Lecture 3 – Authentication and Access Control

Networks and System Security

Importance of Authentication

Definition - Process companies use to confirm that only the right people, services, and apps with the right permissions can access organizational resources. Critical part of cybersecurity because attackers' number one priority is gaining unauthorized access.

Why Important:

- Protects systems, data, networks, websites, and applications from attacks
- Helps individuals keep personal data confidential
- Enables safe online business (banking, investing)
- Weak authentication makes it easier for attackers to compromise accounts through password guessing or credential theft

Risks of weak authentication:

- Data breach or exfiltration
- Installation of malware (ransomware)
- Noncompliance with regional/industry data privacy regulations

Core Operations in Authentication

Three stages:

1. **Registration** - Establishes trusted foundation for identity verification
2. **Authentication checks** - Preserves security throughout routine access
3. **Recovery** - Safeguards continuity without exposing vulnerabilities

The Security Ecosystem

Systemic thinking required - Each stage forms part of interconnected framework. Components must reinforce one another rather than compete.

Goldsmiths
UNIVERSITY OF LONDON

Lecture 3 – Authentication and Access Control

Interdependencies and Risks:

- 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

Key principle: Security breaks not where it is weakest individually, but where design coherence fails.

Resilient design anticipates failure - Even if one component is compromised, others uphold integrity.

Industry shift toward:

- Passwordless authentication
- Biometric verification
- Hardware-based security keys

These approaches build systems that remain secure through redundancy, usability, and layered verification.

Why Passwords? The Core Paradox

Convenience vs Security Trade-off - The most secure approaches are often the least convenient, while the most convenient approaches tend to be the least secure. This is an inherent mathematical and psychological reality, not a design flaw.

Threats to Passwords (Application Layer: 2-10)

1. Cracking of Hashes/Brute Force

- Intruder uses program to generate billions of possible passwords and tries each against the account
- Crudest method: attempt to log in using each generated password (flood of failures should be easy for sysadmin to spot)

- Attackers may target obscure authenticated services (SSH, LDAP) to reduce detection chances

2. Offline Cracking

- Much less obvious than online brute force
- Intruder obtains copy of encrypted password (from downloaded password file, public hash, or authentication group)
- Can perform brute force guessing on own machine using modern hardware and algorithms
- May take only a few minutes for short passwords
- Can use cloud services for cracking
- Returns to login once correct password discovered

Password strength checker: https://www.passwordmonster.com/

3. Phishing/Keyloggers/Sniffers

Simplest way to discover password is to have users tell you it.

Methods:

- **Phishing** - Persuade users to type password into website you control
- **Keylogger** (hardware or software) - Install on computer
- **Sniffing** - Read traffic on unencrypted wireless or wired network

Advantage for intruders: Doesn't matter how long or complex the password is - they can simply read it.

4. Password Recovery/Reset Systems

Intruder doesn't need password from user if they can persuade authentication system to mail it or change it.

Vulnerabilities:

- Helpdesk operators must carefully check identity of anyone asking for password reset

- Online systems with "secret questions" (first school name, birthday) trivial to defeat if information on social networks
- Systems sending reminders to backup email/phone can fail if user changes contact info, allowing abandoned backup to be registered by someone else

Phishing Details

Social Engineering: Information available on Facebook/LinkedIn profiles:

- Name, Date of Birth, Location, Workplace
- Interests, Hobbies, Skills
- Relationship Status, Telephone Number, Email Address, Favourite Food
- Everything a cybercriminal needs to fool you into thinking message/email is legitimate

Link Manipulation:

- Most phishing uses deception to make link appear to belong to spoofed organization
- Misspelled URLs or subdomains are common tricks
- Email clients/browsers show link previews in bottom left or on hover

Spear Phishing:

- Phishing directed at specific individuals or companies
- Attackers gather personal information (social engineering) to increase success probability
- Most successful technique on internet today: accounts for 91% of attacks

Clone Phishing:

- Legitimate, previously delivered email with attachment/link is copied
- Content and recipient addresses used to create almost identical cloned email
- Attachment/link replaced with malicious version

- Sent from spoofed email address appearing to come from original sender

Voice Phishing (Vishing):

- Criminal practice using social engineering over telephone
- Gain access to personal and financial information
- Typically used to steal credit card numbers or information for identity theft schemes

Keyloggers (Keystroke Loggers)

Definition: Record user keystrokes on a device.

Legal vs Malicious Uses:

- *Legal:* Parental monitoring, employee monitoring (if permitted), IT diagnostics (e.g., CrowdStrike)
- *Malicious:* Spying, credential theft, identity theft

Example: DarkHotel campaign used unsecured hotel Wi-Fi to push keylogger software, then erased itself to minimise detection.

Types of Keyloggers:

Hardware Keyloggers:

- Physical devices inserted inline (between keyboard and computer) or embedded in cable/USB adapter
- Pros: Stealthy (if well disguised), hard to detect by software
- Cons: Require physical access for installation/retrieval

Software Keyloggers:

- Installed on system (user-mode, kernel-mode)
- Subtypes/techniques:
  - Form-grabbing (intercept form submissions)
  - JavaScript keyloggers (injected into web pages)

    - API/system hook interception

- Can blend into legitimate processes, hide via rootkit functionality

Warning Signs/Indicators:

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

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 malicious program
- Clear temporary files/residual components
- If deeply embedded, consider system restore or full reinstall
- Change all relevant passwords after removal

Prevention & Mitigation:

- Use firewall and monitor outbound traffic
- Keep system & software patched (closing known vulnerabilities)
- Use strong, unique passwords + 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

DarkHotel Case Study Discussion Points:

- What factors made this effective?
- At what points could better defense have prevented it?
- How might organization detect early signs of compromise via keylogging?

5. Educated Guesswork

- Same techniques used to guess secret question answers can guess passwords
- Anything based on what friends know or available from website is very poor password choice

6. Reuse of Passwords

Reality: Most people have many different accounts on different systems (private and work). Best practice is different password for every account, but much more common to reuse same password on different services.

Problem: Organization must worry not just about attacks on its own systems, but attacks on all other systems where same password used. Organization can no longer completely control whether passwords are secure - must develop plans and systems to detect and respond when password compromised.

NordPass Research (2025): Study of 1,727 adults (619 Americans, 605 Britons, 503 Germans) examining:

- How often people reuse logins
- How many passwords and accounts affected
- Why they still do it

Results by Country:

United States:

- 62% "often" or "always" reuse a password

- Median reuser juggles 3 core passwords unlocking ~5 different accounts
- 50% do it because "easier to remember fewer passwords"
- 33% feel overwhelmed by sheer number of services used monthly
- Troubling 11% see "no significant risk" in repetition

United Kingdom:

- 60% recycle logins
- Memory anxiety eclipses convenience: 40% fear locking themselves out if every password unique
- Convenience and "too many accounts" tie for second place
- Same 11% shrug off threat altogether

Germany:

- 50% reuse passwords (best score but still coin toss)
- Convenience is main motive for 37%
- 29% cite account overload
- 13% believe repetition practically harmless

Overall conclusion: ~57% of consumers across 3 advanced economies still bet on duplicate credentials. Majority large enough to keep credential-stuffing operations profitable for years.

7. Default Passwords

- Equipment and software often has standard pre-configured passwords well known to intruders
- Should always be changed, but hard to find where they may have been used
- Related problem: password set for user by local administrator - unless user required to change password to one administrator doesn't know, doubt can be raised about who was actually logged in

- If users cannot be forced to change passwords on first use, procedures must be carefully designed to ensure suspicion doesn't fall on wrong person

8. Password Embedded in Code

- Passwords sometimes disclosed by being included in scripts or programs
- May appear easy way to automate access to interactive system but carries high disclosure risks
- Alternatives should be used wherever possible
- If no alternative, script/program must be very carefully protected against deliberate or accidental access
- Worst outcome: script containing plaintext password ends up on public website

Prevention Measures

1. Two-Factor Authentication

- Makes most attacks much harder if password not the only thing required to login
- Variety of systems available requiring either:
  - Biometric measurement (e.g., fingerprint)
  - Possession of particular device (dedicated token to smartphone)
- May be somewhat less convenient than simple passwords or limited to particular hardware
- Most appropriate for accounts with access to high-value services or information
- May be easier to use than very long and complex static password for this security level

2. Protecting Password Files

- Systems must have reference to check against user-typed password
- Attacker obtaining copy of reference file can run cracking programs and will almost inevitably discover passwords for several accounts

- Password files should be among best protected information organization holds
- Held on well-secured machines with limited access
- Unless impossible, should hold only salted hashes rather than actual passwords
- Choice of hashing algorithm significantly affects time to crack password file - use strongest (slowest) one available

3. Federated Authentication

Benefits:

- Reduces number of systems where passwords need to be stored
- Ensures secure protocols used to transfer passwords over networks
- Reducing passwords users need to remember helps them use more complex and secure passphrases

However: Because same password/phrase can now give access to multiple systems:

- Even more important to secure central authentication server
- Users must be careful against phishing or key logging attacks

4. Password Complexity

- Making passwords more complex increases difficulty of brute force or educated guessing attacks
- Has no effect on attacks that reset password or record it as user types
- Rainbow tables as alternative to brute-force made even complex passwords vulnerable in few minutes if too short
- Most authorities now recommend passphrases or sequences of random words to ensure sufficient length

5. Password Lock-out

- Common approach to reduce brute-force login attempts risk
- Either lock account or increase delay between login attempts after repeated failures

- Effective in slowing attacks and giving responders time to react to alarm
- Can cause problems when user forgets to update password stored in browser/device if automatic retries trigger lock-out alarm

6. Self-test for Problems

- Number of password cracking programs available
- Makes sense for authorized staff to run them against organization's own password files
- Must be carefully planned to minimize security and legal risks
- Testers should only need to know particular account was cracked, not what password was
- Exercise must help users select and remember better passwords, otherwise risks reducing security

7. Detection/Containment

When password compromised, unauthorized user normally behaves differently from authorized one.

Patterns of Use:

- Many accounts show obvious patterns in when used (times, days of week) and where users log in from
- May be matter of policy: access to sensitive information only permitted at designated locations/times
- Changes to patterns may indicate problem with account
- Unfortunately may also result from legitimate events (vacation, deadline)
- Near simultaneous logins from different parts of planet may only indicate VPN setup problems
- Login attempts from IP addresses on blocklists almost always bad sign

- Some online services use signals as trigger for enhanced authorization measures - ask for extra proof if suspicions

Suspicious Activity:

- Most university attacks aimed at using email facilities to phish more accounts or send bulk email
- Monitoring for spam/phishing emails from university accounts provides early indication
- Limiting rate at which accounts can send mail may limit damage
- Some attackers publish passwords or password files they've obtained (to embarrass organization or seek cracking help)
- Monitoring publication sites can be effective way to discover problems
- Janet CSIRT and other incident response teams developed monitoring tools to increase likelihood alerts indicate actual problems

8. Password Timeouts

- Sometimes proposed to limit impact of password compromises by requiring regular changes
- Time-limits used to be based on time to discover using brute-force, but rainbow tables now imply lifetimes of minutes or hours
- Protecting hashed passwords against discovery now better measure against this threat
- Limited lifetimes may still help by:
    - Disabling unused accounts if account management fails
    - Ensuring changes to password policy/technology can be completed when all old passwords expired
- UK National CyberSecurity Centre advice: any requirement to change passwords runs significant risk of encouraging users to adopt sequences (e.g., changing digit) that increase likelihood of successful password guessing attack

Lecture 3 – Authentication and Access Control

Secure Password Storage Methods

Hashing vs Encryption

Hashing:

- One-way function
- You can't "decrypt" a hash

Encryption:

- Two-way
- If key is exposed, plaintext is recoverable

Principle: Passwords should be hashed, not encrypted (except in special cases).

When Hashes Are Cracked

- Attackers try candidate passwords → compute their hash → compare to stored hash
- Use large wordlists, brute force, GPU acceleration make many hashes crackable if defenders' choices are weak
- Aim is defense in depth: pick algorithms & parameters so cracking is expensive

Salts (Enhancing Security)

Definition: A salt is unique, randomly generated value added to each password before hashing.

Benefits (because salts different for each user):

- Attacker must crack every hash independently (no reuse)
- Rainbow tables are neutralized (same password yields different hash per salt)

Modern password hashing algorithms (Argon2, bcrypt, PBKDF2) embed salting internally.

Peppering (Additional Layer)

Goldsmiths

UNIVERSITY OF LONDON

Lecture 3 – Authentication and Access Control

Definition: A pepper is secret value (shared across passwords) not stored in DB (e.g., stored in HSM or separate vault).

Purpose: Even if DB compromised, attacker lacks full input to crack.

Types:

- Pre-hashing pepper: add pepper before hashing
- Post-hashing pepper (HMAC on hash): treat pepper as HMAC key on top of hash

Caveats: Rotating pepper requires user password resets.

Work Factors / Iterations

Work factor = how expensive the hashing computation is (iterations, memory, CPU).

Tradeoff: Increased work factor → slower legitimate login verification but much harder for attacker.

Must:

- Choose parameters appropriate to hardware
- Revisit them periodically
- When upgrading: re-hash next time user logs in (can keep multiple algorithms during transition)

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 (note potential issues)
- Force password reset for inactive accounts or when migrating completely

Implementation & Pitfalls Checklist

- Always generate salts with secure randomness (not predictable)

Lecture 3 – Authentication and Access Control

- 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 (2FA)

Definition

Method of confirming user's identity using two distinct forms of evidence from different categories.

Goal: Add second layer of defense, so even if one factor (e.g., password) is stolen, unauthorized access is harder.

The Three Factors of Authentication

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

2FA: Uses two factors from different categories **MFA (Multi-Factor Authentication):** Extends to three or more factors

Common 2FA Methods

1. SMS-based codes

- One-time code sent to registered phone number
- Vulnerable to SIM swapping, interception

2. App-based (TOTP)

- Time-based One-Time Password generated by apps (Google Authenticator, Authy, Microsoft Authenticator)
- Offline, more secure than SMS

3. Push notifications

- "Approve login" prompts (Duo, Okta, Microsoft)
- Convenient but can lead to push fatigue (users approve without checking)

4. Hardware tokens

- Physical devices like YubiKey or RSA SecurID
- Provide cryptographically strong authentication

5. Biometric 2FA

- Fingerprint or facial recognition paired with password/PIN
- Privacy and device limitations may apply

Behind the Scenes: How TOTP Works

Process: Shared Secret + Current Time → HMAC-SHA1 → Truncate → 6-digit Code

Details:

- Based on shared secret key between client and server
- Uses current time and key to generate one-time 6-digit code
- Valid for short intervals (typically 30 seconds)
- Implemented using standards: RFC 6238 (TOTP) and RFC 4226 (HOTP)

Example Scenario (User Login Flow)

1. User enters username & password
2. Server verifies password
3. System prompts for second factor (e.g., TOTP or push)

4. Only when both factors verified → access granted

If attacker knows only password: login fails due to missing second factor.

Advantages of 2FA

- Dramatically reduces unauthorized access from stolen passwords
- Simple, low-cost protection for most systems
- Builds user confidence in service security
- Compatible with most major platforms and cloud services

Limitations & Risks

- Phishing kits can capture both password and one-time code
- Man-in-the-middle attacks may intercept login flow
- SIM swapping makes SMS 2FA unreliable
- User fatigue: repeated prompts → careless approvals
- Device loss may lock out legitimate users

Best practice: Prefer app-based or hardware-based 2FA; avoid SMS when possible.

Real-World Examples

- **Google (2018):** Mandated 2FA for employees; reported zero account takeovers afterwards
- **GitHub (2023):** Made 2FA mandatory for all contributors
- **Banks & healthcare systems:** Increasingly use token- or app-based MFA due to regulatory requirements

Best Practices for Implementation

- Educate users on importance of 2FA and setup
- Offer backup/recovery methods (backup codes, trusted devices)
- Use open standards (TOTP, FIDO2/WebAuthn)

- Store shared secrets securely (encrypted vault, hardware module)
- Allow adaptive authentication: add more checks for high-risk logins

Beyond 2FA: Toward Passwordless Authentication

- **FIDO2 / WebAuthn:** Use public-key cryptography and hardware tokens
- **Passkeys:** Modern, phishing-resistant alternative to passwords
- Combine biometrics and hardware authentication for stronger, simpler security

SMS-Based 2FA (Deep Dive)

Overview

Concept: Adds extra layer of security by requiring one-time code sent via text message, in addition to password.

Typical Process:

1. User enters username and password
2. System sends 6-digit OTP (one-time passcode) via SMS
3. User enters OTP to complete authentication

Goal: Mitigate risks of password compromise (phishing, credential stuffing).

The Security Model Assumption

- SMS assumed to provide "separate channel" from internet
- Relies on telecommunications network as trusted intermediary
- **Problem:** This trust model is outdated - mobile network and SIM system never designed for high-security authentication

Vulnerabilities and Attack Vectors

1. SIM Swapping

Mechanism: Attacker convinces mobile carrier to transfer victim's phone number to SIM card they control.

Result: All SMS messages (including OTPs) sent to attacker.

Attack surface: Social engineering of telecom staff, weak carrier identity verification.

Impact: Full account takeover even if password is strong.

Example: In 2019, Twitter CEO Jack Dorsey's account compromised via SIM swapping - attackers gained control over his phone number and intercepted authentication codes.

2. Man-in-the-Middle (MitM) Phishing

Mechanism: Attacker tricks user into logging in on fake website. Site proxies credentials and SMS code to legitimate site in real time.

Tools: Evilginx or Modlishka can automate this.

Impact: Both password and OTP stolen during session.

Observation: SMS codes authenticate possession of a number, not identity of the session origin.

3. SS7 Network Exploits

- Signaling System 7 (SS7) underpins SMS delivery
- Vulnerabilities allow attackers to intercept SMS traffic at telecom level
- Exploitation demonstrated repeatedly since 2014

Usability vs. Security Trade-off

Aspect	**Advantage**	**Weakness**
Accessibility	Works on any phone, no app needed	Relies on insecure SMS protocol
User adoption	Simple and familiar	High risk from phishing and SIM fraud
Cost	Cheap to deploy	Expensive to mitigate breaches

Critical Assessment

Lecture 3 – Authentication and Access Control

- SMS-based 2FA does improve over password-only systems, particularly against automated attacks
- However, provides false sense of strong protection:
    - Vulnerable to human and infrastructural manipulation
    - Inadequate against targeted attacks

Recommendation: Use SMS 2FA only as temporary or fallback mechanism; promote migration to app-based or hardware-based authentication.