# CISC 7320X – COMPUTER SECURITY

## Access Control

# Topics for today

- Why is security important?
- Authentication mechanisms
- Access control

# Why is computer security important?

- Attacks Impact everyone's day-to-day life
  - Millions of compromised computers
  - Millions of stolen passwords
    - Risk of identity theft
- Serious financial damage caused by security breaches



# Security Threats History

- 1990's: fewer attacks, attackers gained fame,
  - Some attacks accidental,
- late 2000's: financially motivated
  - pharmaceuticals, credit card theft, identity theft
  - Phishing evolved into spear-phishing
    - More targeted form of attack
    - Uses target personal information to impersonate a trusted source
- 2010's: politically motivated
  - Government actors: Stuxnet, Flame, Aurora
  - Private activism: Anonymous, Wikileaks

# Security Threats History

- Threats Have Evolved
  - Attackers have become more sophisticated;
- Arms race between attackers and defenders fuels rapid innovation in malware
- Many attacks aim for profit
  - are facilitated by a well-developed “underground economy” or cyber-crime organizations

# Malware Evolution


https://www.slideshare.net/lastlinesecurity/ip-expo-oct-2014

# FAMOUS HACKERS

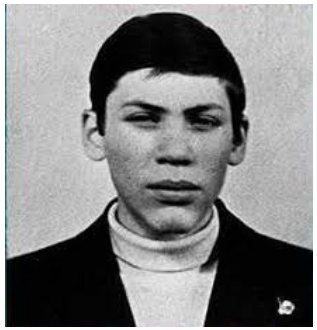
# First Internet Worm

- Robert Tappan Morris was a grad student at cornell in 1988
- Program designed to gauge size of the internet
- Had flows:
  - computers could be infected multiple times
    - each infection caused the computer to slow down more
- Result: 600 computers rendered unusable
- Was prosecuted
  - became the first person convicted under the then-new Computer Fraud and Abuse Act

# KGB Attack on US Computers

- Markus Hess, a German citizen, was recruited by the KGB to break into US army computers
- Used German network to attack 400 US army computers (1980'S)
  - As well as machines in MIT
- Eventually detected by Clifford Stoll
  - a Berkeley astronomer/systems administrator
  - Wrote a book about this, "the Cuckoo egg"
- Hess sentenced to three years in prison

# Citibank Hack




- One of the first high-profile financially-motivated hacking
- Vladimir Levin, Russian crime ring leader, gained access to Citibank accounts in 1995
  - Used a computer in London
  - Got access to list of customer codes and pwds
  - Managed to transfer $3.7 million from these accounts to crime organizations accounts

# NASA Attack

- 16 year old Jonathon James hacked into the Marshall Space Flight Center (1999)
- downloaded proprietary software
- Documents stolen were valued at $1.7 Millions
- Forced NASA to shut down its computer systems for three weeks
  - cost them about $41,000 to fix

# Sarah Palin's Email Hack

- A 20 year old college student, David Kernell obtained access to Palin's email (2008)
  - by looking up biographical details such as her high school and birthdate and using Yahoo!'s account recovery for forgotten passwords
  - posted several pages of Palin's email online
  - Kernell was sentenced to 1 year in prison

# Target attack


- Hackers stole data during 2013 holiday season
- Breach hits 40 million payment cards
- Target paid $18.5 Million multi-state settlement

# Famouse Hacks

- Famous Hacks

- Most Dangerous Hacks

# AUTHENTICATION AND ACCESS CONTROL

CISC 3325 - Information Security

# Identification vs. Authentication

- Identities are well known
  - Typically not secret
    - Your email ID appears in your email header
    - Your check has your bank account number
- Authentication, on the other hand, should be reliable.
- Identification asserts your identity
  - Authentication confirms that you are who you purport to be
- Identifiers may be widely known or easily determined
  - Authentication should be private
- If authentication is not strong enough => not secure

# Authentication

- The ability to prove that a user or application is genuinely who they claims to be
  - Not just for end-users:
    - For example, a web server and client need to authenticated each other
- Usually the user has no control over the type of authentication

# Impersonation

- Pretending to be someone else
  - Attacker may impersonate user, web client or web server
- Authentication can protect against impersonation

https://cyberbullyingisevil.weebly.com/impersonation.html

# Authenticating Users

- How can a computer authenticate the user?
- Authentication basics:
  - Something you ***know***
    - Password, pin, etc.
  - Something you ***have***
    - House key, ATM card, tokens, etc.
  - Something you ***are***
    - Captcha (differentiating humans and computers), fingerprints, face recognition

https://www.smartpaymentassociation.com/index.php/publications-smart-payment-association/position-papers-smart-payment-association/entry/spa-s-position-on-the-regulatory-technical-standard-on-strong-customer-authentication

# Something the user knows

- What does the user know?
  - Passwords, pins, etc.
  - Security questions
  - Recognition of an image

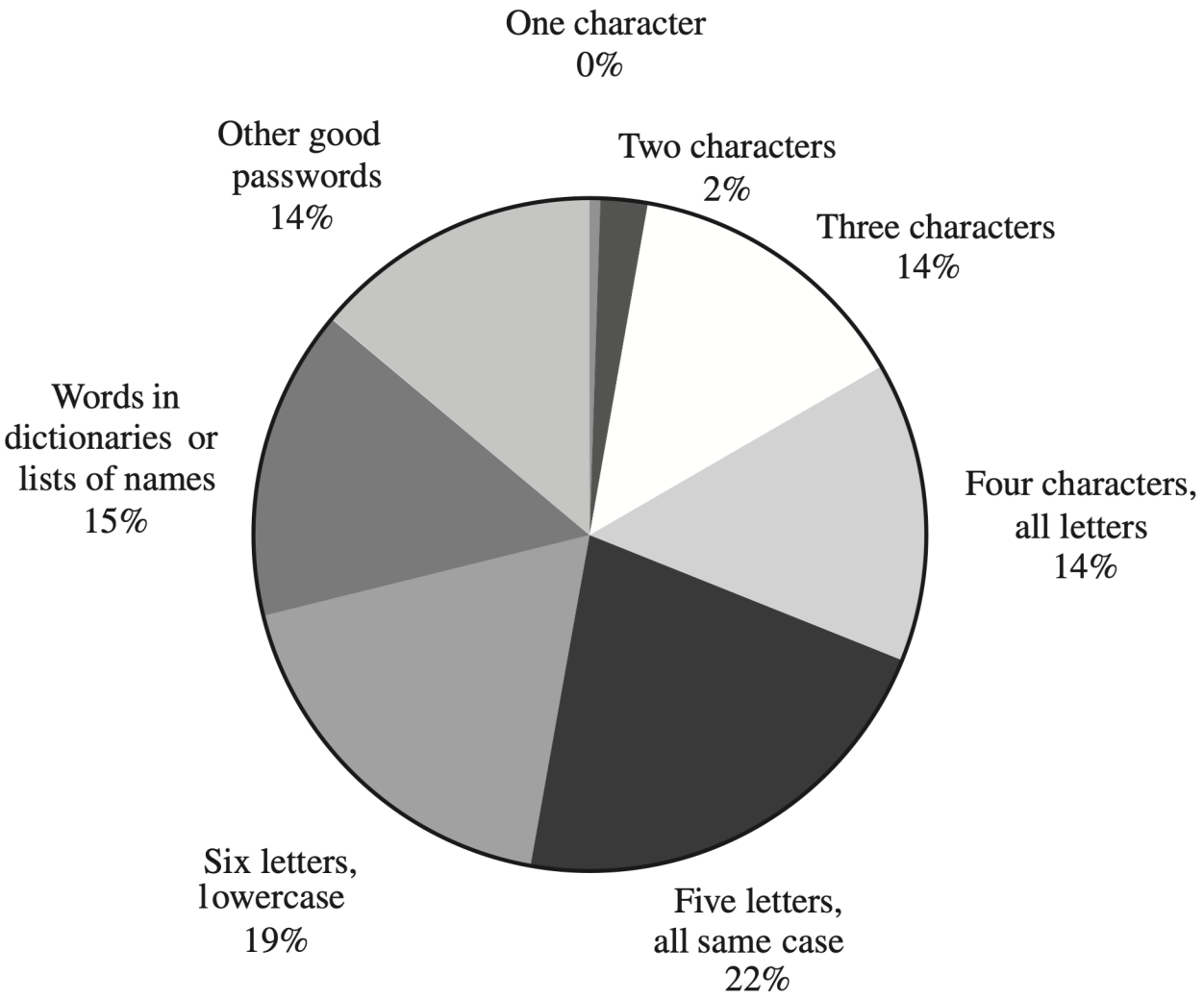
# Something the user knows

- Attacks on “something the user knows”:
  - Dictionary attacks
  - Inferring likely passwords/answers
  - Guessing
  - Detecting how a password is stored
    - Defeating concealment
  - Exhaustive or brute-force attack
  - Rainbow tables
    - a precomputed **table** for reversing cryptographic hash functions

# What is a good password?

- Many passwords are poorly chosen
- A good password has the following characteristics:
  - At least 8 characters
  - Not a word in several languages
  - Must contain several types of ASCII chars
    - uppercase and lowercase letters, digits, punctuation
  - Try not to begin with an uppercase letter

# Distribution of Password Types



# Secure Passwords


- Case study: Ashley Madison hack [2015]
- 36 million hashed passwords leaked
  - Some hashed with MD5
    - a relatively weak hashing algorithm
- Researchers cracked passwords
  - Using automated guessing, other attacks
  - Easier passwords easier to crack

# Ashley Madison Attack

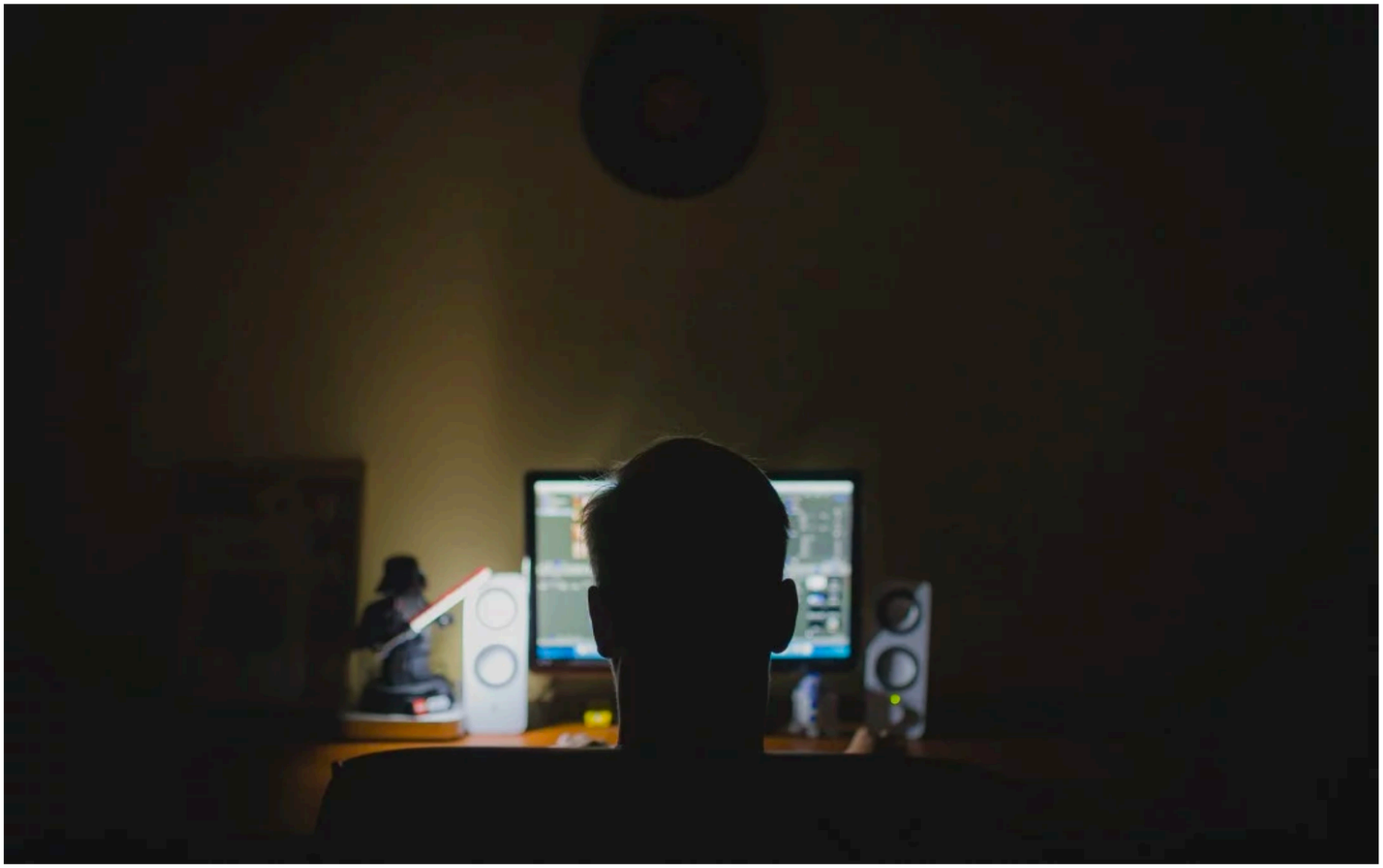
- Most common passwords guessed:


| Password | Times used |
|---|---|
| 123456 | 120,511 |
| 12345 | 48,452 |
| password | 39,448 |
| default | 34,275 |
| 123456789 | 26,620 |
| qwerty | 20,778 |
| 12345678 | 14,172 |
| abc123 | 10,869 |

http://www.zdnet.com/article/these-are-the-top-100-passwords-from-the-ashley-madison-hack/

# Secure Passwords

- Recent studies reaffirm the users' weak passwords
  - Show that the vast majority of passwords used on the Internet are extremely easy to crack
- Case studies, such as Ashley Madison, showed that a large number of passwords can be detected
  - Using off-the-market cracking tools



# 63% of Data Breaches Result From Weak or Stolen Passwords

https://info.idagent.com/blog/2017/06/16/63-data-breaches-result-weak-stolen-passwords

# Secure Passwords

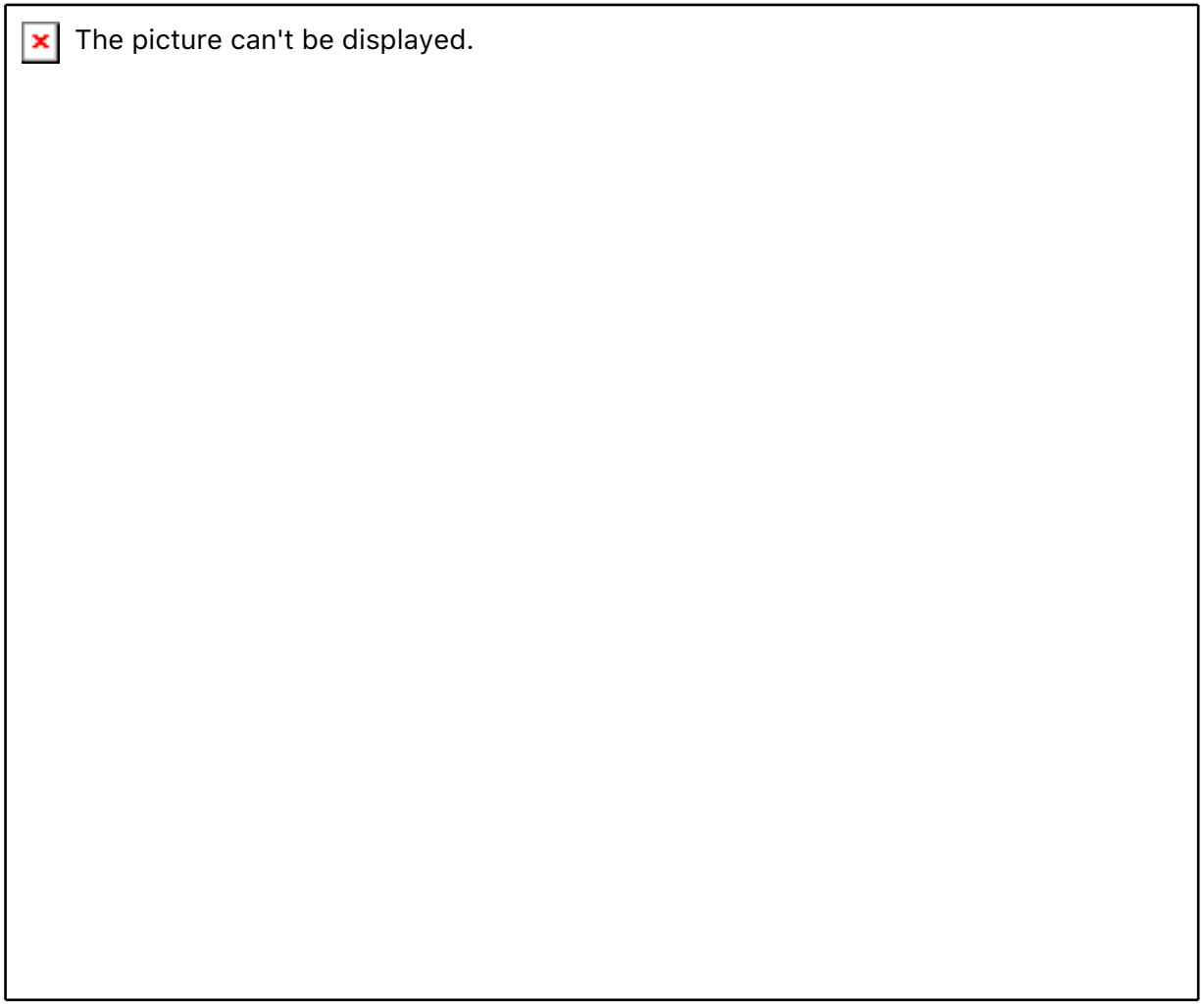

- Conclusions:
  - Weak passwords easier to guess
    - Vulnerable even when hashed
  - Many passwords can be guessed
    - by exploiting the predictability in the way most end users choose passwords
  - Password cracking tools significantly improved
  - Passwords should be hashed
    - Using a strong hashing mechanism

http://www.enterprisefeatures.com/secure-passwords-a-cautionary-tale/


https://www.facebook.com/photo.php?fbid=10156678559998526&set=a.10150307926108526&type=3&theater

# BIOMETRICS AND TOKEN-BASED AUTHENTICATION

# Biometrics: a physical attribute of the user



# Types of biometric authentication

- Fingerprints
- Hand geometry
- Scan of the eye
- Voice
- Handwriting
- Typing style
- Face or facial features

# Problems with Biometrics

- Intrusive
- Expensive
- Single point of failure
- Sampling error
- False readings
- Speed
- Forgery

# Biometrics Passwords

- Recent advances in smartphones have begun to make biometrics cheaper and easier to use.
- Biometrics are still inadequate for extremely sensitive applications
  - but their convenience makes them a great alternative to weak passwords

# AUTHENTICATION MECHANISMS

# Authentication Mechanisms

- Companies are developing authentication methods
  - To be used by third-party applications for authentication
  - So applications do not have to implement it themselves
- Examples: RSA SecureID, Federated Identity Management

# Authentication Mechanisms

- RSA Secure ID:
  - Has a code that changes every 60 seconds.
  - Physical possession of the token should be necessary for successful authentication.

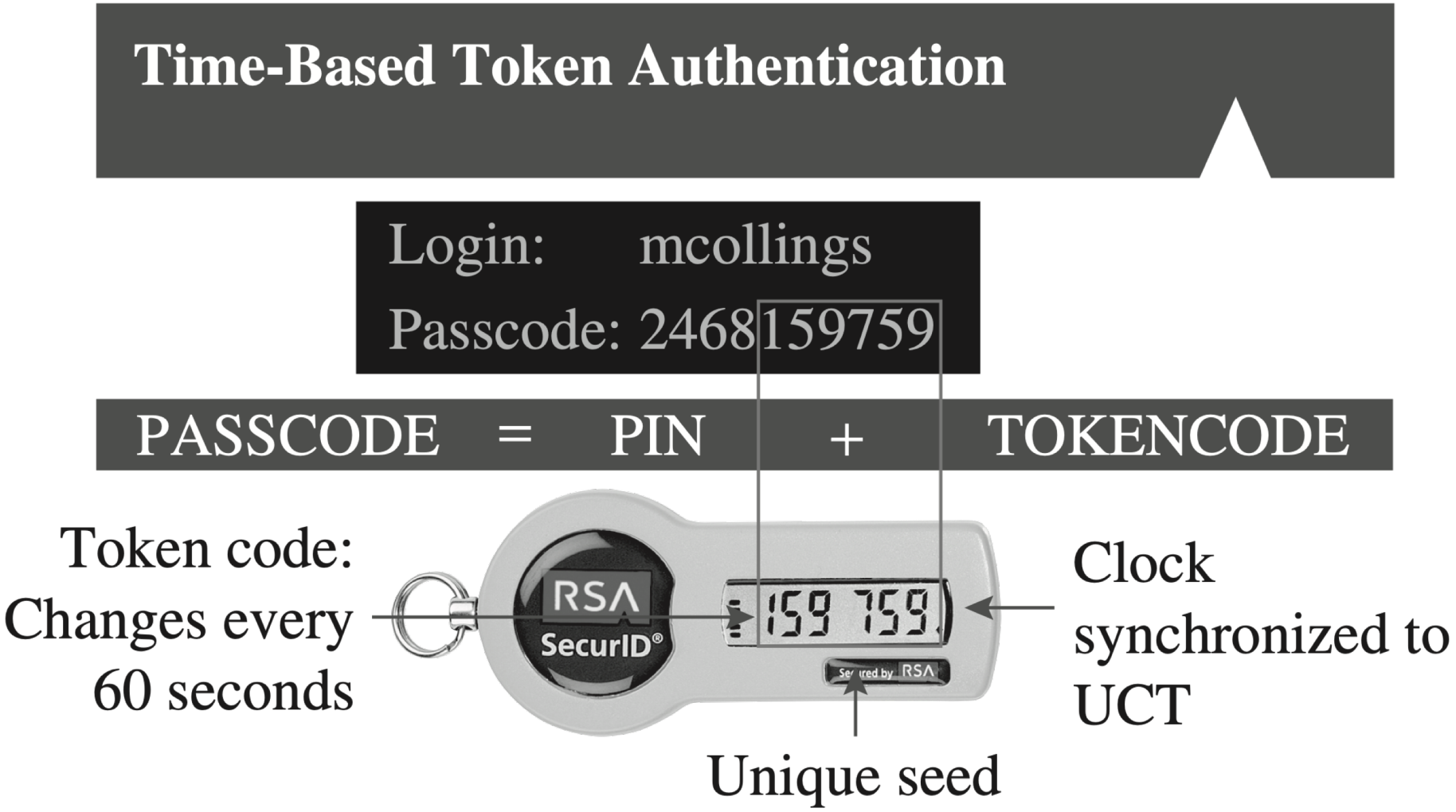
# RSA Secure ID

- Each token generates a distinct, unpredictable series of numbers that change every minute
  - so the authentication system knows what number to expect from your token at any moment.
- User enters the current number displayed on the token to the authentication application

# RSA Secure ID

- Two people can have SecurID tokens, but each token authenticates only its assigned owner
  - Entering the number from another token does not succeed
- The token generates a new number every minute
  - entering number from a previous authentication fails as well

# Tokens: Something You Have



# RSA Secure ID Access

- Uses a variety of optional authentication methods
- Allows end users to choose their preferred authentication method from these options
  - provide the requested level of assurance.
- RSA SecurID Access remembers the preferred method for future authentication requests

# RSA Secure ID Access

- RSA Secure ID Access

# Multi-Factor Authentication

- So far we discussed single-factor authentication
- Has some disadvantages:
  - A token works as long as you don’t give it away
  - Password can be guessed or eavesdropped on
  - We can compensate by combining both

# Multi-Factor Authentication

- Driver license combines two authentication methods:
  - What you have
    - the card itself
  - Who you are
    - Your picture, signature

# Multi-Factor Authentication

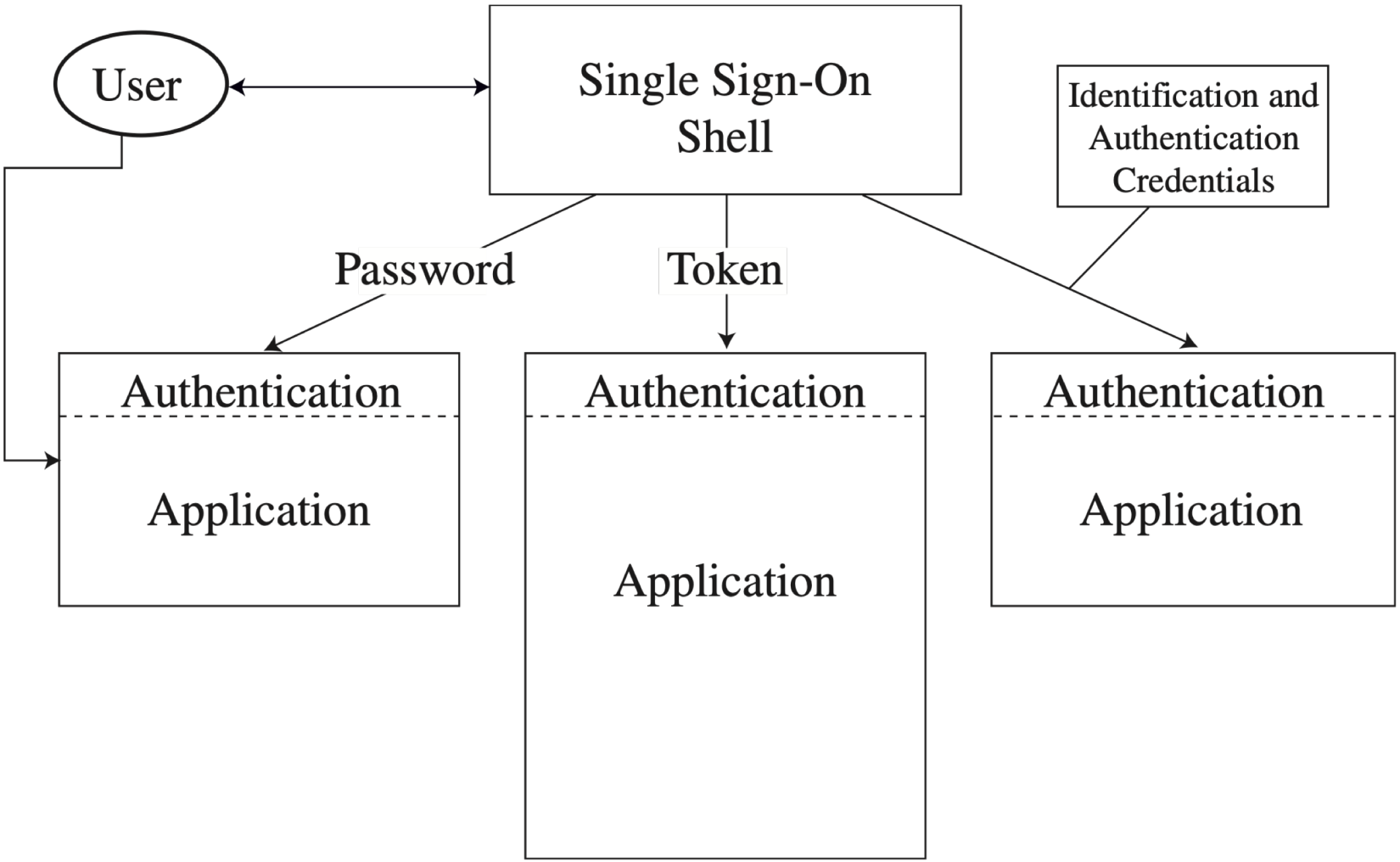
- Authentication may use more than one factor
  - Two, three, etc.
  - We assume that two are better than one
  - However, usability has to be maintained!
    - Not to cause inconvenience to user

# AUTHENTICATION MECHANISMS

# Authentication Mechanisms

- Single sign-on lets a user log on once per session but access many different applications/systems.
  - Often works in conjunction with federated identity management
    - with the federated identity provider acting as the source of authentication for all the applications.

# Single Sign-On (SSO)



# Single Sign-On (SSO)

- Allows multiple web applications to be accessed simultaneously
  - for instance, browse your Gmail account in one tab and open YouTube in another
  - Online banking: A user can typically move from their checking account to savings account
    - without re-entering login credentials
      - even though these two accounts are very distinct.

# SSO

- SSO

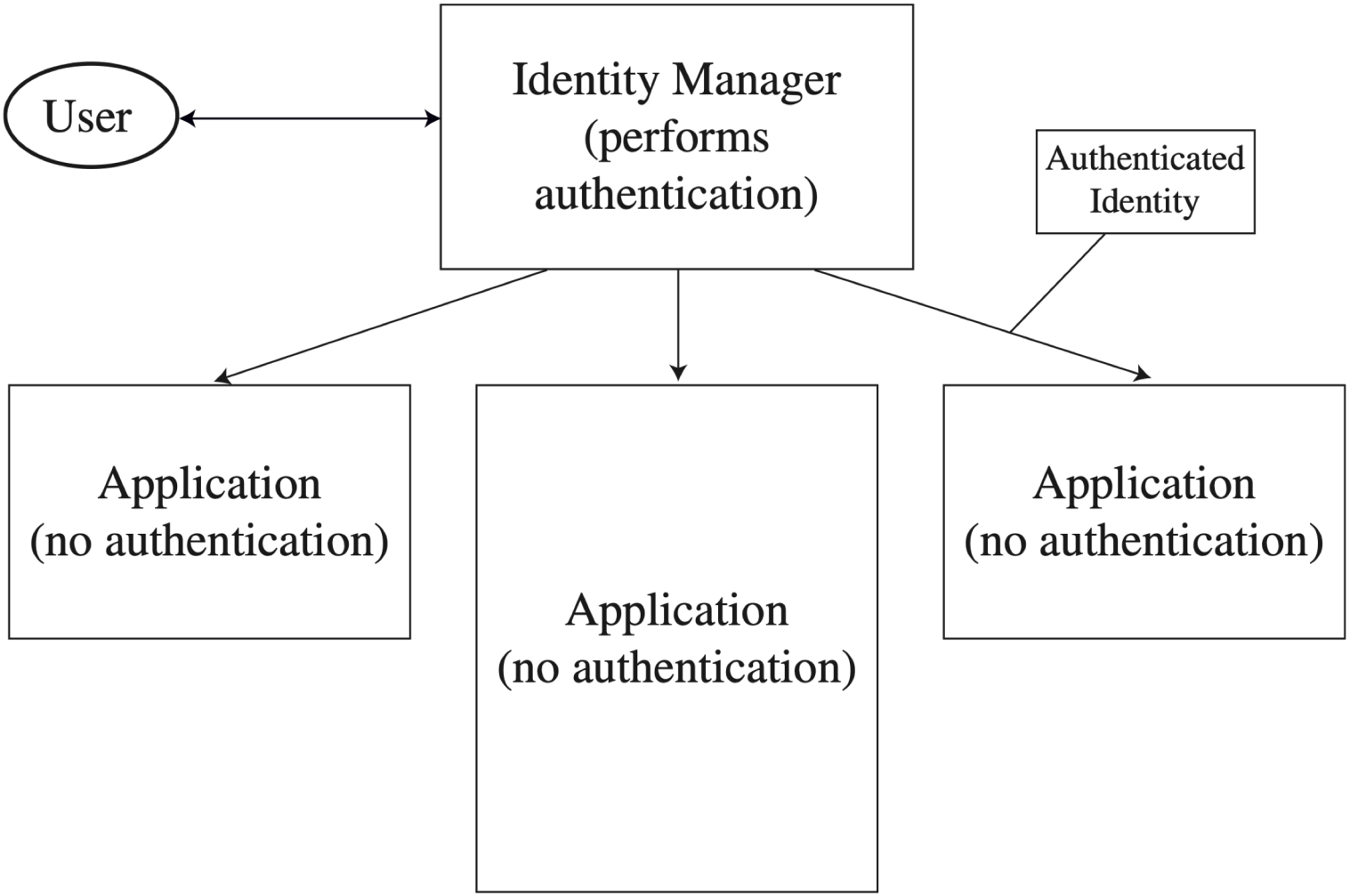
# SSO Disadvantages

- SSO uses of one set of credentials
  - if those credentials are compromised, attacker has access to your entire kingdom
- companies who hold sensitive information should use multi factor authentication (MFA)
- Users still have to remember all the different passwords for each site they're using
  - or resort to a password manage
- Federated Identity Management addresses this and other issues

# Federated Identity Management (FIM)

- A union of separate identification and authentication systems.
- Authentication is performed in one place,
- Separate processes and systems determine that an already authenticated user is to be activated
- A set of agreements and standards
  - enable the portability of identities across multiple enterprises and numerous applications
    - To support large numbers of users.

# Federated Identity Management (FIM)



# FIM

- Federated ID Management

# SSO vs. Federated ID Management

- SSO allows a single authentication credential to access different systems
  - within a single organization
- FIM system provides single access to multiple systems across different enterprises
  - users do not provide credentials directly to a web application, only to the FIM system itself

# Questions?

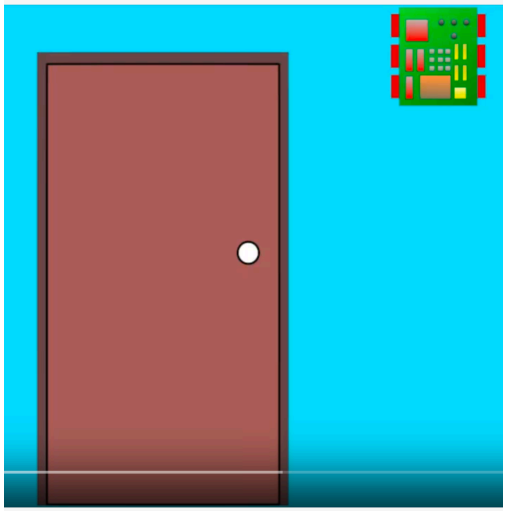
# AUTHORIZATION AND ACCESS CONTROL

# Authorization

- Describing what user can and can not access
  - Different than authentication
- User may authenticate himself correctly to the system
  - However, if he is not authorized to access the system in question, he will be denied access

# Access Control

- Access control is a method of controlling passage into, or out of, an area



- Also: Granting or denying access to information/services

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

# Access Control


- Some resources (files, web pages, …) are sensitive.
  - Need to protect their confidentiality
- How do we limit who can access them?
- This is called the access control problem

http://checkinhere.org/attribute-based-access-control/

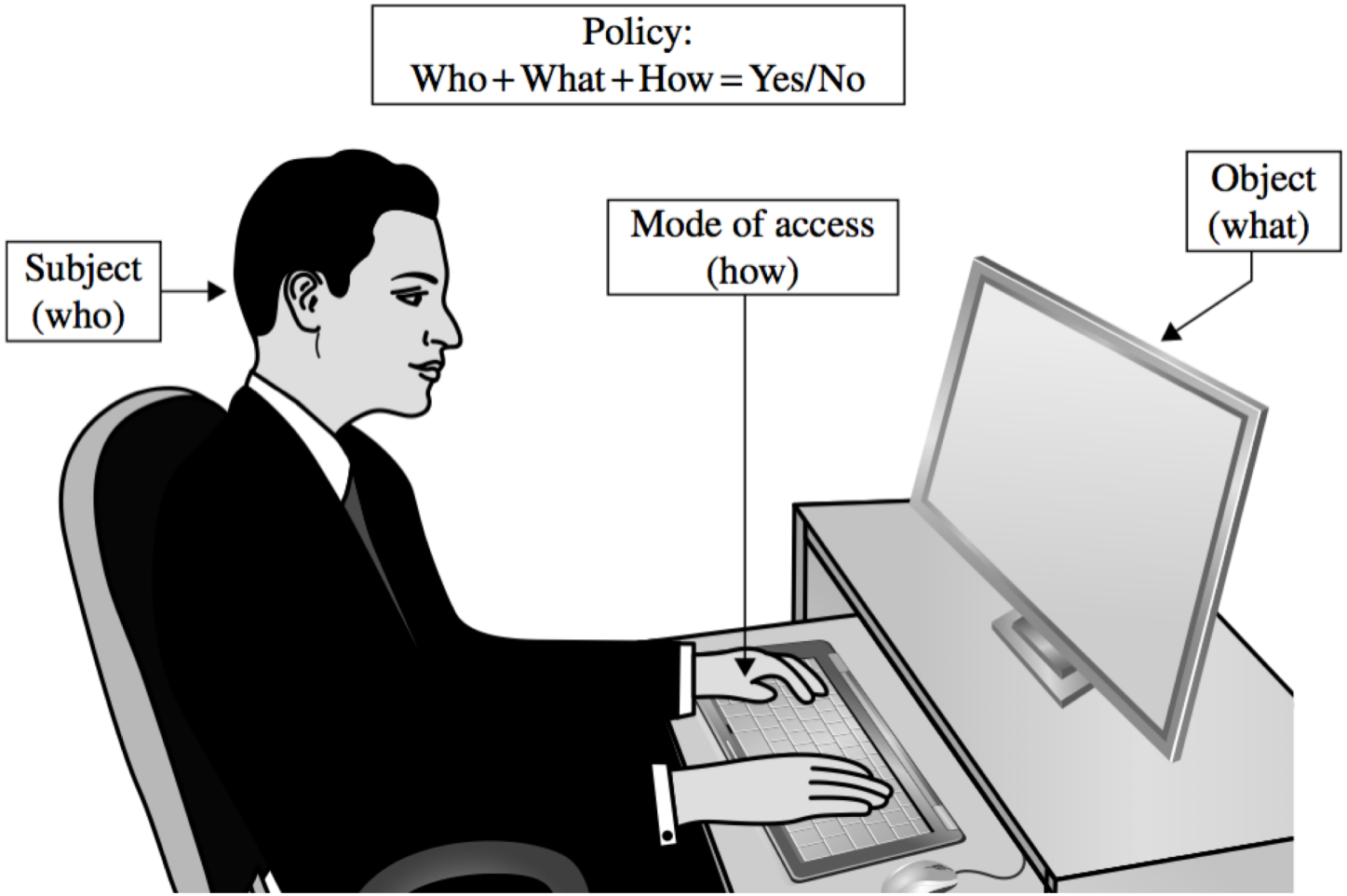
# Access Control


- Basic tasks access control manages:
  - Allow access
  - Deny access
  - Limit access
    - Allowing access to a resource up to a certain point
  - Revoke access
    - Access may change over time

# Access Control Fundamentals

- Selective restriction of access to a place or other resource
- The system makes a decision to grant or reject an access request from an already authenticated subject
  - based on what the subject is authorized to access

# Access Control



# Access Policies

- Goals:
  - Check every access
  - Enforce least privilege
  - Verify acceptable usage
- Track users' access
- Enforce at appropriate granularity
- Use audit logging to track accesses

# Access Control Fundamentals

- *Subjects:* entities that can perform actions on the system
- *Objects*: entities representing resources to which access may need to be controlled
- *Policy:* the restrictions we'll enforce
- *Example:*
  *access*(*S*, *O*) = true  if subject *S* is allowed to access object *O*

# Principle of Least Privilege

- Minimal user profile privileges is set based on users' job necessities
- Applies to users, user accounts, processes, etc.
  - To allow performing needed functionality

- Example: a teacher should not need access to data internal to a human resource system in order to do their job

# Access Control Fundamentals


- Identification and authentication:
  - only legitimate subjects can log on to a system
- Access approval:
  - grant access during operations
    - by associating users with resources they may access
    - based on the authorization policy
- Accountability: identify what a subject did
  - (or all subjects associated with a user)

# Access Control Challenges

- Shift in the types and characteristics of computing devices that are commonly used
  - More smartphones are sold than personal computers
    - As well as other devices
- Data and computation is distributed, and devices are typically always connected via the Internet
- In many systems (e.g., cloud) users do not directly control their data

* What are the most important challenges for access control in new computing domains, such as mobile, cloud and cyber-physical systems? [Bauer et al, 2014]

# Access Control Challenges

- Traditional enforcement of access-control policy may be based on a trusted operating system
  - does not cleanly translate to massively distributed, heterogeneous computing environments
- Environments with many devices that are minimally administered
  - or administered with minimal expertise
- Potentially untrusted clouds hold sensitive data
  - and computations that belong to entities other than the cloud owner

# Access Control Challenges

- Study looked at real-life challenges [Bauer et al. 2009]
- Found few main challenges:
  - Policies are made/implemented by multiple people;
  - policy makers are distinct from policy implementers;
  - access-control systems don’t always have the capability to implement the desired policy.

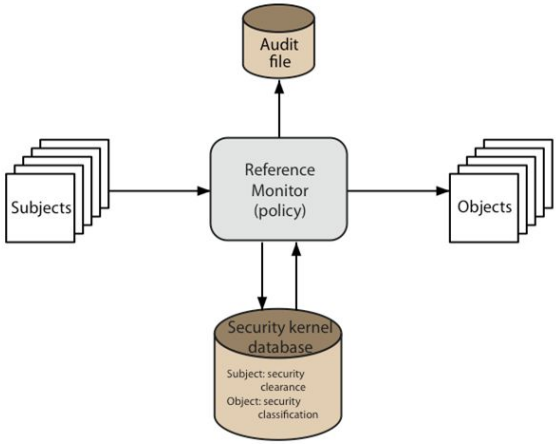
# Questions?

# Implementing Access Control

- Reference monitor
- Access control directory
- Access control matrix
- Access control list
- Capability-based security
- Procedure-oriented access control
- Role-based access control

# Reference Monitor

- Defines a set of design requirements on a reference validation mechanism
- Enforces an access control policy over subjects' ability to perform operations on objects
  - Subjects, e.g., processes and users
  - Operations, e.g. read and write
  - Users, e.g. files, etc.

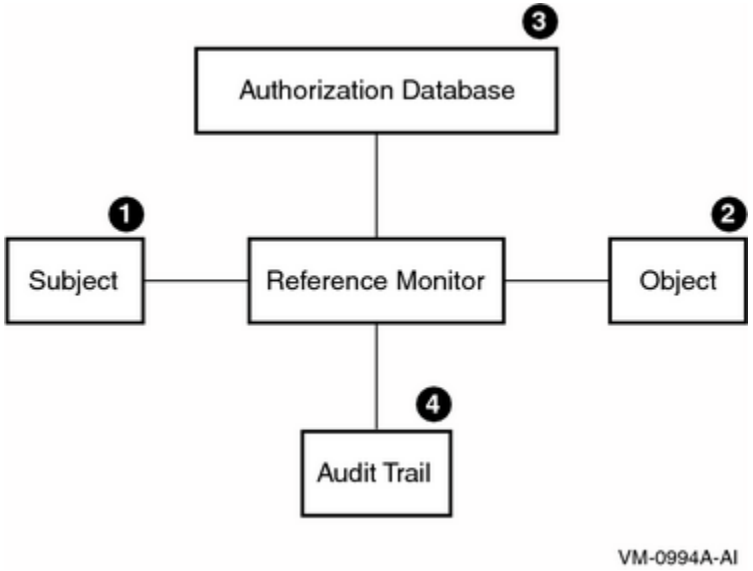



http://slideplayer.com/slide/697485/

# Reference Monitor

- A reference monitor is responsible for mediating all access to data
- Subject cannot access data directly; operations must go through the reference monitor, which checks whether they're OK

# Reference Monitor



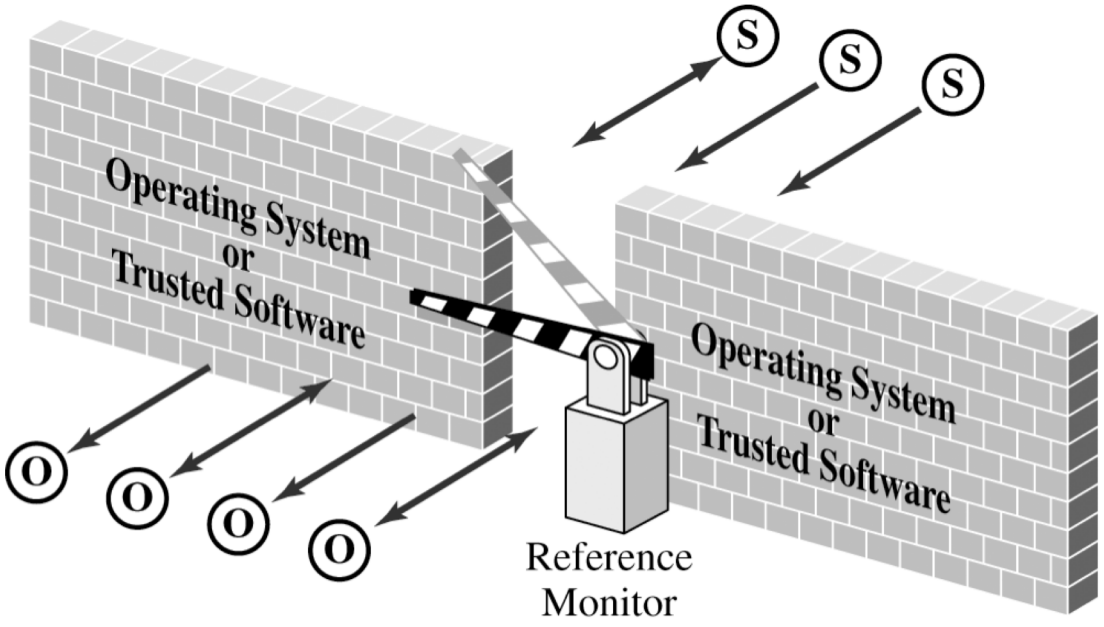

- Authorization Database: Repository for the security attributes of subjects and objects
- Audit trail: Record of all security-relevant events

# Criteria for a Reference Monitor

- Ideally, a reference monitor should be:
  - Non-bypassable: mediate every attempt by a subject to gain access to an object
  - Tamper-resistant: Provide a tamperproof database and audit trail
    - that are thoroughly protected from attackers
  - Verifiable: should be simple and well-structured software
    - so that it is effective in enforcing security requirements
    - Unlikely to have bugs

# Reference Monitor

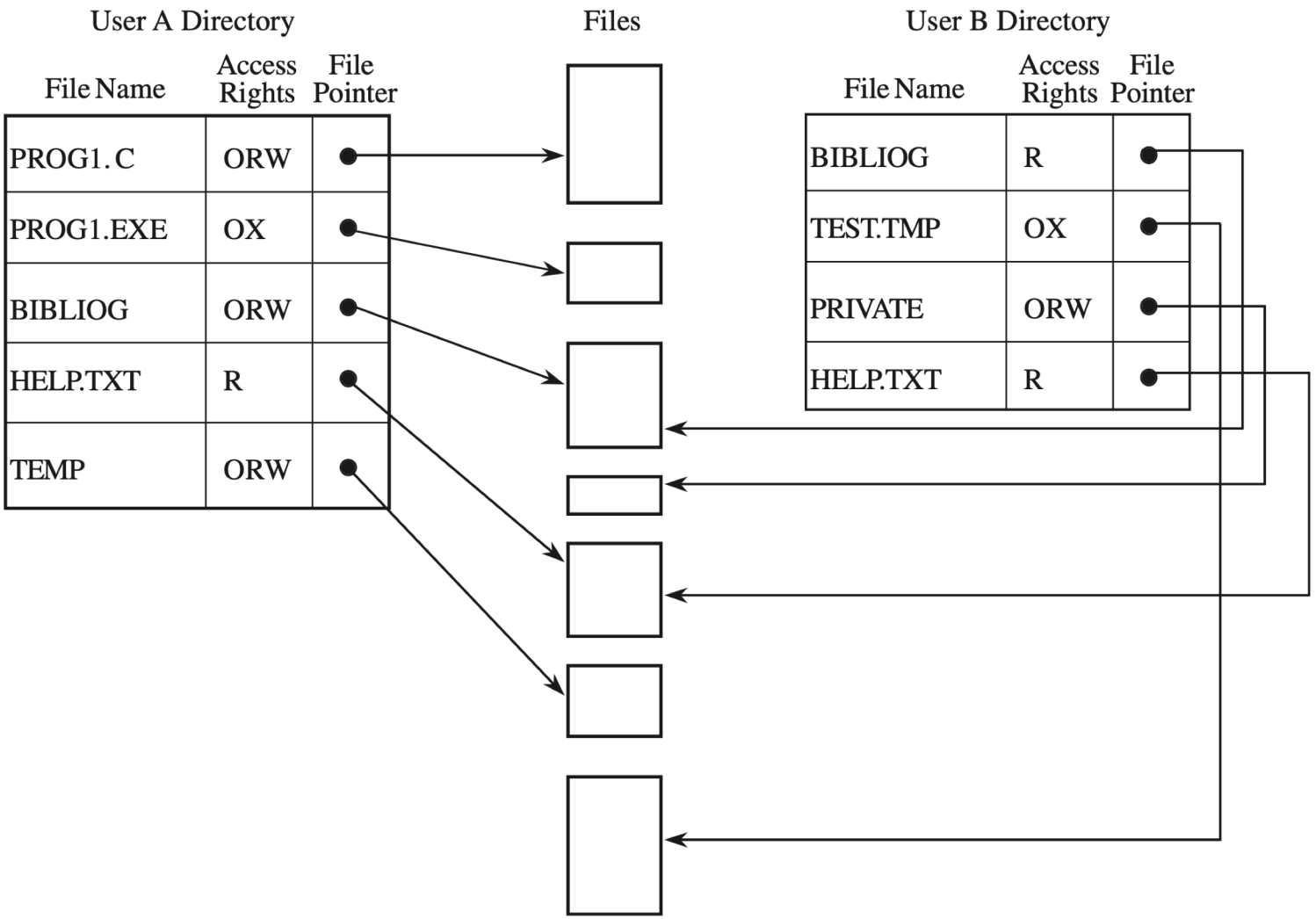
- A reference monitor is the primary access control enforcement mechanism of the operating system



# Access Control Fundamentals


- Examples:
- *access*(Alice, Alice's data) = true
- *access*(Alice, Bob's data) = true
- *access*(Alice, Charlie's data) = false

# Access Control Directory



# Access Control Matrix

- Characterizes the rights of each subject with respect to every object in the system
- Can be written as a rectangular array of cells,
  - one row per subject and one column per object
  - The entry for a particular subject-object pair indicates the access mode
    - that the subject is permitted to exercise on the object
  - Each column is an access control list for the object
  - Each row is an *access profile* for the subject

# Access Control Matrix

| | BIBLIOG | TEMP | F | HELP.TXT | C_COMP | LINKER | SYS_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 | R | W |
| **SYS_MGR** | - | - | - | RW | OX | OX | ORW | O |
| **USER_SVCS** | - | - | - | O | X | X | R | W |

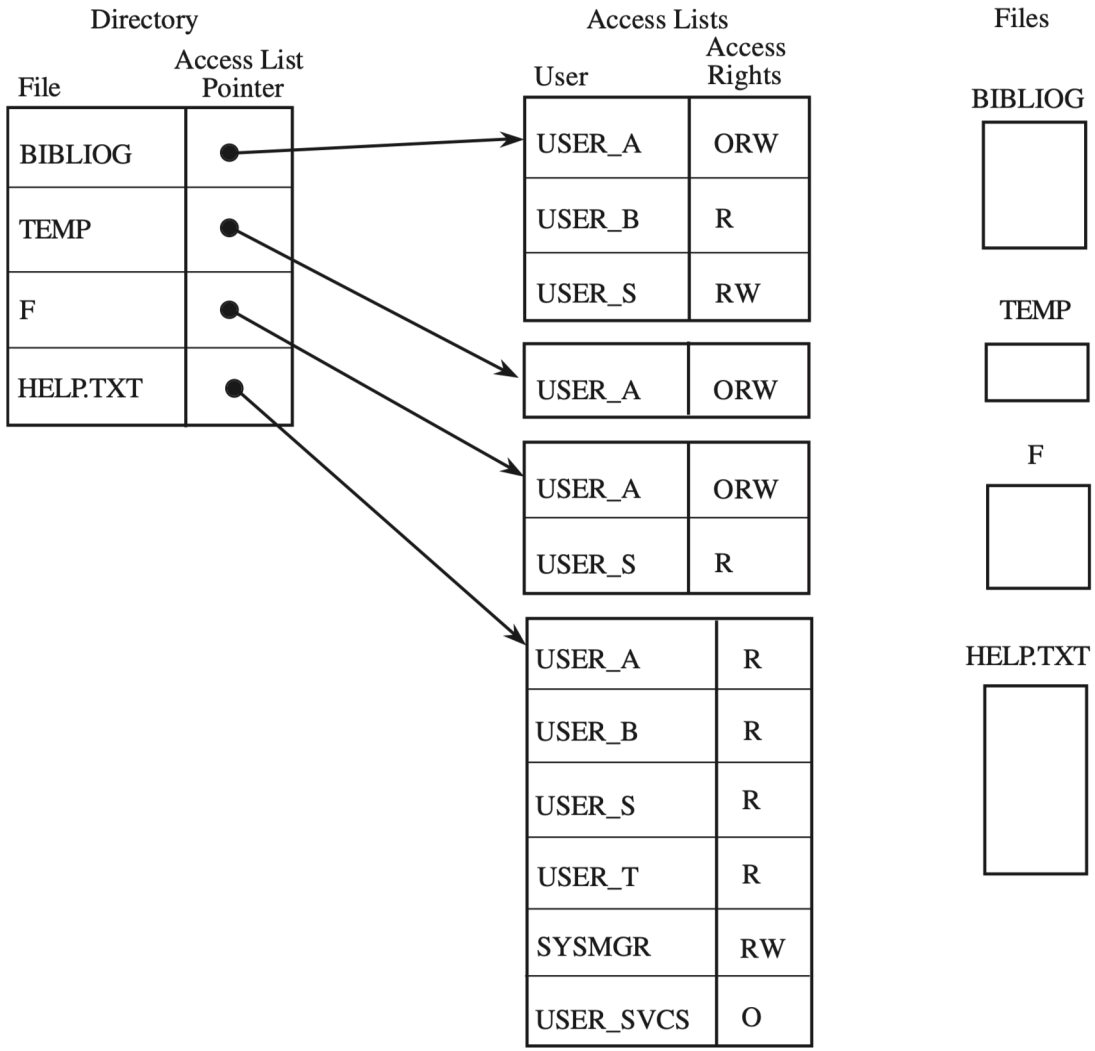
# Access Control List (ACL)

- A list of permissions attached to an object
- An ACL specifies which users or system processes are granted access to objects
  - as well as what operations are allowed on given objects
- Each entry in a typical ACL specifies a subject and an operation
- Example: A file that contains the line:

  Alice: admin, Bob: write

https://internetdevels.com/blog/using-acl-api-to-create-access-control-lists

# Access Control List

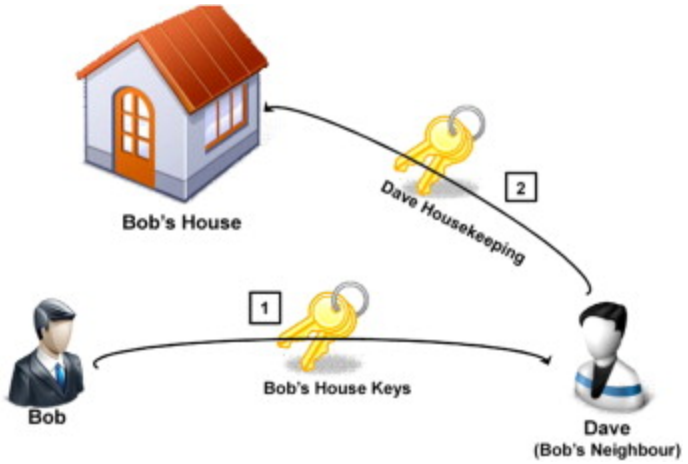


# Capability-Based Security

- Oriented around the use of a token that controls an access
- Based entirely on the possession of the token and not who possesses it
- In a capability-based operating system, the capabilities are passed between processes and storage
  - OS maintains the integrity of those capabilities

# Capability-Based Security

- Example: Bob gives his key to his neighbor Dave while he goes on vacation.




https://www.sciencedirect.com/science/article/pii/S089571771300054X

# Procedure-Oriented Access Control

- Provides a more complex access control
- A procedure that controls access to objects
  - for example, by performing its own user authentication
    - to strengthen the basic protection provided by the basic operating system
- The procedure forms a “capsule” around the object
  - permitting only certain specified accesses

# Procedure-Oriented Access Control

- Procedures can ensure that accesses to an object be made through a trusted interface.

# Procedure-Oriented Access Control Example

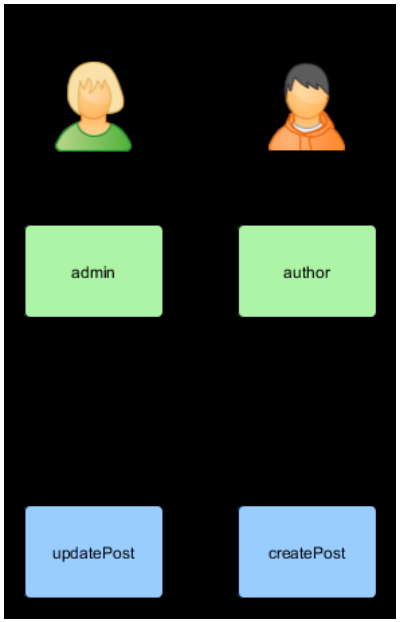
- Example:
  - Neither users nor general operating system routines might be allowed direct access to the table of valid users.
  - The only accesses allowed might be through three procedures:
    - one to add a user
    - one to delete a user
    - one to check whether a particular name corresponds to a valid user.
  - These procedures could use their own checks to make sure that calls to them are legitimate
    - especially add and delete

# Procedure-Oriented Access Control

- Implements the principle of information hiding
  - the means of implementing an object are known only to the object's control procedure
- However, this carries a penalty of inefficiency
  - No fast access checking, even if the object is frequently used

# Role-Based Access Control (RBAC)

- Access control set by an authority designated for the task
- Access is based on the role each subject has in the system




http://www.yiiframework.com/doc-2.0/guide-security-authorization.html

# OAuth

- An open standard
- Allows users to grant third-party websites and applications access to their info
  - Without sharing account credentials
- A form of access delegation
  - Access to the account is delegated to a third party
- Used by Microsoft, Facebook

# OAuth

- Prompts the user to agree to share information of their account with the third-party
  - List which pieces are being requested
- Once confirmed, the identity provider will supply the third-party with access
  - Access data or services offered by the provider directly

# Summary

- Users can authenticate using something they know, something they are, or something they have
- Systems may use a variety of mechanisms to implement access control

# Questions?