

MODULE 2

Authentication

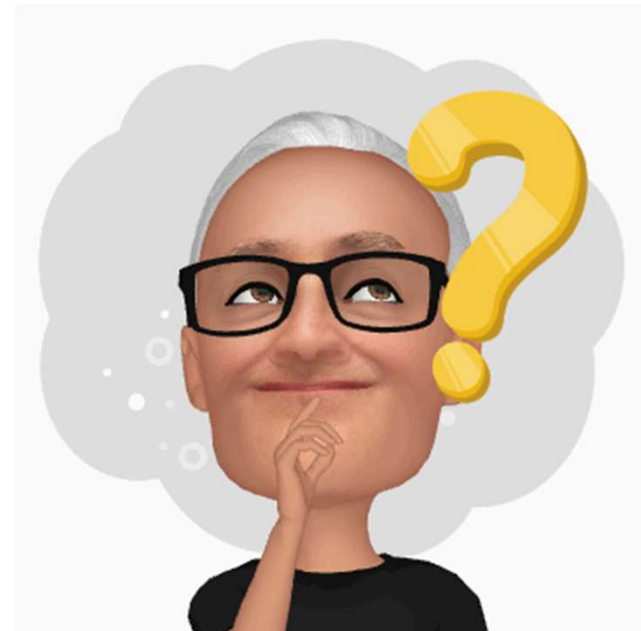


Authentication

- What is authentication?

Authentication is the process of verification that an individual, entity or website is who it claims to be.

Authentication is the front gate of any secure web application



Authentication Factors

- something the user ***knows***
- something the user ***has***
- something the user ***is***

Authentication Factors

- the **knowledge factors**: Something the user knows (e.g., a password, Partial Password, pass phrase, or personal identification number (PIN), challenge response (the user must answer a question, or pattern), Security question
- the **ownership factors**: Something the user has (e.g., wrist band, ID card, security token, cell phone with built-in hardware token, software token, or cell phone holding a software token)
- the **inherence factors**: Something the user is or does (e.g., fingerprint, retinal pattern, DNA sequence (there are assorted definitions of what is sufficient), signature, face, voice, unique bio-electric signals, or other biometric identifier).

Two-factor authentication

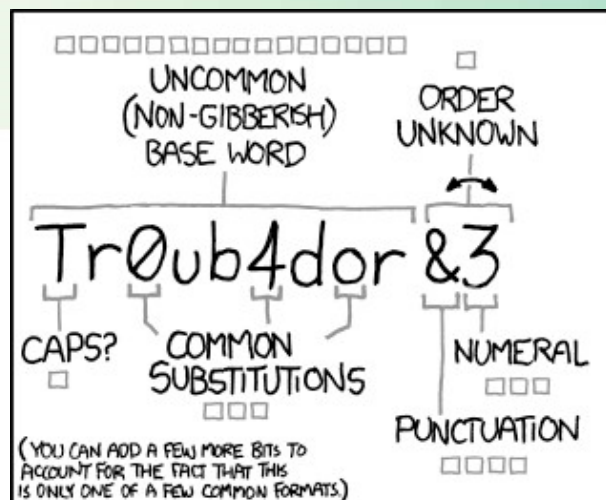
- a bankcard (something the user has) and a PIN (something the user knows).
- Business networks may require users to provide a password (knowledge factor) and a pseudorandom number from a security token (ownership factor).
- Apple requires developers to enter a password (something the user knows) and a pin sent to their registered phone (something the user has)

Passwords

- The measure of password strength is called **Entropy**.
- Strong passwords have the following characteristics
 - Length: Passwords shorter than 10 characters are considered to be weak
 - Complexity: allow virtually any character and be case sensitive in order to increase their complexity

Password Policy

- Password must meet at least 3 out of the following 4 complexity rules
 - at least 1 uppercase character (A-Z)
 - at least 1 lowercase character (a-z)
 - at least 1 digit (0-9)
 - at least 1 special character (punctuation) — do not forget to treat space as special characters too
- at least 10 characters
- at most 128 characters
- not more than 2 identical characters in a row (e.g., 111 not allowed)



~ 28 BITS OF ENTROPY

$2^{28} = 3 \text{ DAYS AT } 1000 \text{ GUESSES/SEC}$

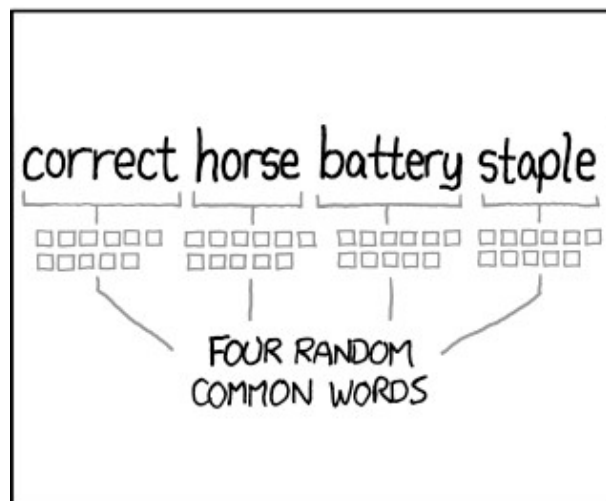
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)

DIFFICULTY TO GUESS: **EASY**

WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE 0s WAS A ZERO?

AND THERE WAS SOME SYMBOL...

DIFFICULTY TO REMEMBER: **HARD**



~ 44 BITS OF ENTROPY

$2^{44} = 550 \text{ YEARS AT } 1000 \text{ GUESSES/SEC}$

DIFFICULTY TO GUESS: **HARD**

THAT'S A BATTERY STAPLE.

CORRECT!

DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT

THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

Authentication Beyond Password Rules

- Don't tip your hand
 - Bad: The username is incorrect
 - Good: Login has failed
 - Bad: Invalid password
 - Good: Login has failed: Incorrect User Name or Password
- Make 'em wait
 - Block log in attempts for 20 minutes if 5 failed attempts
 - Each failed attempt takes longer and longer to return an error
- Log them out if idle
 - Session timeouts to prevent session hijacking.

Mechanics of Authentication

- Ask user for username and password
- Verify the user has provided the proper credentials
- Keep tabs that they are authenticated



JSON Web Tokens (JWT)

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MzkwMjQsInR5cCI6IkpXVCJ9.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MzkwMjQsInR5cCI6IkpXVCJ9

JSON Web Token (JWT) is an open standard ([RFC 7519](https://tools.ietf.org/html/rfc7519)) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object.

| |
|---|
| HEADER: ALGORITHM & TOKEN TYPE |
| <pre>{ "alg": "HS256", "typ": "JWT" }</pre> |
| PAYLOAD: DATA |
| <pre>{ "sub": "1234567890", "name": "John Doe", "iat": 1516239022 }</pre> |
| VERIFY SIGNATURE |
| <pre>HMACSHA256(base64UrlEncode(header) + "." + base64UrlEncode(payload), your-256-bit-secret) <input type="checkbox"/> secret base64 encoded</pre> |

JWT Highlights

- Their compact size allows for quick transfer with requests.
- They're often used as authorization mechanisms, storing user info such as their permissions or roles. These are called "claims."
 - They can contain any data that can be represented in JSON.
- JWT actually contains JSON, but it's encoded.
- Integrated with .NET Framework for claims base security

LET'S CODE!



ELEVATE  YOURSELF

WHAT QUESTIONS DO
YOU HAVE?



Reading for tonight:

