Authentication and Access Control

#week3

Without authentication how do we confirm a user is who they say they are? If someone gets access to something they're not meant to, what damage could they cause? Leads to risks of data breach or exfiltration, instillation of malware

The are three core operations in authentication

- Registration, establishes trusted foundation for identity verification
- Authentication checks, preserves security throughout
- Recovery, safeguards continuity without exposing vulnerabilities
  If one of these operations is compromised then the whole system fails

  - If **recovery** is too simple, it can **bypass or weaken strong authentication**.
  - If **registration** fails to verify identity rigorously, **secure recovery** becomes impossible.
  - If **authentication** is overly complex, users may adopt **unsafe shortcuts** that erode protection.

Authentication extends beyond passwords, it is about authentication in a system,

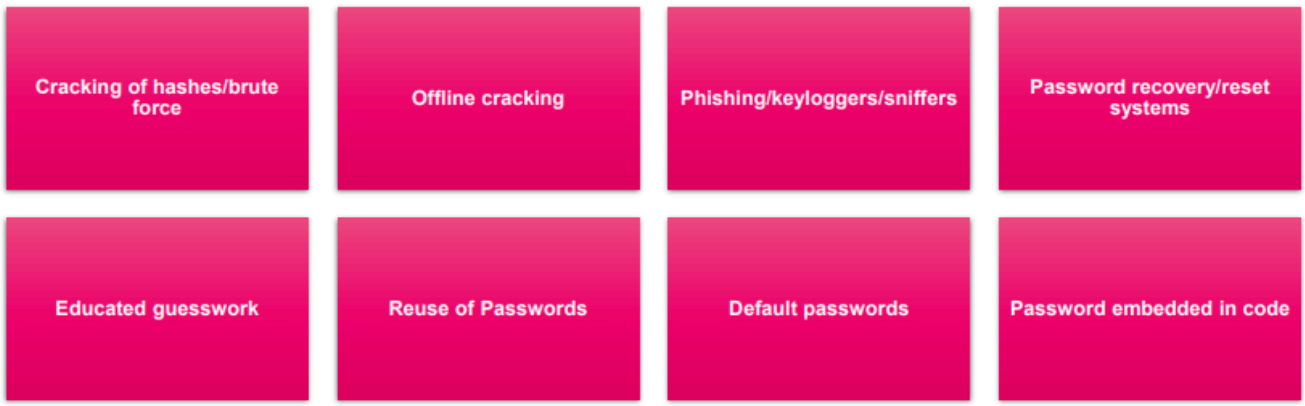
- This principle explains the industry's shift toward:
  - **Passwordless authentication**
  - **Biometric verification**
  - **Hardware-based security keys**

Each of these still have their own weaknesses, its important when designing a system to think of redundancy usability and layered verification

Passwords

We use passwords because they are conveniant, you don't need a third party to verify you

They still have 8 main weaknesses however



- **Cracking of hashes/brute force**

  If the intruder cannot obtain the password, then he can simply use a program to generate billions of possible passwords (often using the same techniques as are suggested for choosing passwords) and try each of them against the account. The crudest way to do this is to simply attempt to log in using each generated password: the resulting flood of password failures should be easy for a system administrator to spot, but since attackers continue to use this approach, it seems it is still reasonably successful. Attempts may be made against obscure authenticated services, such as SSH and LDAP, to reduce the chances of detection.

- **Offline cracking**

  Brute force attacks are much less obvious if the intruder can obtain a copy of an encrypted password, for example if a system's password file can be downloaded, if a hash has been included in a public file, or if an unknown machine can join an authentication group. Once the intruder has one or more encrypted passwords he can do the brute force guessing on his own machine (using modern hardware and algorithms this may take only a few minutes for short passwords), or even use a cloud service, and then return to login to the target once the correct password has been discovered.

- **Phishing/keyloggers/sniffers**

  The simplest way to discover someone's password is to have them tell you it. This can be done by persuading them to type it into a website you control (commonly known as **phishing**), by installing a **keylogger** (either hardware or software) on a computer, or by reading traffic on an unencrypted wireless or wired network. For intruders these methods have the great benefit that it does not matter how long or complex a password the user has chosen: the intruder can simply read it.

- **Password recovery/reset systems**

  An intruder may not need to get the password from the user if he can persuade the authentication system to either mail it to him or change it to something of his choice. Systems to allow the legitimate user to recover or change a password they have forgotten can also let other people do the same. Helpdesk operators need to be particularly careful to check the identity of anyone asking for a password reset. On-line systems that rely on "secret questions" such as "name of first school" or "birthday" are trivial to defeat if that information can be found on a social network. Systems that send reminders to a backup e-mail address or phone number can fail if the user changes address or number allowing the abandoned backup to be registered by someone else.

- **Social Engineering** - On your Facebook profile or LinkedIn profile, you can find: Name, Date of Birth, Location, Workplace, Interests, Hobbies, Skills, your Relationship Status, Telephone Number, Email Address and Favourite Food. This is everything a Cybercriminal needs to fool you into thinking that the message or email is legitimate.

- **Link Manipulation** - Most methods of phishing use some form of deception designed to make a link in an email appear to belong to the spoofed organization or person. Misspelled URLs or the use of subdomains are common tricks used by phishers. Many email clients or web browsers will show previews of where a link will take the user in the bottom left of the screen or while hovering the mouse cursor over a link.

- **Educated guesswork**

  It should be obvious that the same techniques used to guess the answers to secret questions can also be used to guess passwords. Anything based on something your friends will know, or that is available from a website, is a very poor choice as a password.

- **Reuse of Passwords**

  Most people now have many different accounts on different systems in both their private and work lives. Although best practice is to have a different password for every account, unfortunately it's much more common to reuse the same password on different services. That means that an organisation doesn't just have to worry about the above attacks against its own systems, it has to worry about the same attacks on all other systems where the same password has been used. This probably means that an organisation can no longer completely control whether its passwords are secure: it should also develop plans and systems to detect and respond when a password has been compromised.

- **Default passwords**

  Equipment and software often has standard pre-configured passwords which, of course, are well known to intruders. Such passwords should always be changed, though it can still be hard to find out where they may have been used. A related problem is where a password is set for the user by a local administrator. Unless the user is required to change the password to one that the administrator does not know, doubt can always be raised which of the two people who knew the password was actually logged in and responsible for the account's activity. If there are reasons that users cannot be forced to change their passwords on first use then procedures need to be carefully designed and followed to ensure that suspicion does not fall on the wrong person.

- **Password embedded in code**

  Passwords are also sometimes disclosed by being included in scripts or programs. While this may appear an easy way to automate access to an interactive system it carries high risks of disclosure and alternatives should be used wherever possible. If there is no other alternative, then the script or program must be very carefully protected against deliberate or accidental access. The worst possible outcome is for a script containing a plaintext password to end up on a public website.

Phishing

- **Spear phishing** - Phishing attempts directed at specific individuals or companies have been termed spear phishing. Attackers may gather personal information (social engineering) about their targets to increase their probability of success. This technique is, by far, the most successful on the internet today, accounting for 91% of attacks.

- **Clone phishing** - A type of phishing attack whereby a legitimate, and previously delivered email containing an attachment or link has had its content and recipient address(es) taken and used to create an almost identical or cloned email. The attachment or link within the email is replaced with a malicious version and then sent from an email address spoofed to appear to come from the original sender.

- **Voice Phishing** - Voice phishing is the criminal practice of using social engineering over the telephone system to gain access to personal and financial information from the public for the purpose of financial reward. Sometimes referred to as 'vishing', Voice phishing is typically used to steal credit card numbers or other information used in identity theft schemes from individuals.

Keyloggers

These are a system that record a users keystroke, there are both legal and illegal uses.

There can be both hardware keyloggers and software keyloggers

- Hardware
  - Physical devices inserted inline or embedded on cable/usb. Can be bought cheap online
  - Pros: Stealthy, hard to detect
  - Cons: Require physical access for instillation
- Software
  - Installed on a system
  - Can blend into legitimite proccess

Warning indicators consist of the following

- Sluggish system performance; mouse/keyboard lag
- Cursor disappearing or erratic behaviour
- Unexpected background processes (check Task Manager / Activity Monitor)
- Unknown installed programs or browser extensions
- Elevated network traffic from unknown process (sending logs)

Detection & Removal

- **Detection Steps**
  - Use system monitors (Task Manager, Activity Monitor) to spot anomalies Examine installed programs / software list
  - Run antivirus / anti-malware scans
  - Investigate suspicious browser extensions
- **Removal / Remediation**
  - Uninstall the malicious program
  - Clear temporary files / residual components
  - If deeply embedded, consider system restore or full reinstall
  - Change all relevant passwords after removal

Prevention & Mitigation Strategies

- Use a firewall and monitor outbound traffic
- Keep system & software patched (closing known vulnerabilities)
- Use strong, unique passwords + a password manager
- Be cautious with email links/attachments; verify sender legitimacy
- Avoid entering sensitive info on public or untrusted devices
- Use secure practices in software installation (trusted sources)
- Enable multi-factor authentication (MFA) wherever possible

Prevention

- **Two-factor Authentication**

  Most of the attacks described above can be made much harder if the password is not the only thing required to login. A variety of two-factor systems are available which require in addition either a biometric measurement (e.g. a fingerprint) or possession of a particular device (which may range from a dedicated token to a smartphone). Two-factor systems may be somewhat less convenient to use than simple passwords or be limited to particular hardware, so are most appropriate for accounts that have access to high-value services or information. For this level of security they may well be easier to use than a very long and complex static password.

- **Protecting password files**

  To check that the user has typed in the correct password, systems must have a reference to check against. An attacker who can obtain a copy of this reference file can run cracking programs against it and will almost inevitably succeed in discovering the passwords for several user accounts. Password files should therefore be among the best protected information the organisation holds, held on well-secured machines with limited access and, unless this is impossible, holding only salted hashes rather than the actual passwords. The choice of hashing algorithm can significantly affect the time to crack a password file - try to use the strongest (i.e. slowest) one available.

- **Federated authentication**

  Implementing federated or single sign on, using a central authentication server, has several security benefits. It reduces the number of systems on which passwords need to be stored, and should also ensure that secure protocols are used to transfer them over networks. Reducing the number of passwords users need to remember should help them use more complex and secure passphrases. However because the same password/phrase can now give access to multiple systems, it is even more important to secure the central authentication server, and for users to be careful against phishing or key logging attacks.

- **Password Complexity**

  Making passwords more complex increases the difficulty of attacks that rely on brute force or educated guessing. However it has no effect on attacks that reset the password or record it as the user types it in. The invention of rainbow tables as an alternative to brute-force attacks has made even complex passwords vulnerable in a few minutes if they are too short: most authorities now recommend the use of passphrases or sequences of random words to ensure sufficient length.

- **Password Lock-out**

  A common approach to reduce the risk of brute-force attempts to log in to an account is to either lock the account or increase the delay between login attempts when there have been repeated failures. This can be effective in slowing down attacks and giving responders time to react to an alarm. However it can cause problems when a user forgets to update a password stored in a browser or device if the automatic retries trigger the lock-out alarm.

• Self-test for Problems

A number of password cracking programs are available, so it makes sense for authorised staff to run them against the organisation's own password files. This must be carefully planned to minimise the security and legal risks to the organisation, its staff and information: testers should only need to know that a particular account was cracked, not what its password was. The exercise must be designed to help users select and remember better passwords, otherwise it risks reducing security rather than enhancing it.

• Detection/Containment

When a password has been compromised, the unauthorised user will normally behave differently from the authorised one. Logs of when accounts are used, and where from, may reveal early indications when this happens. It may also be possible to directly identify unauthorised use of the account.

- **Patterns of Use**

  Many accounts will show fairly obvious patterns both in when they are used (what times and what days of the week), and where users log in from. Indeed these may sometimes be a matter of policy: access to sensitive information may only permitted at designated locations and times. Changes to these patterns may indicate that there is a problem with the account. Unfortunately they may also be the result of legitimate events, such as the account owner being on holiday or having a deadline to meet. Even near simultaneous logins from different parts of the planet may only indicate that the user is having problems with their VPN set up. Login attempts from IP addresses that appear on address blocklists are almost always a bad sign. Some online services use signals such as these as a trigger for enhanced authorisation measures: if the site has suspicions that the user is not who they claim to be then they are asked to provide extra proof before being granted access.

- **Suspicious Activity**

  Most attacks on university accounts seem to be aimed at using the university's e-mail facilities either to phish more accounts (a deceptive e-mail from within the university is likely to be more convincing) , or else to send bulk e-mail. Monitoring for spam or phishing mails being sent from university accounts can often provide an early indication of problems; limiting the rate at which accounts can send mail may limit the damage caused, though dealing with the problem is still urgent.

  Some types of attacker publish passwords, or more commonly password files, that they have been able to obtain. The aim seems either to be to embarrass the organisation whose security has been breached, or to seek help in cracking hashed passwords. Monitoring the sites used for publication can therefore be an effective way to discover problems. Unfortunately most of the use of these sites is legitimate and harmless, but Janet CSIRT and other incident response teams have developed monitoring tools that increase the likelihood that an alert will actually indicate a problem.

- **Password Timeouts**

  A method sometimes proposed to limit the impact of password compromises is to require users to change them regularly. Time-limits for password age used to be set based on the time taken to discover them using brute-force methods, however since the invention of rainbow tables this would imply lifetimes of minutes or hours. Protecting hashed passwords against discover is now a better measure against this threat. Limited lifetimes may still help by disabling unused accounts in case account management procedures fail to do so, and by ensuring that changes to password policy or technology can be completed when all old passwords have expired, but these may well imply different expiry time-limits from those used previously. And, as the UK National CyberSecurity Centre's advice points out, any requirement to change passwords runs a significant risk of encouraging users to adopt sequences of passwords (e.g. by changing a digit) that increase the likelihood of a successful password guessing attack.

Secure Password Storage Methods

Passwords are stored hashed, a one way encryption. Passwords are then compared with this hash to ensure they are the same. Encryption is not used because it is two way and if encrypted passwords can be stolen they can be decrypted.

Hash passwords can still be brute-forced but hashing is designed in a way that makes it incredibly computationally expensive.

Enhancing Security

Hashing can be improved with salting and with peppering

A salt is a unique random value added to each password before hashing, as salts are different for each user an attacker must crack every hash independently. This also neatralises rainbow

tables as the same password will yield different hash per salt. Modern hashing algorithms embed salting internally

Peppering is a secret value that is added with the salt before hashing. These means even f a DB is compromised the attacker lacks full input to crack.

The aim of both of these is to make hashing increasingly computationally expensive

Handling Legacy / Poor Hashes

- Many systems have old MD5, SHA1, or unsalted hashes.
- Upgrade strategies:
  - *Re-hash on login*: when user authenticates, compute new hash and store it.
  - *Layered approach*: e.g. bcrypt(md5(password)) as transitional, but note potential issues.
  - *Force password reset* for inactive accounts or when migrating completely.

Implementation & Pitfalls Checklist

- Always generate salts with **secure randomness** (not predictable)
- Store salt (public) alongside hash
- Keep pepper secret & separate from DB
- Embed work factor/parameters in hash metadata (so system can adjust)
- Prevent performance degradation (don't make hash so slow it hurts system)
- Be careful with password length / input encoding / Unicode
- Plan for algorithm upgrades and migration

Two-Factor Authentication

A method of confirming a users identity using two distinct forms of evidence, essentially adding a second layer defence meaning if one factor is compromised the system remains safe.

The Three Factors of Authentication

- 2FA uses two of these from different categories.
- MFA (Multi-Factor Authentication) extends this to three or more.

Factor Type	Example	Description
Something you know	Password, PIN, Security question	Knowledge-based
Something you have	Smartphone, Smartcard, Hardware token	Possession-based
Something you are	Fingerprint, Face ID, Retina scan	Inherence-based