# COMP3052.SEC Computer Security

## Session 04: Authentication

# Acknowledgements


- Some of the materials we use this semester may come directly from previous teachers of this module, and other sources ...
- Thank you to (amongst others):
  - Michel Valstar, Milena Radenkovic, Mike Pound, Dave Towey, …

# This Session

- Authentication
- Problems with passwords
- Storing passwords
- Cracking passwords
- Multi-factor authentication
- Biometrics

# Authentication

- To allow someone access to an asset we must ensure:
  - They are permitted to access that asset
  - They are who they say they are
- We can attempt to verify identity using credentials
  - Something they are
  - Something they have
  - Something they know

# Usernames and Passwords

- Identification – Who you are
- Authentication – Verify that identity
- Authentication should expire
  - “Remember my credentials” turns this into something you have
- Time of check to time of use – TOCTTOU
  - Repeated authentication
  - At the start and during a session

# Passwords

- Passwords are digital keys
  - Simple to implement using existing libraries
  - Demonstrates someone is who they say they are
- Understood by the users (mostly)
- But: In many cases passwords are a terrible way to handle authentication

# Problems With Passwords

- People forget them
- They can be guessed
- Spoofing and Phishing
- Compromised password files
- Keylogging
- Many of these are made many times worse by weak passwords

# Weak Passwords

- If left to their own devices, people will use terrible passwords
  - Spouse's name
  - Known dates from their life
  - Small variants on their own name
  - qwerty1234

Check Nordpass, too:
https://nordpass.com/most-common-passwords-list/

| Rank | Password | Change from 2018 |
|---|---|---|
| 1 | 123456 | Unchanged |
| 2 | 123456789 | 1↗ |
| 3 | qwerty | 6↗ |
| 4 | password | 2↘ |
| 5 | 1234567 | 2↗ |
| 6 | 12345678 | 2↘ |
| 7 | 12345 | 2↘ |
| 8 | iloveyou | 2↗ |
| 9 | 111111 | 5↘ |
| 10 | 123123 | new |

Top passwords 2019, SplashData

# Password Policies

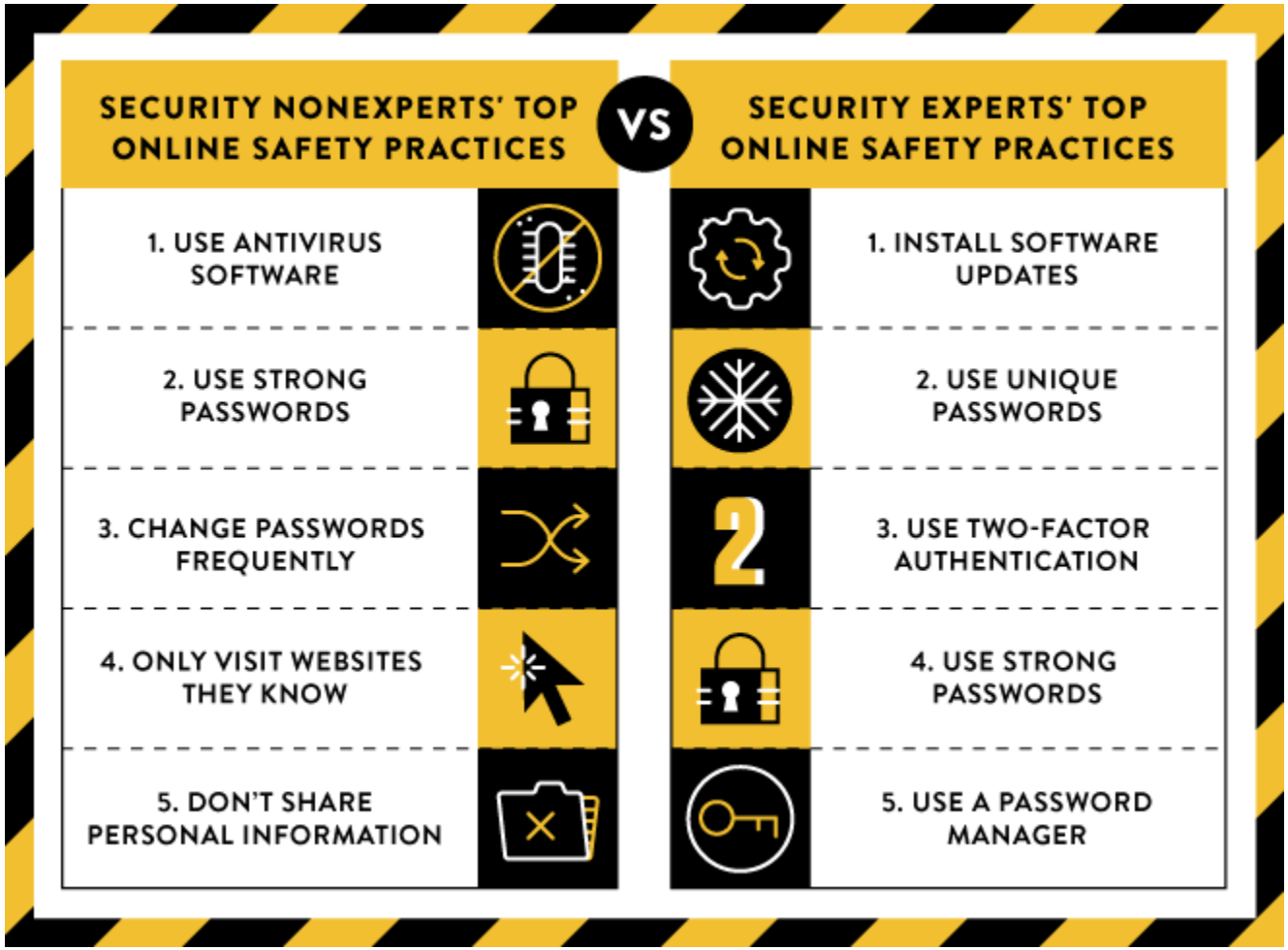
- Many companies (and now websites) enforce password policies
  - Certain length, certain types of characters
  - No dictionary words
  - Change regularly
  - No previously used passwords


https://www.pinterest.com/pin/635500197397659729/

# Password Policies

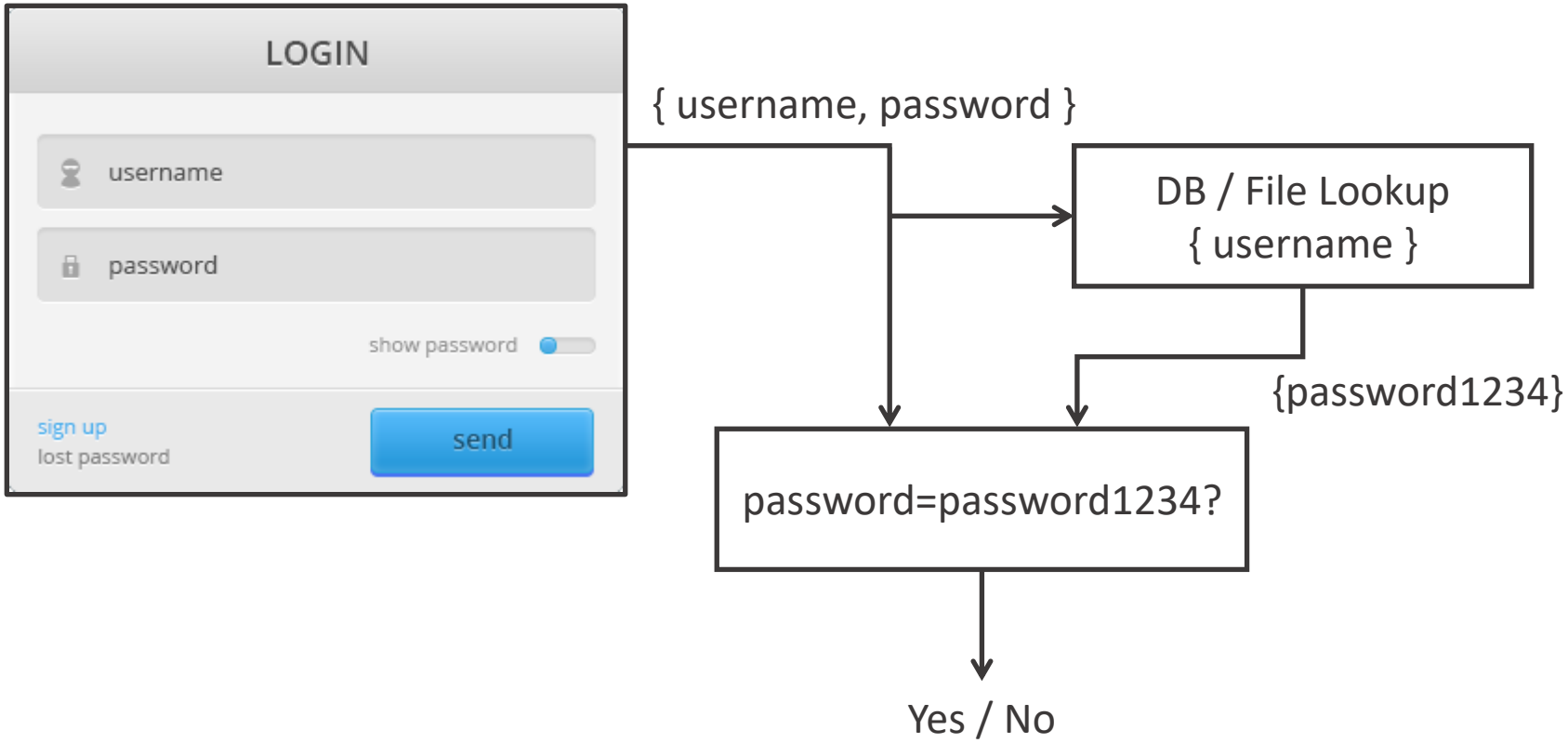
- This is not a great solution
  - People attempt to make their life easier by re-using passwords
  - When they're forced to change to unique passwords, they'll simply increment a counter

# Users Don’t Understand Security



# Password Authentication

- The bad way:

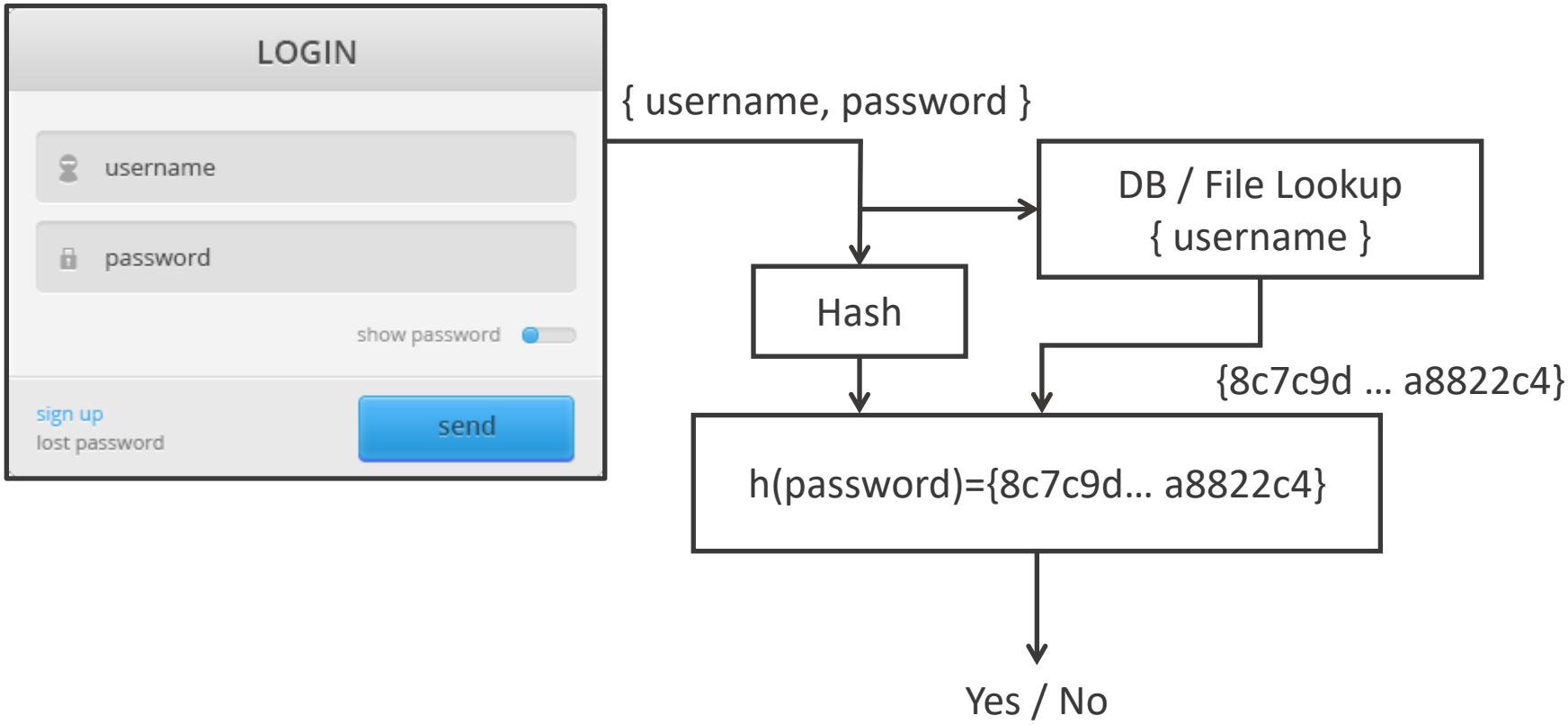


# Storing Passwords

- Storing passwords in plain text is a terrible idea
  - You might be hacked
  - Administrators can read them
- Storing encrypted passwords is better, but not perfect
  - Where are the keys stored?
  - Administrators can still read them

# Storing Passwords

- Using a one-way hash function is a much better solution:



# Password / Shadow Files

- Operating systems have taken steps to stop people reading hashes for offline attacks
  - Linux stores hashes in a shadow file /etc/shadow
  - Windows stores this in ..\system32\config\SAM
- These files are now read protected
- Administrators or people booting another OS will often find a way in

# Cracking Passwords

- Cracking a password isn't always illegal, though obviously it sometimes is!
- Password cracking falls into two basic types:
  - Offline: You have a copy of the password hash locally
  - Online: You do not have the hash, and are instead attempting to gain access to an actual login terminal
- Online is usually attempted with phishing

# Password Cracking

- Offline password cracking is simply a case of trying lots of possible passwords, and seeing if we have a hash collision with a password list
- This could be done with a Brute Force approach
  - Difficulty is calculated as $\{char\_count\}^{\text{length}}$
  - GPUs are fast, but not fast enough for long passwords

# Dictionary Attacks

- Most password cracking is now achieved using dictionary attacks rather than brute force
  - Using a dictionary of common words and passwords
  - Apply small variations to this list, trying them all
  - Combine words from two different lists
- *qwerty1234password1* is unbreakable using brute force, but won't last against a dictionary attack

# Password Strength

- Which of these passwords is strongest?

michael2001

password!

N!ghtingal3

helikesfootba_ll

# Password Strength

- What's the search space?

helikesfootba_ll

| | | |
|---|---|---|
| 3 words from 10000 | | $10000^3$ |
| 1 symbol from 15 | | 15 |
| 1 position from 16 | x | 16 |
| | | 240 Trillion |

**For a weak hash at 8Mh/s: 1 year**

# Password Strength

- A small improvement?

helikesnonlinearfootba_ll

| | | |
|---|---|---|
| 4 words from 20000 | | $20000^4$ |
| 1 symbol from 15 | | 15 |
| 1 position from 25 | x | 25 |
| | | 6 Quintillion |

**For a weak hash at 8Mh/s: 237,000 years**

# Password Salting

- We can improve security by prepending a random "salt" to a password before hashing
- The salt is stored unencrypted with the hash:

| Salt+Plaintext | | Salt/Hash |
|---|---|---|
| 4a6fbbpassword1234 | → | 4a6fbb/9f7cc17e654574de53f49a589d667b0a |
| tt5aspqwertylol | → | tt5asp/45dd00406a84755e6ae3e011c4215c27 |
| gy6sc3password1234 | → | gy6sc3/73f7f6dc3183623c0036269039dc5060 |

# Password Salting

- If we use a different random salt for each user, we get the following security benefits:
  1. Cracking multiple passwords is slower – a hit is for a single user, not all users with that password
  2. Prevents rainbow table attacks – we can't pre-compute that many password combinations

- Salting has no effect on the speed of cracking a single password – so make your passwords good!

# Hashing Speed

- When password cracking, the most important factor is hashing speed
- Newer algorithms take longer
  - Partly because they're more complex
  - But some have been specifically designed to take a while
- Iterate to increase complexity - PBKDF2
- bcrypt can't be easily used on GPUs

# System Design Issues

- Different situations require different password strengths and policies
  - A PIN code can be short, because you can only test three before you're locked out
  - Locking people out after *n* tries is fine in principle, but could lead to DoS
- Poorly designed interfaces can lead to a lot of problems
  - E.g. ATM machines have to be designed to prevent "shoulder surfing"

# System Design Issues

- Attacks on password storage are also extremely common
  - Obtaining shadow files
  - SQL Injection of web databases
- Can be made worse by stupid mistakes
  - On systems that log failed authentication attempts, what if you type your password in the wrong place?

# If Cracking Fails: Pretexting

- Obtaining private details by offering some "pretext" as a reason for needing them
- We continue to rely on email addresses, DOB and Mother's maiden names as our "last line of defense" for security
- How much information do we need to ring up a company as someone else?

# Alternatives

# Alternatives

- Passwords are something you know
  - Anyone who also knows this thing, becomes you

- What about something you have?
- Or something you are?

# Something You Have

- A key for a lock
- A keycard for a door
- A long password written down

- Again, anyone who obtains this item, becomes you
- Can be used in combination with something you know

# Multi-Factor Authentication

- Combines something you know with something you have
- Common examples:
  - Text codes to mobiles
  - One time passwords, Google Authenticator, Microsoft Authenticator etc.
  - USB devices e.g. Yubico/Yubikey
- New devices and TOCTTOU are common uses for two-factor authentication

# Biometrics

- Measurements of the human body, something you are
- Various forms, fingerprint recognition, iris / retina recognition, voice, gait, typing rhythm
- A password you always have with you, but you can’t change

# Biometric Accuracy

- Usually operate by finding “features” within data, then use these to learn a template for a given individual
- The accuracy of a biometric system is extremely important
  - False positive rate
  - False negative rate
- There will usually be a trade-off between FP and FN rates

# Fingerprint Recognition

- The main pattern is found
  - Arch, Loop, Whorl
- Minutiae – feature points
  - Ridge ending
  - Short ridge
  - Bifurcation
  - Crossover
  - Delta
  - Island
  - Enclosure

# Fingerprint Recognition

- Modern fingerprint recognition is about 99% accurate
  - Is this enough?
- The iPhone fingerprint recognition was bypassed within days, but it requires custom hardware
  - Probably robust to normal people stealing phones

# Where you are

- Location of access
  - Operator console vs any console
  - Office workstation vs home PC
  - Geographical location
- This is a useful addition to multi-factor authentication
- Not reliable on its own


# Summary

- Authentication
- Problems with passwords
- Storing passwords
- Cracking passwords
- Multi-factor authentication
- Biometrics

Anderson (2nd Ed)
Chapter 2
(especially 2.4)

Gollmann
Chapter 4