Authentication

CS 390 – Web Application Development

J. Setpal

November 9, 2022

Outline

Outline

WIWYT – Authentication, Cryptography, OAuth

- Build secure, breach-resilient systems.

WIWYT – Authentication, Cryptography, OAuth

- Build secure, breach-resilient systems.
- Store information securely.

WIWYT – Authentication, Cryptography, OAuth

- Build secure, breach-resilient systems.
- Store information securely.
- Build software that can communicate with other software, without risking the user base.

Outline

What is Authentication?

What is Authentication?

Authentication is a **verification routine** used to ensure that a user is who they say there are.

What is Authentication?

Authentication is a **verification routine** used to ensure that a user is who they say there are.

The need for authentication boils down to a lack of trust

What is Authentication?

Authentication is a **verification routine** used to ensure that a user is who they say there are.

The need for authentication boils down to a lack of trust (eg. checking your PUID during an exam).

What is Authentication?

Authentication is a **verification routine** used to ensure that a user is who they say there are.

The need for authentication boils down to a lack of trust (eg. checking your PUID during an exam).

Q: How do we establish this verification routine?

What is Authentication?

Authentication is a **verification routine** used to ensure that a user is who they say there are.

The need for authentication boils down to a lack of trust (eg. checking your PUID during an exam).

Q: How do we establish this verification routine?
A: **Shared Knowledge**.

What is Authentication?

Authentication is a **verification routine** used to ensure that a user is who they say there are.

The need for authentication boils down to a lack of trust (eg. checking your PUID during an exam).

Q: How do we establish this verification routine?
A: **Shared Knowledge**. "If you know/have/are `xyz`, and I know that only you know/have/are `xyz`, you're you!"

Authentication in the Web

What are examples of shared knowledge? Information the user:

a. **Knows:** Passwords, Tokens, Pattern-Matching

Authentication in the Web

What are examples of shared knowledge? Information the user:

a. **Knows:** Passwords, Tokens, Pattern-Matching

b. **Has:** Security Key

Authentication in the Web

What are examples of shared knowledge? Information the user:

a. **Knows:** Passwords, Tokens, Pattern-Matching
b. **Has:** Security Key
c. **Is:** Biometrics

Authentication in the Web

What are examples of shared knowledge? Information the user:

a. **Knows:** Passwords, Tokens, Pattern-Matching
b. **Has:** Security Key
c. **Is:** Biometrics

So, how do we go about:

a. Build Authentication Policy?

Authentication in the Web

What are examples of shared knowledge? Information the user:

a. **Knows:** Passwords, Tokens, Pattern-Matching
b. **Has:** Security Key
c. **Is:** Biometrics

So, how do we go about:

a. Build Authentication Policy?
b. Securely Storing Data?

Authentication in the Web

What are examples of shared knowledge? Information the user:

a. **Knows:** Passwords, Tokens, Pattern-Matching
b. **Has:** Security Key
c. **Is:** Biometrics

So, how do we go about:

a. Build Authentication Policy?
b. Securely Storing Data?
c. Implement Authentication?

Authentication in the Web

What are examples of shared knowledge? Information the user:

a. **Knows:** Passwords, Tokens, Pattern-Matching
b. **Has:** Security Key
c. **Is:** Biometrics

So, how do we go about:

a. Build Authentication Policy?
b. Securely Storing Data?
c. Implement Authentication?

Breadth-First Answer: Use popular pre-existing frameworks.

Authentication in the Web

What are examples of shared knowledge? Information the user:

a. **Knows:** Passwords, Tokens, Pattern-Matching
b. **Has:** Security Key
c. **Is:** Biometrics

So, how do we go about:

a. Build Authentication Policy?
b. Securely Storing Data?
c. Implement Authentication?

Breadth-First Answer: Use popular pre-existing frameworks.
Depth-First Answer: **Let's talk about it**.

Outline

Setting Password Policy

Q: What's the better password?

a. `a*s!c,s,[T wd(*UE#)dw$I!wl;kmw` (30 characters)

b. `Don't or fox find pinch swarm!` (30 characters)

Setting Password Policy

Q: What's the better password?

a. `a*s!c,s,[T wd(*UE#)dw$I!wl;kmw` (30 characters)

b. `Don't or fox find pinch swarm!` (30 characters)

A: Trick Question! They're equally complex.
Search Space: $7.72 \cdot 10^{57}$ possible combinations.

Setting Password Policy

Q: What's the better password?

a. `a*s!c,s,[T wd(*UE#)dw$I!wl;kmw` (30 characters)

b. `Don't or fox find pinch swarm!` (30 characters)

A: Trick Question! They're equally complex.
Search Space: $7.72 \cdot 10^{57}$ possible combinations.

Idea: Humans interpret information differently from computers.

Setting Password Policy

How can we capitalize on this?

a. Set minimum password lengths

Setting Password Policy

How can we capitalize on this?

a. Set minimum password lengths
b. Check against known breaches. `https://haveibeenpwned.com/`

Setting Password Policy

How can we capitalize on this?

a. Set minimum password lengths
b. Check against known breaches. `https://haveibeenpwned.com/`
c. Rate-limited Requests.

Setting Password Policy

How can we capitalize on this?

a. Set minimum password lengths
b. Check against known breaches. `https://haveibeenpwned.com/`
c. Rate-limited Requests.
d. **2 Factor Authentication**.

Setting Password Policy

How can we capitalize on this?

a. Set minimum password lengths
b. Check against known breaches. `https://haveibeenpwned.com/`
c. Rate-limited Requests.
d. **2 Factor Authentication**.
e. CAPTCHAs.

Setting Password Policy

How can we capitalize on this?

a. Set minimum password lengths
b. Check against known breaches. `https://haveibeenpwned.com/`
c. Rate-limited Requests.
d. **2 Factor Authentication**.
e. CAPTCHAs.

Q: Should we force regular password changes?

Setting Password Policy

How can we capitalize on this?

a. Set minimum password lengths
b. Check against known breaches. `https://haveibeenpwned.com/`
c. Rate-limited Requests.
d. **2 Factor Authentication**.
e. CAPTCHAs.

Q: Should we force regular password changes?
A: **NO**. Why not?

Setting Password Policy

How can we capitalize on this?

a. Set minimum password lengths
b. Check against known breaches. `https://haveibeenpwned.com/`
c. Rate-limited Requests.
d. **2 Factor Authentication**.
e. CAPTCHAs.

Q: Should we force regular password changes?
A: **NO**. Why not?

Forcing a change incentivizes building passwords using a pattern, or remembering them insecurely.

Outline

Saving Data Securely

We know how to work setup our data, but what about saving it?

Saving Data Securely

We know how to work setup our data, but what about saving it?

We need to make it such that even *when* there is a breach,
end users are minimally impacted.

Saving Data Securely

We know how to work setup our data, but what about saving it?

We need to make it such that even *when* there is a breach,
end users are minimally impacted.

Enter **Encryption**. It's the process of encoding information such that an unauthorized individual is unable to access a given set of information.

Encoding vs Encrypting

An important concept is the difference between encoding and encrypting.

Encoding vs Encrypting

An important concept is the difference between encoding and encrypting.

While encryption involves encoding data, the two are **not interchangeable** terms.

Encoding vs Encrypting

An important concept is the difference between encoding and encrypting.

While encryption involves encoding data, the two are **not interchangeable** terms.

Encoding data is used only when talking about data that is not securely encoded. Base64 is an encoding, SHA-256 is encryption.

Outline

What is OAuth?

OAuth is a secure mechanism for services to access data between one another securely.

What is OAuth?

OAuth is a secure mechanism for services to access data between one another securely.

Let's Implement OAuth!

If you can view this screen, I am making a mistake.

Outline

Thank you!

Have an awesome rest of your day!

Slides: `https://www.cs390.dev/slides/authentication.pdf`

If anything's incorrect or unclear, please ping: `jsetpal@purdue.edu`
I'll patch it ASAP.