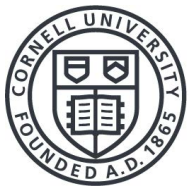

COURSE PROJECT

Authenticating Humans

Fred Schneider

Faculty Author
Cornell Bowers Computing and Information Science

Cornell University

Project Completed by: Germaine Wong

COURSE PROJECT — PART ONE

Authenticating With Passwords

1. The first step is to identify the human principals that need to be authenticated in your system. You must then decide which humans should be authenticated with passwords, and you should design a password-based authentication scheme for those humans. Be sure to consider how the passwords will be chosen, how the passwords will be stored, and how to recover forgotten passwords.

Principals to authenticate include the end users, a system administrator, and a third party auditor. In a secure, anonymous communication system the use of password-based authentication alone is insufficient because anonymity needs to be preserved and passwords that are stored can be leaked or stolen via phishing or spear phishing. The principals that would need to be authenticated with a password are the end users, system admin and auditor.

The password-based authentication scheme for an end user would look like the following:

1. end user enters their password

2. end user hashes password with salt

3. server gets salt that has been stored and compares hashes to see if they match

4. MFA

5. user ends session or session timeout

The password-based authentication scheme for a system administrator would look like the following:

1. admin enters their username and password

2. system hashes password and compares it to the hash in the database and if it matches then a hardware token is requested

3. MFA

4. admin manually logs out or system timeout occurs

The password-based authentication scheme for a third party auditor would look like the following:

1. auditor enters their username and password

2. password gets hashed and compared to the server's stored hash

3. system sends an OTP

4. MFA

5. when auditor logs out, the session token is revoked

There are several ways that passwords can be chosen. If it is self-chosen, a minumum length of 16-20 characters and complexity is needed for users, admins and auditors. For the system administrator the additional requirements are that it's not on the global banned password list or the includes any company references. Another way of choosing a password is the use of a password manager to generate random, strong passwords for each account which helps prevent the reuse of passwords, it avoids common patterns and substitutions (p@$$word), and uses random number generators which make the password unpredictable. Using pseudonymous accounts like Proton Mail can be used to register an account without revealing the user's personal identity.

There are a few ways that passwords can be stored, depending upon the person and system. One way is to use a password manager, which has many benefits, including: storage via AES-256 and no access to master password for the manager. Some popular ones include Proton Pass, Bitwarden, and 1Password. This is good for the end user as well as the third party auditor. The use of AES-256 to encrypt passwords for internal systems and databases is meant for system administrators and auditors alike, as well as the the use of RBAC for least privilege access for auditors.

Recovering forgotten passwords is based on the type of principal and the system used. For an end user, the recovery workflow follows these steps: 1. Provide their pseudonymous email address. 2. If this is their backup method, a one time password is sent (mainly for laptops/desktops). If using an app (on a phone or tablet) a recovery token can be sent via an E2EE message. 3. The user accesses their OTP and enters it, or they access the recovery token. 4. Secret question is asked. An end user can also store encrypted backups of passwords in a password manager. For a system administrator, the recovery workflow follows these steps: 1. Going to the "forgot password" section of the admin portal. 2. Recovery token sent through a

secure messaging channel. 3. Admin enters OTP, which immediately expires. 4. MFA process. 5. Admin generates new password. For an auditor, the recovery workflow follows these steps: 1. Navigate to the "recover password" page on a secure page specified for auditors. 2. OTP is sent via a secure messaging service. 3. Auditor enters the OTP. 4. MFA. 5. Auditor sets new password.

Facilitator's Comments

The facilitator did not supply a comment.

Grade

100%

COURSE PROJECT — PART TWO

Authenticating With Physical Tokens

1. You now need to decide which humans should be authenticated with physical tokens (including humans that will be authenticated using multifactor authentication). You should design a token-based authentication scheme for those humans. Be sure to consider how the humans get their physical token, what authentication protocol the token uses, and how to handle lost tokens. Feel free to revise some of your designs from Part One, if appropriate.

The human principals that need to be authenticated with physical tokens, including MFA, are the end user, the system administrator, and the third party auditor.

For an end user, there are two authentication schemes that work, depending upon what the user uses to authenticate - a physical key, and a digital token.

One token-based authentication scheme for an end user involves a digital token and works as follows:

1. The user installs the communication app on their phone/tablet and includes their phone number as part of the enrollment process.

2. The app sends a one time code to the phone to prove that the phone number belongs to the user.

3. The user enters the code received and is then authenticated.

4. The communication app creates a token that is stored on the user's phone/tablet.

5. Upon opening the communication app, the token is used to prove who you are without needing a password.

Another is the use of a physical YubiKey. The authentication scheme for an end user with a YubiKey is as follows:

1. The end user plugs in a YubiKey to their device which holds a private encryption key.

2. The application sends a challenge (r) to the token and the token then uses its secret key to digitally sign the challenge.

3. The system asks for and the user enters a PIN.

4. The system grants access and generates a temporary session token.

For a system administrator a Smart/PIV Card authentication scheme is the best option because it is used for authenticating into secure internal systems:

1. A card containing a private key and public key certificate signed by the organization's CA is issued to the admin. The card contains a PIN that only the admin knows.

2. The admin inserts their card into the workstation's card reader.

3. The admin then enters their PIN.

4. The workstation sends a challenge to the card (nonce).

5. The card signs the challenge using its private key.

6. The certificate is checked to see if it is trusted and still valid.

7. If so, the admin is granted access to the workstation.

The token based authentication scheme for a third party auditor involves the following:

1. The auditor is issued a digital credential signed by a trusted authority (system operator) which contains the role of auditor, scope of read-only logs and an expiration date/time.

2. The auditor connects over a secure channel and sends the signed access token plus proof of the valid credential.

3. The system verifies the credential signature, checks the access policy, and logs the attempted login.

4. If it checks out, the auditor is authenticated and can access the encrypted logs.

The way to get one of the two types (YubiKey, Smartcard) of physical tokens a person will need to buy it from trusted sites like yubico.com, and pivkey.com. To get an MFA authentication app for a phone/tablet a person would need to download their chosen authentication app from Google Play or Apple Store. Examples include Microsoft Authenticator, Google Authenticator, and Authy.

The two authentication protocols the different tokens use are: for the YubiKey - FIDO2/WebAuthn (cryptographic credentials); for an Authentication App - TOTP (time-based, one-time password).

The handling of lost tokens depends on the type of token. For a YubiKey, a person would need to login into their account via a backup YubiKey or through MFA and remove the key as an authentication method. For authentication apps, if you have a cloud backup, you can use that to perform account recovery. A person would need to sign into their account online to review and revoke sessions and remove the stolen device's name from being an authentication tool and then change passwords for the account and anything the authenticator was linked to. Lastly, a person would need to set up the authenticator app on a new device.

Facilitator's Comments

The facilitator did not supply a comment.

Grade

100%

COURSE PROJECT — PART THREE

Authenticating With Biometrics

1. The final step is to identify which humans should be authenticated with biometrics (including humans that will be authenticated using multifactor authentication). You will then design a biometric-based authentication scheme for those humans. Be sure to include what biometric will be used, how you will handle false-positives and false-negatives, and how you might handle humans who have objections to using your chosen biometric (if applicable). Feel free to revise some of your designs from Part One and Part Two, if appropriate.

The humans that need to be authenticated with biometrics that also include MFA are the end user, the system administrator, and the third party auditor.

Scheme for end users: 1. User goes to their communication system's login page or opens the communication app on their device and enters the username or account ID. 2. A challenge (nonce) is sent to the YubiKey Bio from the device for signing. 3. The user is prompted to place their finger on the YubiKey and this fingerprint is verified locally against the biometric template stored on the YubiKey. 4. If the fingerprints match, the device signs the challenge. 5. YubiKey Bio uses the private key stored on the device to sign the challenge received and only this signed challenge is sent to the authentication server. 6. The authentication server verifies the signed challenge using the public key of the pair. 7. If the signature matches the challenge, the authentication server grants access. This scheme helps with biometric privacy because the fingerprint stays on the YubiKey and it helps provide hardware isolation (device and YubiKey are separate hardware), the YubiKey can be registered to an anonymous identity, and during account setup on a secure anonymous communication system the YubiKey is registered in a way that it isn't tied to a real-world identity.

Scheme for system administrator: 1. The admin inserts their biometric Smartcard into the reader tied to a workstation. 2. The workstation reads the metadata on the card to initiate the authentication process. 3. The system prompts the admin to enter their PIN associated with their Smartcard. 4. The admin places their finger on the sensor tied to the workstation and that biometric data is compared to the stored template on the Smartcard. 5. If the biometrics match is successful and the PIN is correct, the Smartcard unlocks the cryptographic key pair and

permissions. 6. The workstation then accesses the encrypted credentials. 7. The result is successful authentication and the admin is able to access their secure environment. This scheme provides biometric privacy because the fingerprints are only matched on the card and not in the workstation, hardware-based trust which helps make the card tamper resistant and requires no external DB lookup, biometric check ensures that there is a physical human present, and FIDO2-based login for privileged access.

Scheme for third party administrator: 1. Auditor enters their username and password. 2. The system prompts for the YubiKey Bio to be inserted and generates a challenge (nonce). 3. The auditor inserts their YubiKey into the device 4. The system prompts the auditor to enter their PIN. 5. If the PIN is correct, the YubiKey prompts for a fingerprint and the auditor places their finger on the YubiKey. 6. This is followed by a challenge-response mechanism between the device and the YubiKey. 7. The system verifies the response from the YubiKey, using the public key. 8. If the validation is successful, the auditor is authenticated and granted access for a specific period of time. This scheme helps with biometric privacy for the auditor because the fingerprint is only stored in the YubiKey and it also is tamper resistant.

In a secure anonymous communication system, there cannot be objections to biometrics because multi-factor authentication used to access an account or a system is required to ensure privacy and authenticity.

In terms of handling false positives: For a YubiKey, one way a false positive occurs is if the physical key is stolen. A way to handle false positives is that the authentication attempt is logged, which can be audited for issues. Another way is that the user of the key receives a notification of a login attempt and is given the option to deny they tried to log in. For an auditor, the login and authentication is logged and the IP address of their device is verified. In terms of a Smartcard, an automatic revocation of the session to restrict access is necessary, along with a lockdown of the workstation for a short period of time and a notification for an investigation to happen sent to the owner of the Smartcard and their manager for verification. This event is logged for future forensic analysis. In addition, have the admin re-enroll their biometrics after a set period of time as well to ensure the template is accurate.

In terms of handling false negatives: for a YubiKey, provide user friendly feedback (i.e. "Authentication failed, please check that your YubiKey is able to connect to your device and try again"), log all authentication attempts, allow an end user or third party auditor to retry authentication as the key may not have been fully inserted and contact was not established, if other MFA factors are in place allow the use of alternative authentication factors to gain access (ex. PIN, OTP), after five unsuccessful login attempts lock the account temporarily and provide instructions in the secure communication system on how to troubleshoot the issue for the user.

For a third party auditor, they need to see if their YubiKey is still working and work with the company's security team to seek resolution of issues. For a Smartcard, logging detailed information of failed authentication attempts helps to identify what could be the cause of not being able to gain access, setting up alerts for the security team for five failed authentication attempts, allowing five login attempts with a timer used to space the attempts out (the sensor could be dirty or there could be moisture and fixing these could allow for successful authentication), and a temporary lock being placed on the account after five unsuccessful attempts.

Facilitator's Comments

The facilitator did not supply a comment.

Grade

100%

eCornell

Provide Feedback on this PDF

Submit Feedback