# Access Control

## CISC 3325 — CUNY Brooklyn College
## Lecture Notes

# Access Control



In this chapter, we will:

1. Introduce physical security threats and provide adequate solutions.
2. Define **identification** vs. **authentication**.
3. Learn authentication mechanisms and examples. How can we authenticate a user?
4. Analyze the important role of Biometrics (and its dangers) in authentication.
5. Define **authorization** and discuss **access control** models.

# Physical Security



Before we reach the discussion on Access Control at the end of these lecture notes, we first need to cover the prerequisite topic of physical security, which is frequently touched by the topic of Access Control.

**Physical security** refers to the measures taken to protect the physical environment and infrastructure that stores the information system resources, including hardware, software, and other networking devices against physical threats such as theft, fire, water, floods, and so on.

Although this necessity might sound quite mundane and obvious, one should focus on maintaining physical security with the same seriousness as protecting their software, network, etc.

Even if you opt to store all your data and information on the cloud, thereby freeing yourself from maintaining the physical security of your data, remember that your data is now stored on a company's physical database servers.
How well does this company maintain the physical security of your data from theft, disasters, and spying?

# Physical Security



This table lists several categories of physical security threats and the negatively-affected CIA traits:

| Threat Category | Affected CIA Traits |
|---|---|
| Physical damage | Availability |
| Theft | Confidentiality, Availability |
| Unauthorized entry to the facility | Confidentiality, Integrity |
| Natural Disasters (Fire, Flood, Earthquakes) | Availability |
| Human Intervention (Sabotage, Vandalism, Strikes) | Availability, Confidentiality |
| Emergencies (Fire, Smoke, Building collapse, Explosion, Water leaks, Toxic materials) | Availability |

***Follow-up question***: Why does unauthorized entry affect Confidentiality, but natural disasters don't?

# Physical Security



Physical damage is attributed to one or more of the **seven sources of physical loss** below:

1. **Temperature**: Extreme variations of temperature that harm electronic devices.
2. **Gases**: War gases, commercial vapors, humidity, dry air, and so on. Examples include transformer explosion gas, air-conditioning failures, smoke or smog, printer liquids and toners, and cleaning liquids.
3. **Liquids**: Water and chemicals. Examples would be water pipe leakages, sanitary leakages, fuel leaks, spilled drinks, acids, and chemicals used for cleaning.
4. **Organisms**: Viruses, bacteria, people, animals, and insects (bugs!).
5. **Projectiles**: Objects in motion, e.g., moving vehicles, cars, trucks, and explosions.
6. **Movements**: Collapse, shearing, shaking, vibrations, and so on.
7. **Energy Anomalies**: Electric surges, failures, magnetism, static electricity, radiation, sound, light, radiowaves, and microwaves. Examples include static electricity or carpets, cosmic radiation, explosion, and decomposition of magnetic tapes.

# Physical Security



Normally, organizations have standard policies and procedures to protect their facilities, including their data/computer center:

- **General management policies and procedures**, such as security guards, allowing visitors inside the facility after proper checks, escorting visitors, building access, and surveillance cameras at each and every important location, both outside and inside of the facilities.
- **IT security policies and procedures**, such as guarding against unauthorized access to restricted areas, control and privileges of system administrators, password policies, remote system access policies, access card privileges, and environmental controls required for the data centers such as temperature, humidity, static, and dust controls.

The next slides describe examples of physical threats based on some of the categories from the previous slide and mention which control(s) to use to address each threat.

# Physical Security Threats and Solutions



1. **Threat**: Surges in electrical power, often caused by lightning, may burn out electronic components in the computer.
   **Control**: Use an uninterruptible power supply (UPS), which regulates the incoming voltage and produces a clean output signal. If the voltage gets high, the UPS trims it. If the voltage drops, the UPS uses its internal battery to supply the computer with power for a few minutes, enough to either turn off the computer or start an alternative generator.
2. **Threat**: Fire, in cases where water as an extinguisher might damage electronic devices.
   **Control**: Use special liquid chemicals, e.g., NOVEC, which extinguishes fire and evaporates quickly enough before damaging devices.
3. **Threat**: Theft of electronic devices and hardware.
   **Control 1**: To mitigate theft of mobile devices, use special applications that can remotely lock your device and/or wipe out its content (to prevent the thief from accessing your data.)
   **Control 2**: To mitigate theft of a laptop, make sure to encrypt all sensitive data.
   **Control 3**: Also for laptops: if used in a public place, tie it to a heavy object (cabinet, desk, etc.) with a lock.

# Physical Security Threats and Solutions



4. **Threat**: **Piggybacking** or **Tailgating** (= when an intruder illegally enters a building/office right after an employee, while the door is still open.)
   **Control**: If there is no security guard, install a **mantrap**, which is an entrance consisting of two doors: Door B won't open until the outer Door A fully closes. Next is a video showing how a mantrap works!
5. **Threat**: Magnetic fields that can interfere with hard disks and magnetic tapes.
   **Control**: Disable entrance into a computer room with devices using big, powerful magnets, and use tiled floors or anti-static carpets to prevent static electricity.
6. **Threat**: When many employees work on the same device, tracking who did which change is a problem.
   **Control 1**: Assign each user's account a password so that, each time a user accesses their account, they are asked to enter their password.
   **Control 2**: In case of fast working environments, e.g., in an emergency room (where medical personnel must access a computer near a patient as fast as possible,) give each employee an electronic ID card that they can swipe at or tap on the computer to instantly sign into their account.

# Physical Security Threats and Solutions



Demonstrating of mantrap doors:

https://www.youtube.com/watch?v=1jGklBwZ7ZQ

# Physical Security Threats and Solutions



7. **Threat**: Loss of data due to wear or deterioration of the storage devices (e.g., disks, papers, etc.)
   **Control**: Data backups! Either periodic backups (every X minutes, days, etc.) or whenever an important file is changed/updated.
8. **Threat**: Papers with confidential info are viewed after being dumped.
   **Control**: Shred documents with sensitive information or any personal information, e.g., names, addresses, IDs, bank account numbers (even partially-displayed ones.)
9. **Threat**: Physical espionage, e.g., peeping over the shoulder or through the window at someone's screen.
   **Control 1**: Apply a privacy filter that tints the screen so that glancing at an angle makes a screen's contents invisible.
   **Control 2**: Install thick curtains in rooms with computers that store confidential data.
10. **Threat**: Data whose bits are corrupted either intentionally or unintentionally (due to errors during transition.)
    **Control**: error-detecting and error-correcting (error-control) codes to detect and properly correct the corrupted bits.

Any questions?

# Physical and Technical Controls



**Physical and technical controls** (or **physical access controls** shortly) restrict the personnel access to the office buildings, labs, server rooms, data centers, or computer operation rooms where critical assets related to IT operations are operational.

Besides such restrictions, enhanced corrective actions can reduce these risks.

- Higher levels of screening, reorganizing movement patterns at some key locations, and not displaying names of important buildings, can reduce the physical threats.
- Also, closed-circuit television cameras, motion detectors, and other devices can monitor the activities of people and detect any intrusions.

There are a number of physical security controls available in the market that the organization should consider for implementing physical access controls both inside and outside of the facility; we'll introduce each of these access controls in the next slides.

# Physical and Technical Controls



1. **Security Guards at each of the entry and exit points**: security personnel filter who is allowed to enter a facility or a specific room therein. Criteria that security personnel use are whether the person is an employee, whether they have clearance to enter the particular room, whether they carry disallowed weapons or other substances, and how suspicious their behavior is. If a guard notices that they can't handle a dangerous situation by themselves, they will call for backup or notify first responders.
2. **Photo ID cards**, which contain the name and a photo of the cardholder (aside from other information, such as the person's role or department) are used to grant access to a facility to employees and easily identify a violator of rules or policies within the facility.
3. **Magnetic Access Cards** contain magnetic encoded strips with an employee's information. Once a card is swiped at a card reader at the entrance to a facility or room, the door will open and let the employee in.
4. **Wireless Access Cards** contain chips that establish contact with a wireless card reader without the need to physically attach the card to the machine. The card reader senses the card automatically and allows the user who is in possession of the card to enter the door, which opens automatically.

# Physical and Technical Controls



5. **Locks and Keys**: One of the oldest access control methods ever used besides security guards. There are two types:
    1. Preset locks, which are legacy locks that require a physical key. If you lose the key, you must change the lock.
    2. Programmable locks that require a person to enter a combination of numbers by pressing the respective keys on the lock's keypad.
6. **Electronic Monitoring and Surveillance Cameras** allow remote monitoring of an area within or outside a facility when a guard isn't present. The video footage of the CCTVs (Closed Circuit Television) is normally recorded and stored for future investigations. A drawback of CCTVs is that they are passive devices that only monitor but don't prevent an intrusion if such happens.
7. **Alarms and Alarm Systems** are closely related to electronic monitoring systems but, unlike cameras, are active devices that will notify whenever there is unauthorized access. Alarms are very similar to Intrusion Detection System (IDS) that can detect any physical intrusion or any other events such as a fire, burglary, smoke, or environmental disturbance such as flooding (when appropriate sensors, such as smoke detectors, are installed.)

# Physical and Technical Controls



Physical Access Control and an illustration of entering/exiting through a door:

https://www.youtube.com/watch?v=2QTFiQVdrgg

# Physical and Technical Controls



Any questions?

# Side Channel Attacks



A sensitive, secret computer installation may be made very secure. It may be surrounded by high electrified fences, employ a small army of guards, be protected by powerful firewalls complemented by watchful system programmers working three shifts, and run virus detection software continuously.

Yet, it is possible to spy on such an installation “from the side” by capturing and listening to information that is continuously and unintentionally leaked by electronic devices inside.

This attack type is called a **side channel attack**, and is one in which the attacker listens to data that the computer or its hardware components 'emit' and uses this information to infer what kind of activity the computer is doing and, if the spying equipment is strong, even infer sensitive data.

Let's explain what we mean by 'emitting' data.

# Side Channel Attacks



When any electronic device is working, as part of performing its duty, it may emit energy, which could be vibration, sound, electro-magnetic signals, radiation, etc.

The frequncy, magnitude, duration, and pattern of such emission may reveal what type of activity this device is engaging in, e.g., when the emission continues for 2 hours non stop at a high magnitude, we can infer that the device is busy on one very long and resource-consuming task.

What's more, when an attacker knows what the device type, make, model, etc. is, with enough observation time (and, of course, with suitable hacking equipment/exploits,) he or she can learn deeper details about what the device is doing, including specific, sensitive data that the device processes at that moment.

- Because the attacker gained enough knowledge about a device model, they can distinguish between subtle enegry emissions and, thereby, learn what data goes through the device!

# Side Channel Attacks



**Examples**:

1. Some of the older computer monitors were built with technology that causes the monitor to emit radiation. An engineer decided to take advantage of this fact; he:
   a. Found an office building whose workers use computers.
   b. Packed an antenna and equipment onto his van and drove it next to that building.
   c. Set up the antenna to be directed towards one of the building's rooms.

   Within half an hour, the engineer succeeded in making a monitor screen in the van show the data displayed on one of the screens in the building. The engineer literally tapped on ALL the data that went through that office computer!

   **Controls**:
   1. Use computers whose screens do not radiate, e.g., LCD screens.
   2. Surround computers with a conductive material to prevent any radiation from escaping.
   3. Guard any empty areas near the building, and ensure that public parking lots are located at a considerable distance away from the building.

# Side Channel Attacks



2. A working CPU emits electromagnetic radiation that can be detected outside the computer, and even outside the building.
   A spy who knows the type of CPU being spied on can execute many programs on the same type of CPU, measure the radiation emitted, and thus associate certain patterns of radiation with certain types of computer operations, such as loops, idle, or input/output.
   Once such an association has been established, the spy can train a computer program to analyze radiation emitted by a spied computer and draw useful conclusions about the activity of the spied CPU at various times.
   A similar situation is possible also because CPUs emit sound.
   **Controls**:
   1. Surround computers with a conductive material to prevent any radiation from escaping.
   2. Generate random electromagnetic radiation and sound near the computer to mask the CPU's radiation and fool spies.

# Side Channel Attacks

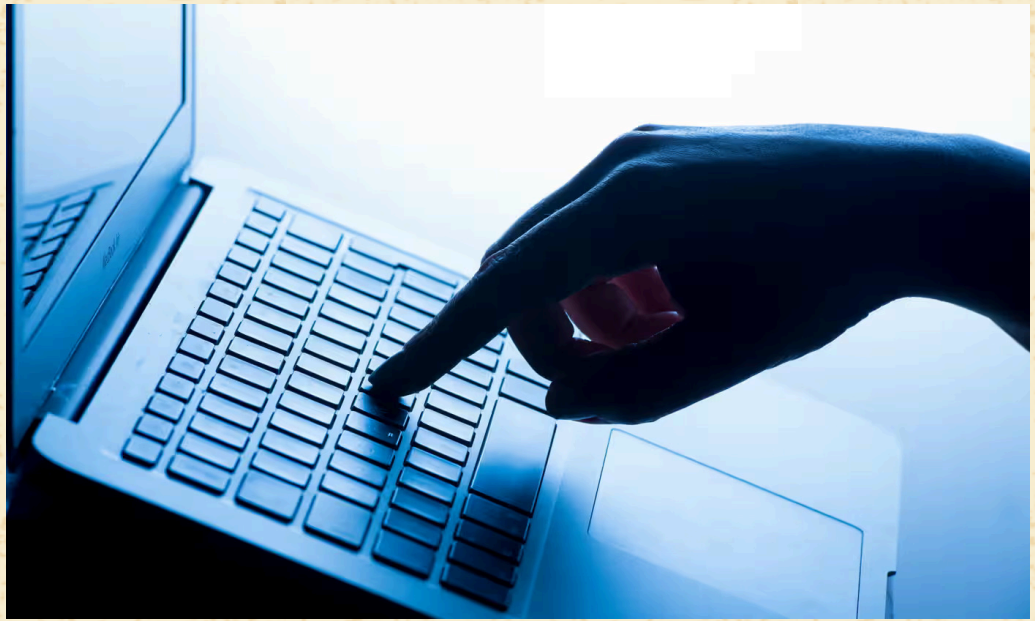


3. An exploit called a **keyboard logger** is a small spy malware whose purpose is to record what keys are pressed and send this log to the hacker. Another spyware may repeatedly capture screenshots and send them to the hacker. Imagine the consequences!
   **Control**:
   1. Keep the computer's anti-virus software working and up-to-date.
4. Another issue with keyboards is that pressing keys emits sound, and, in most keyboards, each key has a unique sound. The difference between the sounds can be noticed by a software of good sensitivity, allowing listening to the keyboard up to 50 feet (!) away.
   1. Purchase a keyboard whose keys produce the same (or at least undistinguishably close) sounds.
   2. Generate random sounds in the vacinity of the keyboard to fool spies.

# Side Channel Attacks



Article: AI can identify passwords by sound of keys being pressed, study suggests



Pressing keyboard's keys. Photo by Dominic Lipinski/PA.

# Side Channel Attacks



A YouTube video about such 'acoustic' keyboards:

https://www.youtube.com/watch?v=ekZQFBnb5MA

# Side Channel Attacks

***Fun Class Discussion***:

People tend to engage in loud conversations, whether at home or work.

What solution(s) would you recommend to prevent spies from listening to employees' conversations, which, from time to time, contain sensitive data?

Any questions?

# Identification vs. Authentication



In the previous slides, we spoke about physical security and physical access control. Using this knowledge, we are now able to approach the general topic of Access Control, but, first, we need to define two more concepts.

The first, **Identification**, is the process of describing one's identity in a unique way. That is, by presenting a unique trait such as username, EMPLID, other sort of unique ID, etc., that won't confuse one with any other person/program/website.

- Note that this unique trait is publically known: it's not a secret piece of information.

The second, **Authentication**, is the process of proving that a user or application is indeed who they claims to be. That is, authentication is the verification of one's identity.

- If authentication is not strong enough, the system is not secure, because other people or programs will access one's account by **impersonating** (= pretending to be) that user.

# Identification vs. Authentication



Unlike an ID, which is publically known and is easily determined, authentication is based on a secret that only the real user or program know/have.

**Access Control** is defined as a combination of (1) authenticating a user/program/web client and (2) deciding on whether this entity is allowed/disallowed to perform the action it requests to take (log into an account, open a file, update information, etc.)

- Quick heads-up: The process by a server/company of deciding whether someone or something is allowed/disallowed to perform an action is called **authorization**, and we will touch on this concept at the end of Topic 3.

# Authentication



Now, back to the idea of authentication.

A system may choose to authenticate a user in one of 3 ways:

1. Something the user ***knows***.
   - E.g., a password, PIN, answer to a secret question, recognition of a secret image, etc.
2. Something the user ***has***.
   - E.g., a phone, magnetic card, house key, token, etc.
3. Something the user ***is***.
   - E.g., biometrics (fingerprints, face recognition), CAPTCHA (to differentiate between humans and computers,) etc.

Let's analyze each next.

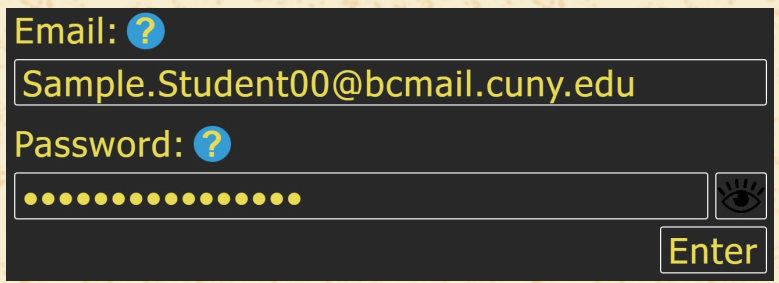
# Authentication: Something One *Knows*



The most common type of knowledge that is used for authentication is a password, which is secret knowledge associated with one's unique identifier.

For example, to log into a website, one must present an ID and a password. The system first checks if a user of the given ID exists in the system. If so, the system will check that the password matches the existing password in the database and, if yes, let the user into their account.

***Fun Class Activity***: When you log into your Poll System account, you are asked to type two items: an email address and a password. Question: which one is used for **identification**, and which one is used for **authentication**?




The Poll System Website login page. Miriam Briskman, CC BY-NC 4.0.

# Authentication: Something One *Knows*



Several significant threats are associated with the use of passwords:

- Shoulder surfing (= spying over the shoulder) at someone's keyboard.
- Reading the password off of printer paper.
- Sniffing the password in transit or in RAM (i.e., accessing main memory while the password is still stored there.)
- Retrieving the authentication database and extracting the password (if the database isn't encrypted.)
- Hacking the password:
  - Guessing out.
  - Exhaustive (= brute-force) attack: checking all possible passwords using a loop.
  - Dictionary attacks: checking out a set of commonly-used passwords.
  - Inferring likely passwords/answers knowing the user's identity, personality, hobbies, etc.
  - Using Rainbow tables: precomputed tables for reversing password encryption.
  - Finding how a password is stored (e.g., encrypted or not,) which helps in using one of the above methods.

# Authentication: Something One *Knows*



Controls of the aforementioned threats:

- [For Users] When typing a password, ensure no one (and no camera) stands right behind you.
- [For systems] Prevent access to Main Memory while a password is stored there.
- [For systems] Save each password in the database only in an encrypted form.
- [For Users] Use strong passwords that:
  - Contain at least 8 characters,
  - Aren't plain dictionary words or object/human names.
  - Don't consist of a detail relating to your identity, e.g., your birthday, address, etc.
  - Contain several types of characters: uppercase and lowercase letters, digits, punctuation, etc.
  - Aren't reused across your other accounts.
- [For systems] Enforce the acceptance only of passwords that meet the above traits of strong passwords.

# Authentication: Something One *Knows*



***Fun Class Activity***:

Consider each of the following two passwords: "`7B&3CK!h+=`" and "`123456`".

1. Which of these passwords is stronger?
2. Which of these passwords will be easily remembered by a user?
3. What trade-off is demonstrated here?

# Authentication: Something One *Knows*



A possible solution to the above tradeoff it to use **passphrases**, which are a sequence of usually 3 or more English words that are used instead of a password. **For instance**: "`unelected-obvious-elbow-disaster-vegan`".

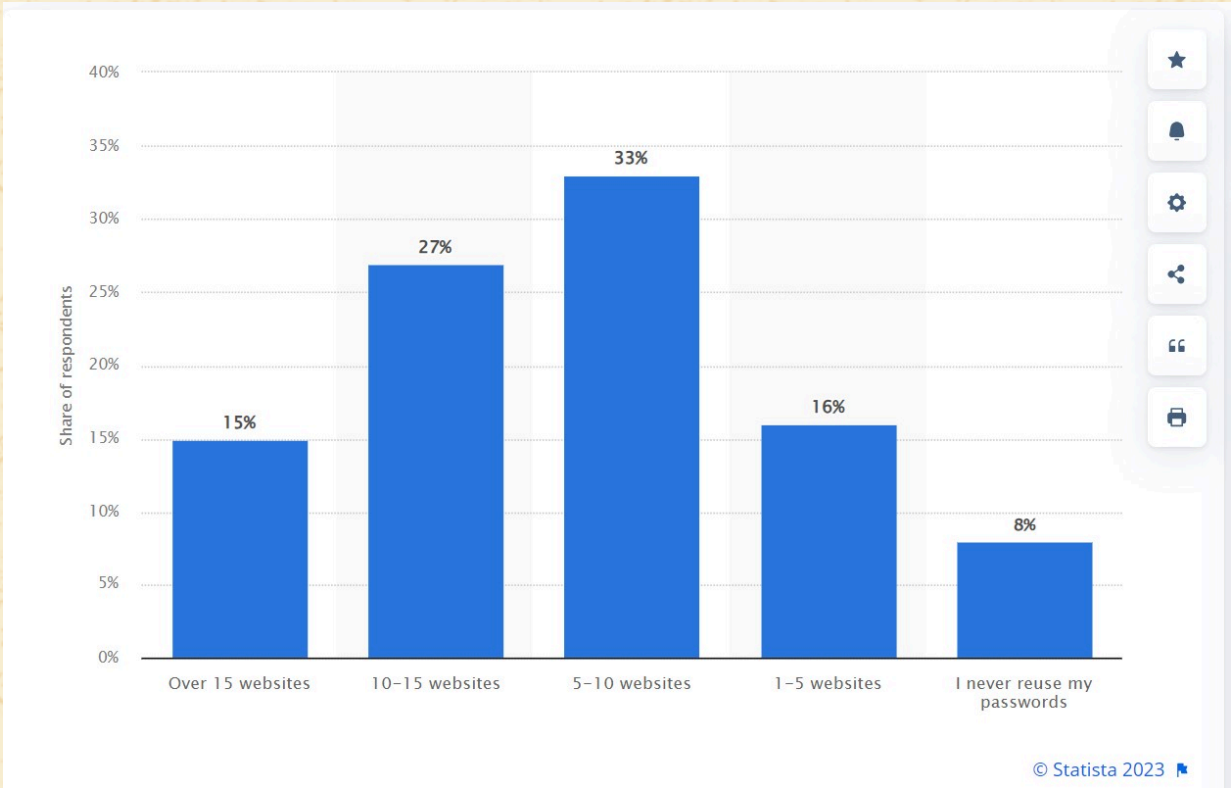
A passphrase will be at least as strong as a password only when the words that build up the passphrase are chosen randomly and independent of one another. [That means that "`red-hot-chili-peppers`" or "`i-love-new-york`" aren't strong enough.]

When a passphrase's words are random enough, the passphrase is considered strong enough AND the user only needs to remember a few words (3 to 5 on average,) instead of remembering more than 10 random characters of a traditional password, which eliminates the tradeoff.

The US National Institute of Standards and Technology (NIST) in its publication on Digital Identity Guidelines encourages the use of strong and long enough passphrases as a substition to using passwords. A summary of this publication can be found here.

# Authentication: Something One *Knows*






Distribution of respondents who reuse their passwords across multiple websites worldwide in 2021. Taken from this Statista webpage.

# Case Study: Ashley Madison Hack of 2015



Remember we promised to return to the case of the leak of Ashley Madison? We keep promises!

In 2015, a hacker group broke into Avid Life Media (ALM), the firm that owned Ashley Madison, and stole sensitive user information, including 36 million passwords of users.

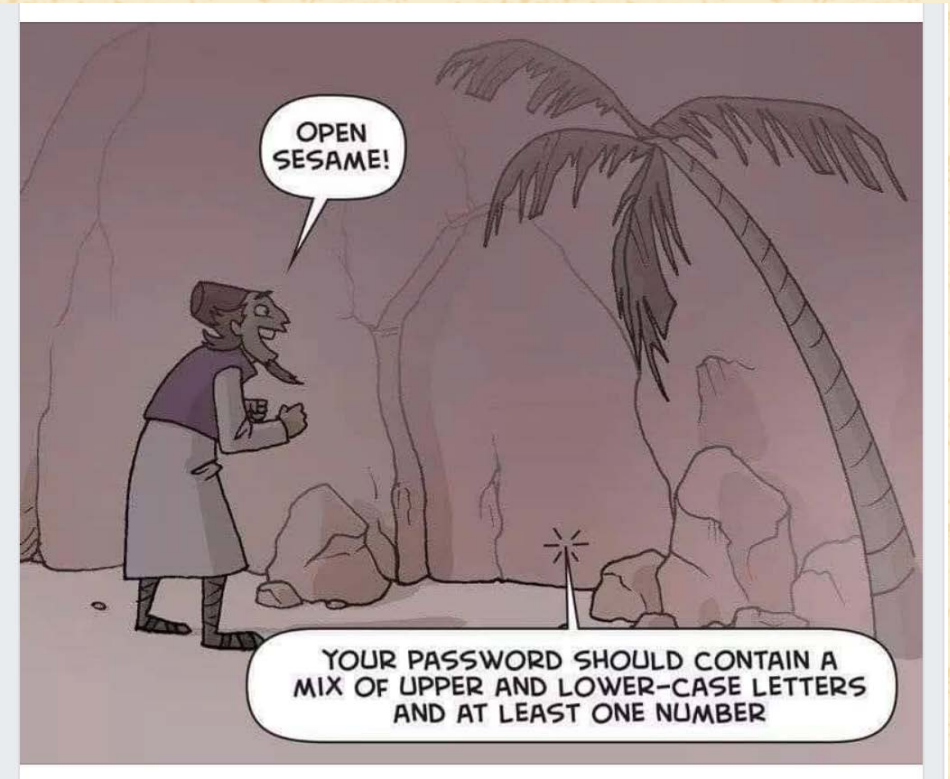
The hackers demanded that Ashley Madison permanently shuts down, but they refused, so the hackers published all the data they stole.

Because Ashley Madison passwords were encrypted using a very weak algorithm called MD5, information security researchers who obtained the stolen data cracked (decrypted) passwords with relative ease. The easier the password was, the faster researchers were able to crack it.

Here is a list of the 100 most commonly-used passwords. The most popular password was `123456`.

# Authentication: Something One *Knows*






Open Sesame! Taken from this Facebook meme.

# Authentication: Something One *Knows*



Any questions?

# Authentication: Something One *Has*



Companies are developing various authentication methods, including third-party services for authentication.

One such 3rd-party authentication method is **physical tokens**, which are tangible items, such as cards or devices that contains security information about a user and can be verified by the company's system.

An example of a physical token is RSA SecureID. We'll explain how it works next.


An RSA SecureID token. Taken from rsa.com.

# Authentication: Something One *Has*



The RSA SecureID hardware token works as follows:

1. The device features a display of 6-digit long numbers.
2. Every 60 seconds, the device generates a distinct, unpredictable number. That is, the number of the screen changes every minute.
3. The user is asked to enter the current number displayed on the token into the authentication application (e.g., the login page of a website.)

Each RSA SecureID token is paired with a single user: the token information is linked with the user's account/details so that, whenever the user is asked to be authenticated, the system would know what number to expect at any moment.

An RSA SecureID token belongs to the authentication category of "something that one *has*" because the user must have a physical possession of this token to successfully authenticate.

# Authentication: Something One *Has*



Remember there is a trade-off with password ease-of-memorization vs. strength? We have a trade-off with physical tokens, too:

- The user needs to carry the token at all times.
- If the token is stolen or lost, the user will not be able to authenticate until the token is replaced with a new one.

As such, an alternative to using hardware tokens (= 'hard tokens') is using **software tokens** (= 'soft tokens'), which are phone applications that perform the same exact actions as hard token (but can be conveniently downloaded from the app store!)

**Examples of soft tokens**: the soft token version of RSA SecureID (links: Google Play Store | Apple App Store), the Google Authenticator app (links: Google Play Store | Apple App Store), and the Microsoft Authenticator app (which you can use to log into your College Outlook email account!) (links: Google Play Store | Apple App Store)

# Authentication: Something One *Has*



Watch the Microsoft Authenticator App in action:

https://www.youtube.com/watch?v=PaSaq99c9n8

# Authentication: Something One *Is*



The authentication category of 'something that one *is*', as y'all noticed correctly, is about a unique bodily trait of a person, the best example of which is biometrics.

**Biometrics**, which means 'measurement of physical traits', uses unique physical characteristics of the individual for identification and authentication.

To pass authentication, the collected biometric information must match the existing data in a biometric database.

**Advantage**: It is more reliable for identifying individuals compared to passwords, which can be forgotten or cracked.

**Examples**: fingerprints, hand or finger geometry, eye (retina) scan, iris recognition, vein recognition, voice recognition, handwriting, typing style, facial recognition, DNA matching, and many more!

# Authentication: Something One *Is*



Demonstration of Palm Vein authentication:

https://www.youtube.com/watch?v=IMd2nRYx2Q4

# Authentication: Something One *Is*



Still, Biometric methods have challenges:

- **Intrusive**: Some biometric methods require awkward, privacy-challenging scans, e.g., DNA matching.
- **Expensive**: Accurate readers/scanners require costy hardware.
- **Overly sensitive**: Even a minor change (e.g., a wound in one's fingertip) will fail authentication.
- **Collected incorrectly**: The biometrics machine might have stored an incorrect reading of your biometric data in the database (so you will not be able to authenticate.)
- **Slow**: because biometrics methods need to process a great deal of data, the authentication process might be slow.
- **Prone to forgery**: e.g., a hacker breaking into a biometrics database might replace your fingerprint with theirs, or forges your fingerprint with a fake play doh finger from an existing print you left somewhere (e.g., door handle.)

Systems should store biometric data in an encrypted form to prevent hackers from re-using biometric data in case they break into the database.

# Authentication Mechanisms



A few efficiency-boosting authentication mechanisms are widely used. Let's find out more about **Multi-Factor Authentication**:

https://www.youtube.com/watch?v=_3rIQVXGKZc

# Authentication Mechanisms



Now, let's discover **LDAP**, an open-standard protocol for accessing and maintaining directory services and user information:

https://www.youtube.com/watch?v=SK8Yw-CiRHk

# Authentication Mechanisms



**IDAM** (Identity and Access Management) refers to the policies for managing digital identities and providing authentication controls to ensure data integrity:

https://www.youtube.com/watch?v=Uk5vu5xW3V0

# Authentication Mechanisms

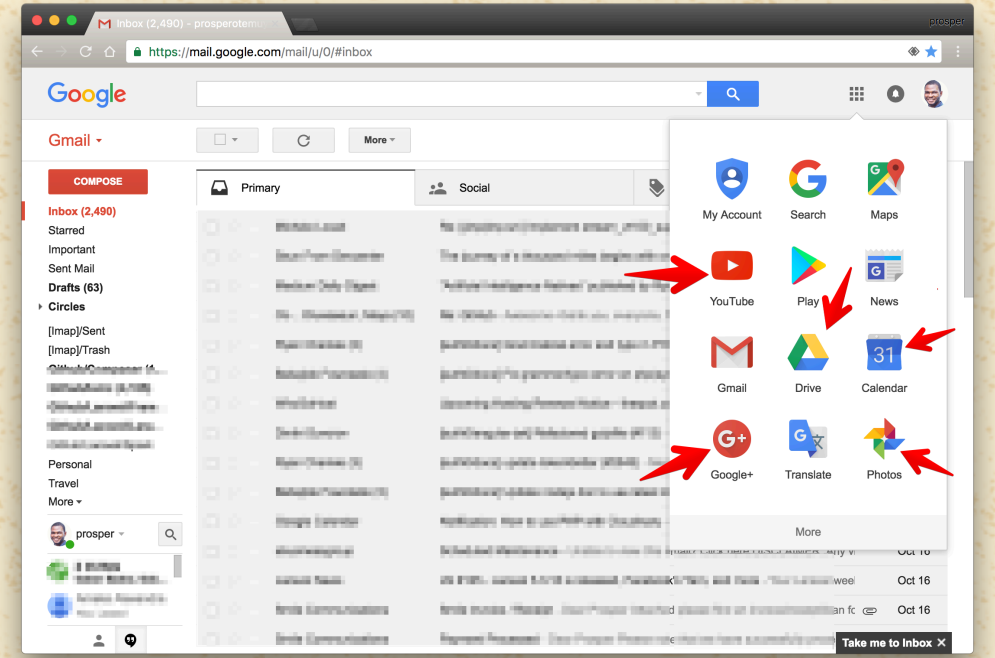


**SSO** (Single Sign On) makes a user's life more convenient by providing access to many apps provided by the same company with a just one login:

https://www.youtube.com/watch?v=YvHmP2WyBVY

# Authentication Mechanisms

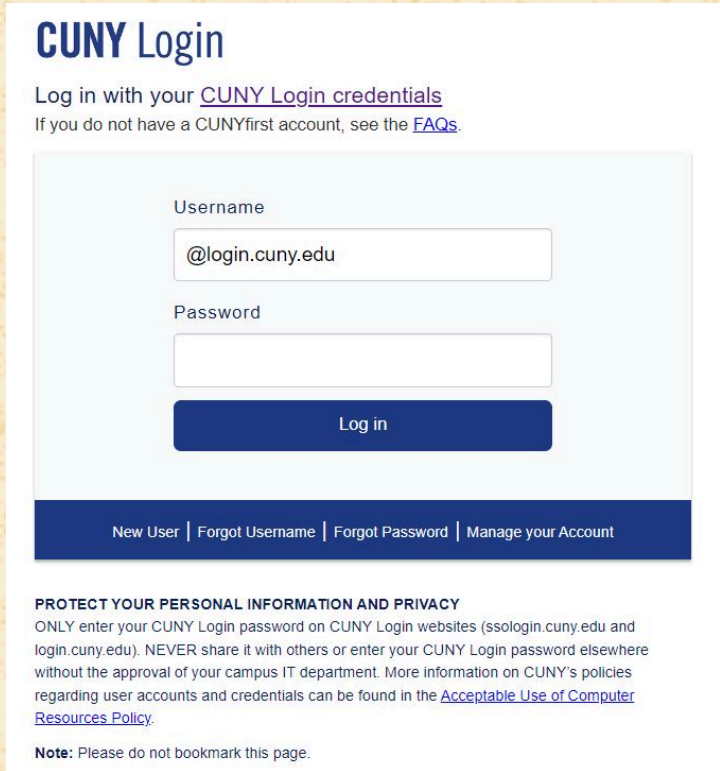





Google apps provided via SSO. Taken from this article.

# Authentication Mechanisms






CUNY's SSO. Miriam Briskman, CC BY-NC 4.0

# Authentication Mechanisms



**OAuth** (The Open Authorization standard) allows users to grant access to 3rd-party websites and applications:

https://www.youtube.com/watch?v=zEysfgIbqlg

Any questions?

# Authentication vs. Authorization



To refresh our knowledge: What is authentication? (How did we define it?)

We promised to introduce another concept that is related to authentication called authorization.

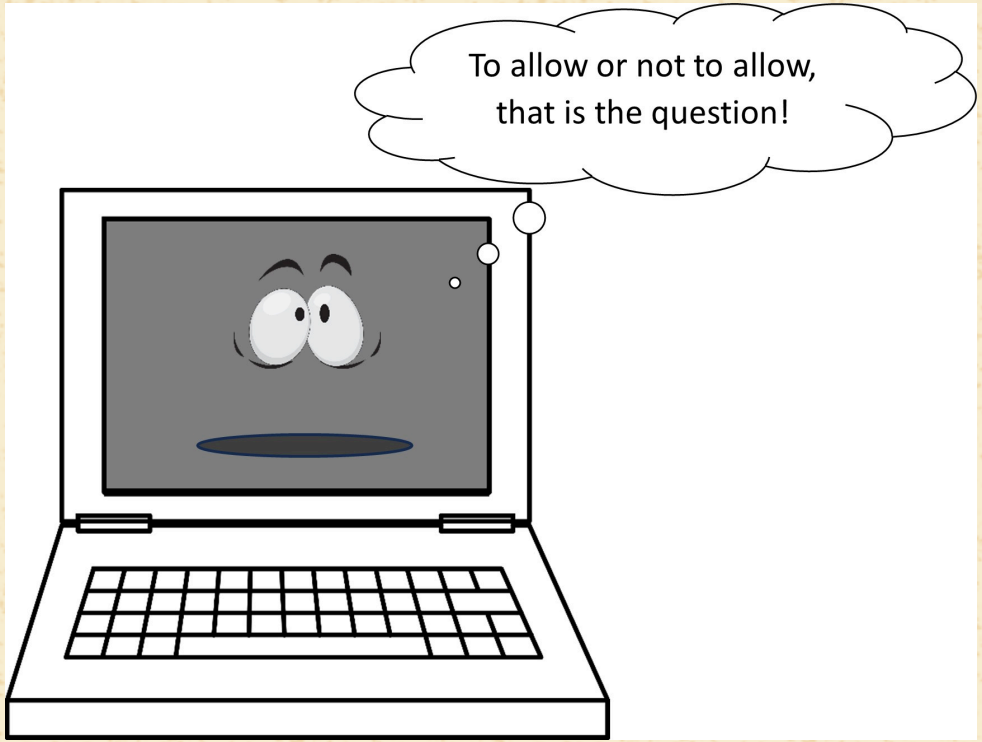
**Authorization** is a decision-making process of a system regarding whether a user, program, website, etc. is or isn't allowed to take an action.

In other words, as a result of an authorization process, the system will return one of two answers: **yes** or **no** as a response to your request to do something (e.g., accessing a file, deleting a file, installing a program, etc.)

Authorization is different from authentication because, for example, a user may authenticate themselves correctly to a system (e.g., enter the correct username + password combo,) but if he or she is not authorized to access the system in question, they will be denied access.

# Authentication vs. Authorization



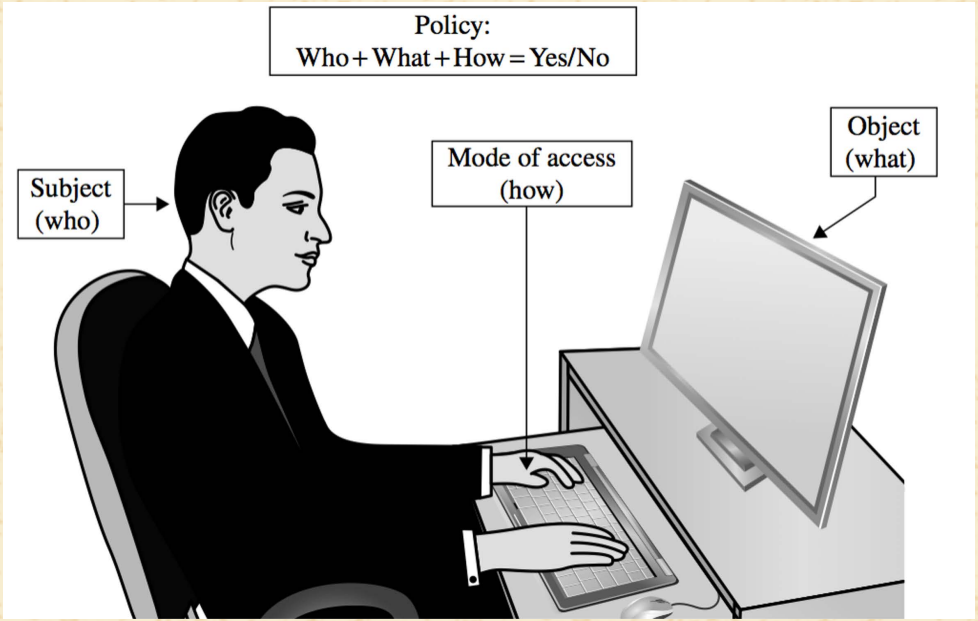



The concept of Authorization: to allow or not to allow! Miriam Briskman, CC BY-NC 4.0, using this computer clipart and this eyes clipart.

# How are Authorization Decisions Made?



A system will allow or disallow an action based on three criteria: (1) Who the user is, (2) What action they want to perform, and (3) In which way they want to take this action:




Making Authorization Decisions. "Figure 1-6: Access Control" (page 10), Pfleeger, Charles P., et al. *Security in Computing*. Prentice Hall, 2015.

# Principle of Least Privilege



Whenever possible, a system should minimize access capabilities depending on the user's role or job.

- For instance, a teacher should not need access to data internal to a human resource system in order to do their job.

This behavior is called the **Principle of Least Privilege**, and its purpose it to prevent possible mistakes, misuse, or abuse by the user of sensitive data.

Systems that define such 'scopes' of access also provide ways of requesting the elevation of one's privileges when needed.

- Example: A regular Linux user can't install programs before requesting to elevate their privilege. To do so, they would call the `su` ('super user') Linux command to temporarily elevate their access capabilities and be able to install programs.

# Authorization



Besides allowing or disallowing access and besides enforcing the principle of least privilege, authorization also consists of tracking user access, both for the detection of authorization issues and to prevent repudiation of a user's past actions. (How did we define non-repudiation?)

One way to keep track of authorization given to a user is by recording granted/denied authorizations in a **log file**. See this CISC 3320 slide with an example of a log file.

Back to slide 55, we can define the conceptual function 'access' as follows:

access(Subject, Object, Mode of Access) = yes/no

That is, the access function will return one of two possible values: yes or no depending on the Subject (a user or program that can perform an action), and Object (an entity, e.g., a file, that can be accessed/modified) and Mode of Access (means by which the Subject requests to access the Object.)

# Authorization



**For example**: The school students Alice and Bob can view (read) their own unofficial transcripts, and Serena, the school principal, can view any student's unofficial transcript. The following statements are true:

- access(Alice, `alice_transcript.pdf`, read) = yes
- access(Bob, `bob_transcript.pdf`, read) = yes
- access(Alice, `bob_transcript.pdf`, read) = no
- access(Bob, `alice_transcript.pdf`, read) = no
- access(Serena, `alice_transcript.pdf`, read) = yes
- access(Serena, `bob_transcript.pdf`, read) = yes

***Quick Recap Questions***:

1. Why is: access(Alice, `bob_transcript.pdf`, read) equal to 'no'?
2. Why is: access(Serena, `alice_transcript.pdf`, read) equal to 'yes'?

# Authorization Mechanisms



We now know that an authorization decision is made based on who the subject is, what the object is, and how the subject tries to access the object. However, how is the subject, object, and access mode information stored? There are a few approaches:

1. An **Access Control Directory** (ACD) is a file that stores filenames, access rules, and pointers to files on a computer system. Each user has their own ACD, which indicates how every file on this system can be accessed.

   The operating system controls all the users' ACD files and prevents users from directly changing them (because this would lead users to grant themselves access that isn't permitted.)

   ACDs create several challenges, such as the length of ACD files, which may be very large especially when files or devices, such as printers are shared by many users (so every ACD file will have to list the same printer.)

# Authorization Mechanisms



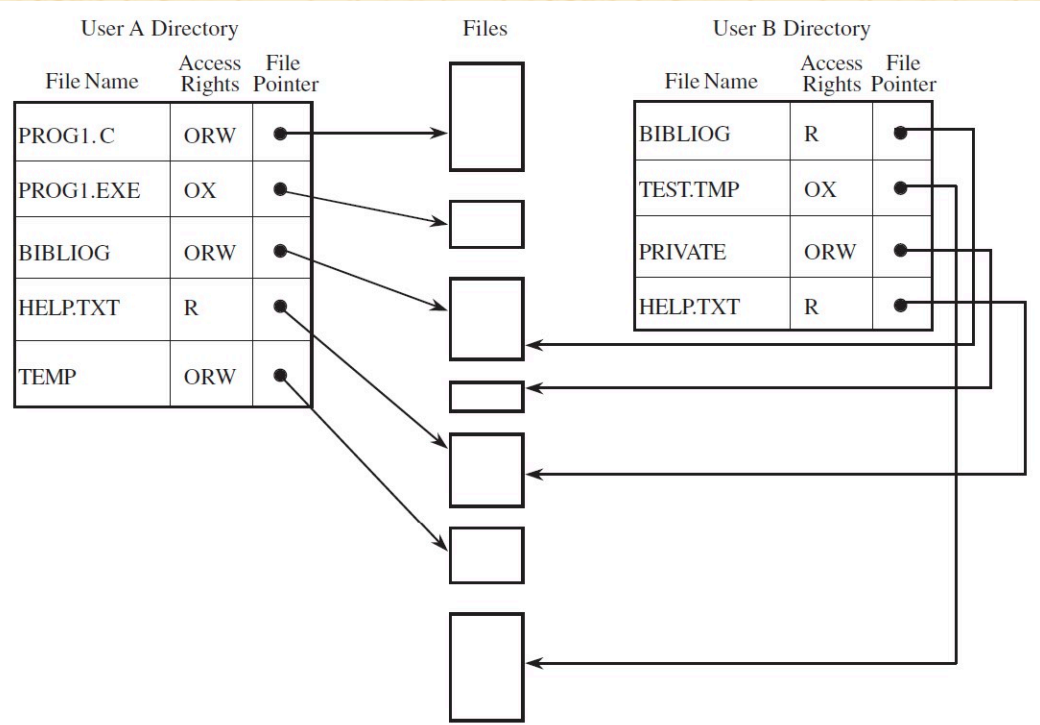



**FIGURE 2-9** Directory Access Rights

"Figure 2-9: Directory Access Rights" (page 77), Pfleeger, Charles P., et al. *Security in Computing*. Prentice Hall, 2015.

# Authorization Mechanisms



2. An **Access Control Matrix** (ACM) is a two-dimensional array that stores access rules for files. Each row represents a user, and each column represents the file under question.
   When some user X wants to access some file Y, the operating system will look into this array and check if user X is allowed to access file Y in the requested way (e.g., read, write, execute, own the file, etc.)

**TABLE 2-8** Access Control Matrix

| | Bibliog | Temp | F | Help .txt | C_ Comp | Linker | Clock | Printer |
|---|---|---|---|---|---|---|---|---|
| **USER A** | ORW | ORW | ORW | R | X | X | R | W |
| **USER B** | R | — | — | R | X | X | R | W |
| **USER S** | RW | — | R | R | X | X | R | W |
| **USER T** | — | — | R | X | X | X | R | W |
| **SYS MGR** | — | — | — | RW | OX | OX | ORW | O |
| **USER SVCS** | — | — | — | O | X | X | R | W |

"Table 2-8: Access Control Matrix" (page 79), Pfleeger, Charles P., et al. *Security in Computing*. Prentice Hall, 2015.

# Authorization Mechanisms



3. An **Access Control List** (ACM) is a one-dimensional array that stores access rules for files. An ACM will be either (1) a list of all the users who can access a specific file, or (2) a list of all the files that can be accessed by a specific user (it depends on what the OS developers prefer it to be.)

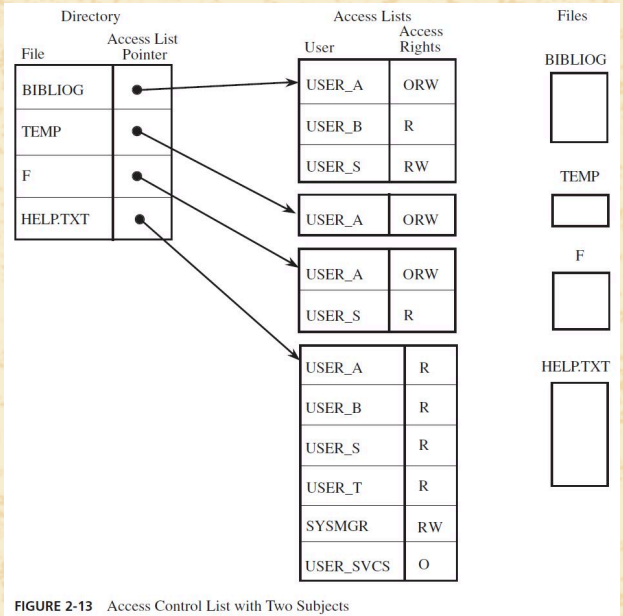


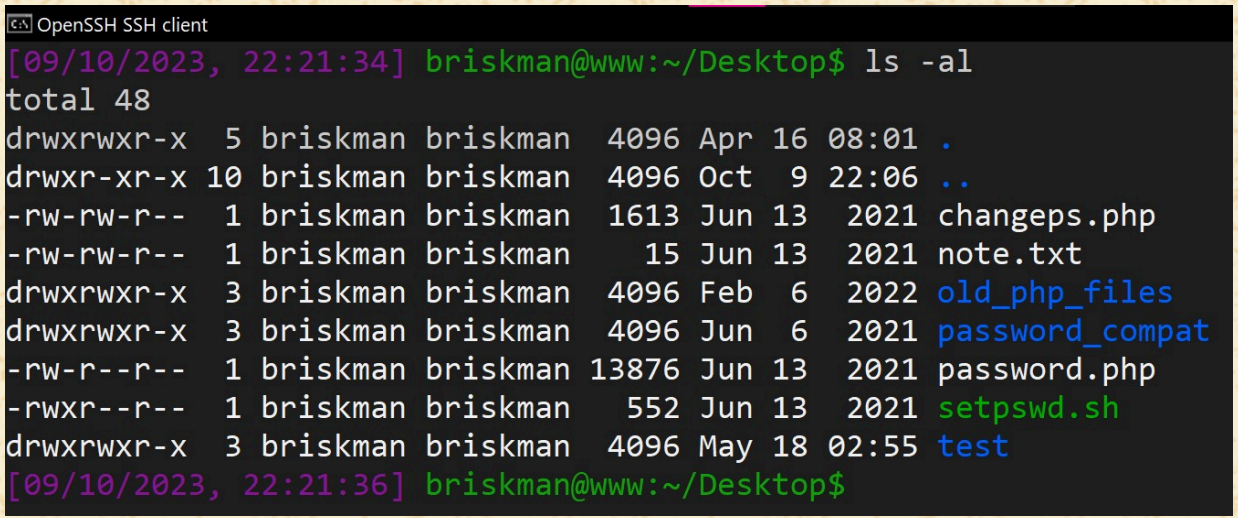

FIGURE 2-13 Access Control List with Two Subjects

"Figure 2-13: Access Control List" (page 81), Pfleeger, Charles P., et al. *Security in Computing*. Prentice Hall, 2015.

# Authorization Mechanisms



3. Two common uses for access control lists are **file system ACLs** and **network ACLs**. The diagram on the previous slide is an illustration of file system ACLs.
   One practical example of the use of file system ACLs is in Linux/UNIX computers, in which each file is associated with 1 owner and a group of users. Each group contains a list of one or more users.




Listing files on Linux using `ls -al`. Miriam Briskman, CC BY-NC 4.0.

# Authorization Mechanisms



3. A **network ACL** will list all the IP addresses that are permitted or not permitted to access a device on the network. Network ACLs act like filters that block access to disallowed users/devices. ACLs consist of the permit/deny rule, source IP address, destination IP address, and the traffic type (IP or TCP).

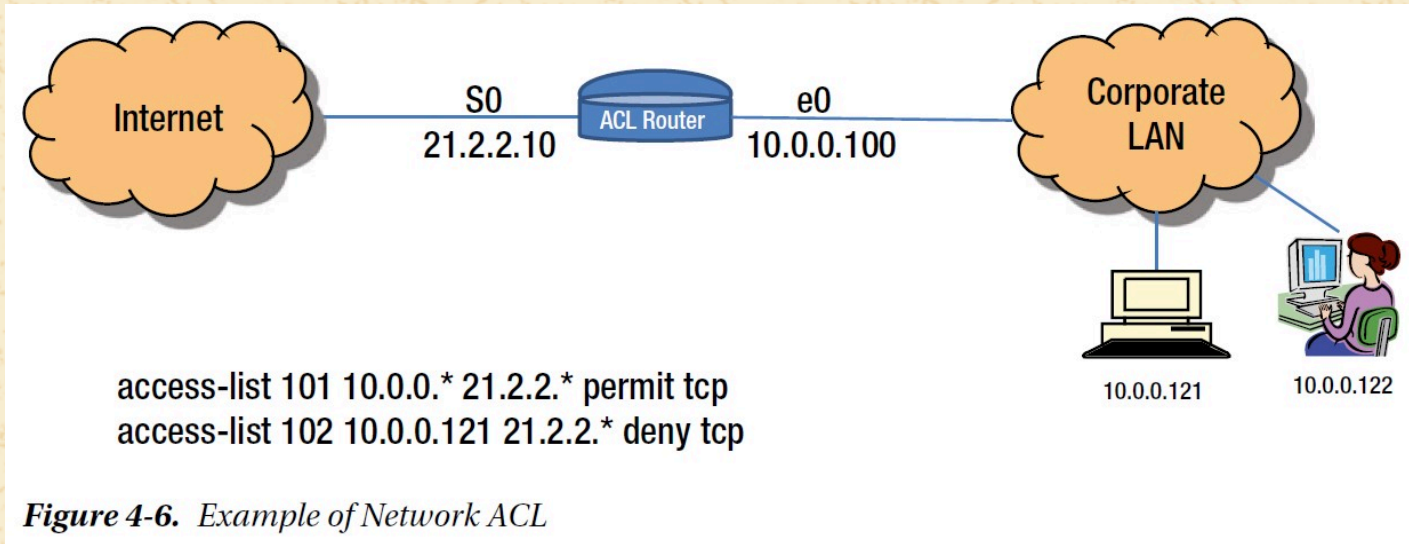



***Figure 4-6.*** *Example of Network ACL*

Example of Network ACL. Figure 4-6: Example of Network ACL" (page 74), Nayak, U., & Rao, U. H. (2014). *The InfoSec handbook: An introduction to information security* (1st ed.). APRESS.

# Authorization Mechanisms

Any questions?

# Access Control Methods: Sources

Rao, Umesh Hodeghatta, and Umesha Nayak. "Chapter 4: Access Controls." *The InfoSec handbook: An introduction to information security.* Springer Nature, 2014, pp. 63-76. URL: https://link.springer.com/content/pdf/10.1007/978-1-4302-6383-8.pdf#chapter+4:+access+controls

Rao, Umesh Hodeghatta, and Umesha Nayak. "Chapter 14: Physical Security and Biometrics." *The InfoSec handbook: An introduction to information security.* Springer Nature, 2014, pp. 293-306. URL: https://link.springer.com/content/pdf/10.1007/978-1-4302-6383-8.pdf#chapter+14:+physical+security+and+biometrics

Salomon, David. "Chapter 1: Physical Security." *Elements of computer security.* Springer Science & Business Media, 2010, pp. 17-35. URL: https://link-springer-com.brooklyn.ezproxy.cuny.edu/content/pdf/10.1007/978-0-85729-006-9.pdf#1+physical+security