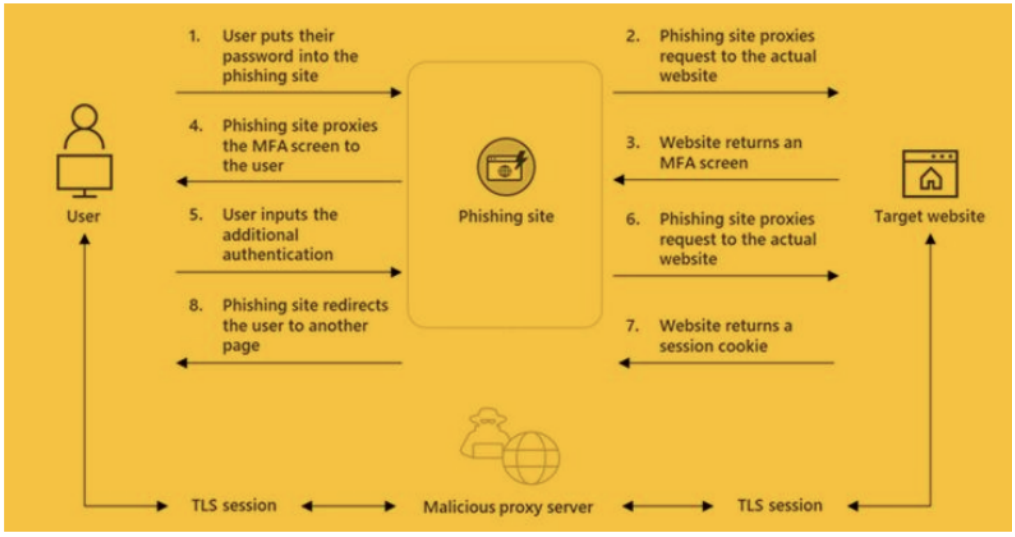
3. MFA based on SMS
    a. 

b. Why do this?
    - Usability. Most people have a phone and know how to text
    - Codes are short lived, so if stolen they must be used immediately
    - We do this → it works and most people know how to do it (it is used famously)

c. Problems:
    - Sim swapping attacks (when someone gets your phone number moved to a SIM card that's in a phone they control) → steal phone number of user so that adversaries receive the text message and the email of users to lock people out of their accounts
    - SS7 attacks (more obscure) → SS7 is protocol that controls phone communication, adversary can attack the SS7 protocol
    - Phishing attacks (adversary tricks you into sending them the code on your phone)
    - MFA notification fatigue attacks
        a. Adversary continues to send you the notification (phone call or hitting push) until the user finally accepts it by mistake

4. MFA based on Phone Call
    a. Why do this?
        - Usability: most people have a phone and know how to answer it
        - Codes are short lived, so if stolen they must be used immediately
    b. Problems:
        - Same as above

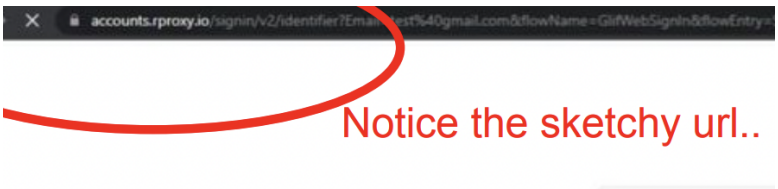
5. TOTP MFA Phishing Attack

a. 

b. Here, adversary does not compromise the target website

c. The issue is that the user got confused about the website and entered the information on the wrong website (entered the information on phishing site)

d. This attack does not require the certificate, cookie, etc. of the users since the user is giving all the information to a phishing website that interacts with the real website

e. 

f. The url is a phishing site that mimics gmail to trick users

6. TOTP(Time-based One Time password)

a. Computer algorithm that generates a one-time password using the current time as a source of uniqueness

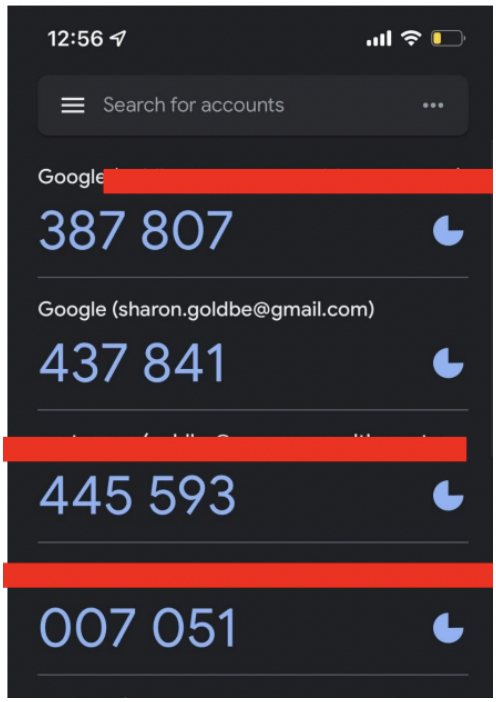
b. Google “authentication App” on your phone

c. Or one of these apps
    - 2FA authenticator
    - Aegis Authenticator
    - andOTP
    - Authy
    - FreeOTP
    - Google Authenticator
    - Microsoft Authenticator
    - TOTP Authenticator
    - Per-app authenticator apps
    - Other 2FA options

d. Codes are based on the key that is stored on the phone

e. Beneficial since the protocol can be used in many different apps

7. How does TOTP work?
    a. Let C be a counter where C = (Now-Time0)/30 seconds
    b. Let k be a 256-bit secret that is securely stored on your phone + on server
    c. Code = grab_6_digits_from($HMAC_k(C)$)
    d. Using symmetric key to connect the phone with server
    e. Convert the 256 bits into 6 digits shown above using the key and the counter

f. 

g. Server and code can generate the code and the user can tell

h. Adversary has to steal the actual physical phone to successfully launch the attack (in order to steal the key, they need to steal the phone)

8. TOTP MFA

a. Why use TOTP

- Codes are short lived, so if stolen they must be used immediately
- Uses your phone which is something you are not likely to lose
- The secret key k is stored securely on your phone & hard to steal
- No more Sim swapping and SS7 attacks

b. Problems

- Phishing attacks

9. Webauthn

a.

b. Developed for hardware keys

c. It has 2 parts:

   - Registration, where the authenticator creates a public key scoped to the web server
   - Authentication, where the actual MFA happens

d. Uses public key

e. ex)

   Alice Server (Google)

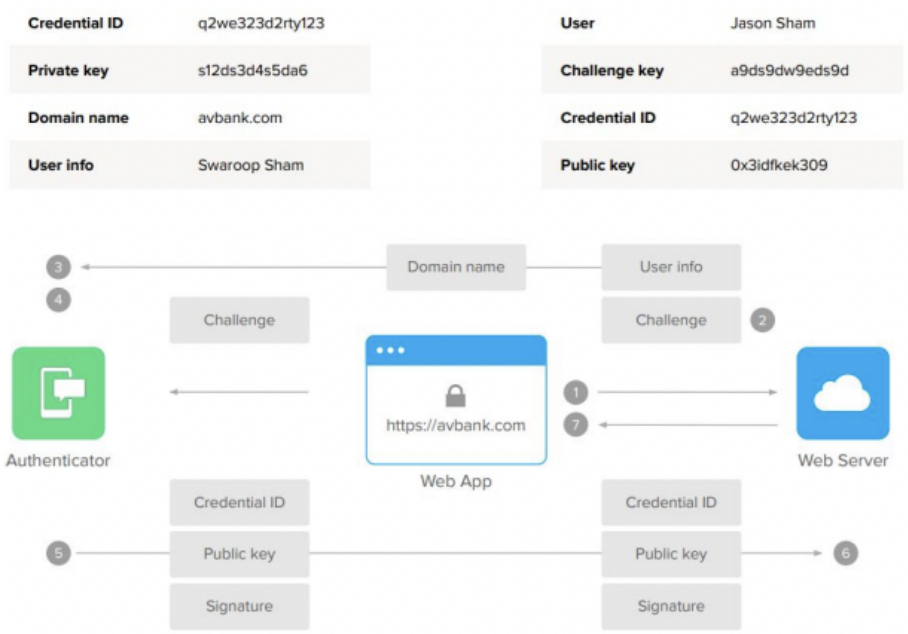
   Server (google) will send a challenge (information tied with a random number)

   Alice(browser) sets the domain d = google.com

   Alice creates o = $Sign_{sk}(d||c)$ and sends it to the server

   Google verifies the signature that is sent ($Ver_{pk}(d||c, o)$) = 1

10. Webauthn registration flow



a.

b. Steps:

- Browser sets the domain d as the web server
- The web server provides a challenge for the user (The challenge is randomly chosen by the web server)
- The browser sends the information of the user and the domain to the authenticator (not the server)
- Browser chooses a secret key and public key and signature and sends it to the web server

c. Web server stores public key

d. The Browser attaches the domain name to the challenge

e. Why the browser and not the web server? → due to phishing attack (the server can be a wrong server made by an adversary)

f. Signature → $Sign_{sk}$(user,domain,credID, challenge, pk) → we need signature to prove that the information being sent to the server is from the browser (not an adversary)