CS370 Operating Systems

Colorado State University
Yashwant K Malaiya

Fall 2025 L25
Security and Protection

Slides based on

- **Text by Silberschatz, Galvin, Gagne**
- **Various sources**

Project Notes

- **D3: Project Report Due 12/01/2025** Please see report requirements. Slides (8- 10) should also be ready by 12/01/2025.
- Presentation schedule (12/8 to 12/10) will be shared later.
- **Project Slides** for both options need to be posted in Teams channel *Project Slides and Videos* 24 hours before schedule.
- Research Project **Videos** (7-8 min) should also be posted there by 24 hours before.
- Development Project **Demo schedule** (interactive using Teams, recorded) will be available from the TA assigned, who will email you. Each team should sign up for one 15-min slot.
- Special rules apply to Option B teams in the Online Sec 801, to be shared later.

Project Report Formatting Guidance

Examining Human Elements in Aviation Mishaps: Using Human Reliability Assessment

Rajeshwari Deoraj
CS530: Fault-tolerant Computing
Prof. Yashwant K. Malaiya
Colorado State University
Fort Collins, Colorado
Rajeshwari.Deoraj@colostate.edu

Jaya Veera Surendra Gupta Kurivella
CS530: Fault-tolerant Computing
Prof. Yashwant K. Malaiya
Colorado State University
Fort Collins, Colorado
surendra.kurivella@colostate.edu

***Abstract*—Aviation safety remains a paramount concern as human errors continue to underpin a significant number of incidents. This study examines the application of Human Reliability Assessment (HRA) approaches to the Aviation Safety Reporting System (ASRS) data, aiming to enhance our understanding and mitigation of these errors. By thoroughly examining the ASRS data, we are able to pinpoint common patterns and trends that characterise aviation safety events. Utilizing established HRA methodologies, such as the Human Error Assessment and Reduction Technique (HEART), our**

the development of interventions aimed at amplifying safety measures across various flight scenarios.

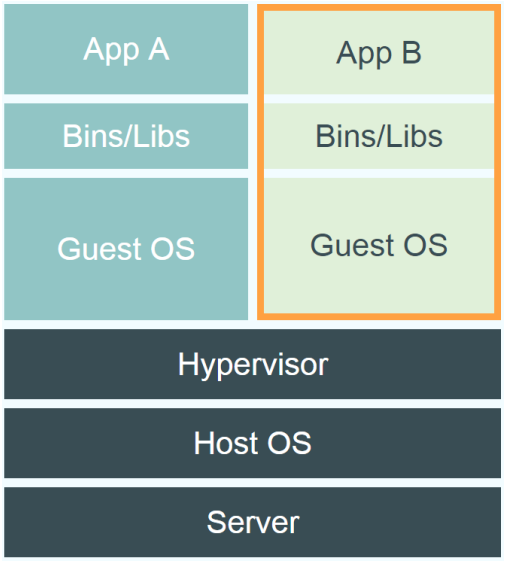
II. Introduction

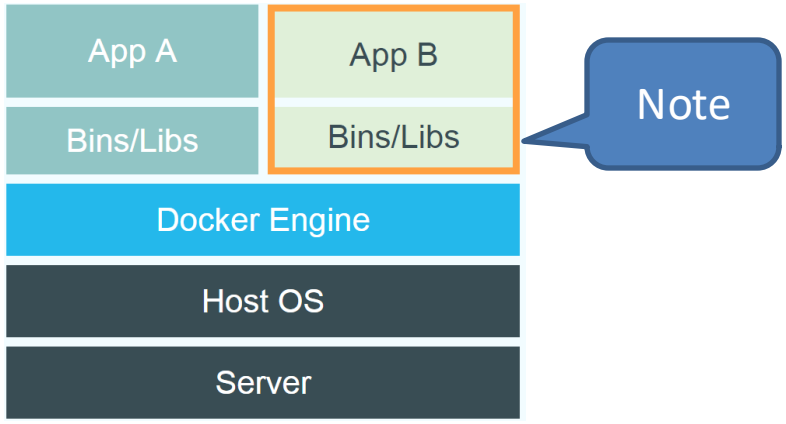
The aviation industry serves as a cornerstone of global transportation, facilitating the movement of millions of individuals daily. As air traffic surges and aviation operations become increasingly intricate, ensuring the safety of air travel takes

Colorado State University

Linux Containers and Docker

- Linux containers (LXC 2008) are “lightweight” VMs
- Comparison between LXC/docker (2013) and VM





- Containers provide “OS-level Virtualization” vs “hardware level”.
- Containers can be deployed in seconds.
- Very little overhead during execution, even better than Type 1 VMM.

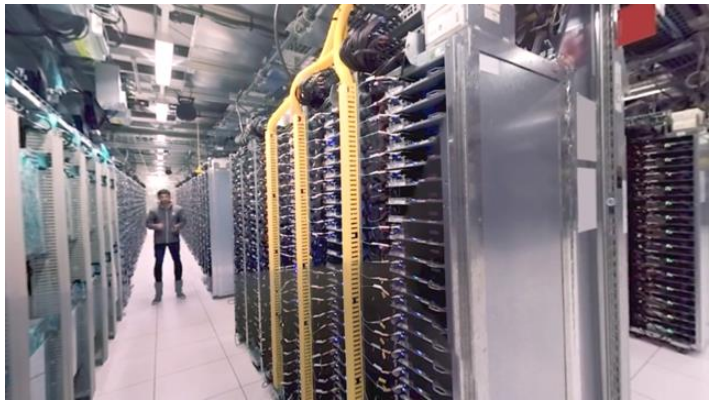
VMs vs Containers

VMs	**Containers** (“virtual environment”)
Heavyweight several GB	Lightweight tens of MB
Limited performance	Native performance
Each VM runs in its own OS	All containers share the host OS
Hardware-level virtualization	*OS virtualization*
Startup time in minutes	Startup time in milliseconds
Allocates required memory	Requires less memory space
Fully isolated and hence more secure	Process-level isolation, possibly less secure

Data Center architecture

Giant warehouses with:

- Racks of servers
- Storage arrays
- Cooling infrastructure
- Power converters
- Backup generators



Or with containers

- Each container filled with thousands of servers
- Can easily add new containers
- "Plug and play"
- Pre-assembled, cheaper, easily expanded

Data Center Challenges

Reliability Challenges

Typical failures in a year of a Google data center:

- 20 rack failures (40-80 machines instantly disappear, 1-6 hours to get back)
- 3 router failures (have to immediately pull traffic for an hour)
- 1000 individual machine failures
- thousands of hard drive failures

http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/people/jeff/stanford-295-talk.pdf

Capacity provisioning

User has a variable need for capacity. User can choose among

Fixed resources: Private data center

- Under-provisioning when demand is too high, or
- Provisioning for peak

Variable resources:

- Use more or less depending on demand
- Public Cloud has elastic capacity (i.e. way more than what the user needs)
- User can get exactly the capacity from the Cloud that is actually needed

Why does this work for the provider?

  - Varying demand is statistically smoothed out over many users, their peaks may occur at different times
  - Prices set low for low overall demand periods

Amazon EC2 Instance types

On-Demand instances

- Users that prefer the low cost and flexibility of Amazon EC2 without any up-front payment or long-term commitment
- Applications with short-term, spiky, or unpredictable workloads that cannot be interrupted

Spot Instances (cheap)

- request spare Amazon EC2 computing capacity for up to 90% off
- Applications that have flexible start and end times

Reserved Instances (expensive)

- Applications with steady state usage
- Applications that may require reserved capacity

Dedicated Hosts

- physical EC2 server dedicated for your use.
- server-bound software licenses, or meet compliance requirements

Amazon EC2 Prices (samples from their site)

General Purpose - Current Generation Region: US East (Ohio)

instance	vCPU	Network Perf. Up to	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage
t2g.nano	2	5 Gigabit	0.5	EBS Only	$0.0042 per Hour
t2g.micro	2	5 Gigabit	1	EBS Only	$0.0084 per Hour
..	..	..	..	..	..
trn1.32xlarge	128	512	512	4 x 1900 NVMe SSD	$21.50 per Hour
trn1n.32xlarge	128	1600	512	4 x 1900 NVMe SSD	$24.78 per Hour

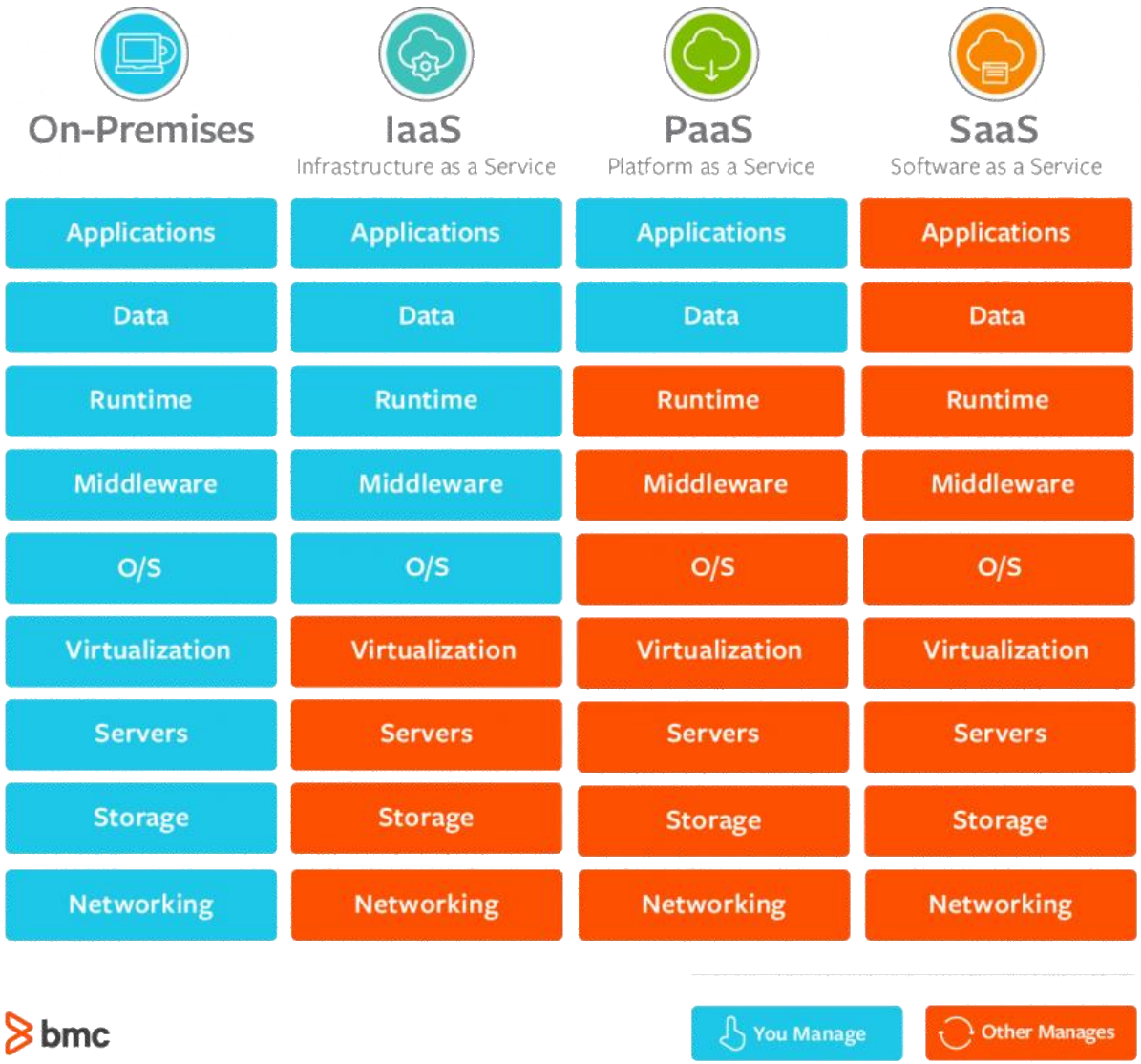
EBS: elastic block store (storage) , automatically replicated

The cloud Service Models

Service models

- IaaS: Infrastructure as a Service
  - infrastructure components traditionally present in an on-premises data center, including servers, storage and networking hardware
  - e.g., Amazon EC2, Microsoft Azure, Google Compute Engine
- PaaS: Platform as a Service
  - supplies an environment on which users can install applications and data sets
  - e.g., Google AppEngine, Heroku, Apache Stratos
- SaaS: Software as a Service
  - a software distribution model with provider hosted applications
  - Microsoft Office365, Amazon DynamoDB, Gmail

The Service Models



https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/

bmc

Colorado State University

Cloud Management models

- **Public clouds**
  - Utility model
  - Shared hardware, no control of hardware,
  - Self-managed (e.g., AWS, Azure)
- **Private clouds**:
  - More isolated (secure?)
  - Federal compliance friendly
  - Customizable hardware and hardware sharing
- **Hybrid clouds**:
  - a mix of on-premises, private cloud and third-party, public cloud services.
  - Allows workloads to move between private and public clouds as computing needs and costs change.

Different Regions to Achieve High Availability (HA)

- AWS datacenters is divided into regions and zones,
  - that aid in achieving availability and disaster recovery capability.
- Provide option to create point-in-time snapshots to back up and restore data to achieve DR capabilities.
- The snapshot copy feature allows you to copy data to a different AWS region.
  - This is very helpful if your current region is unreachable or there is a need to create an instance in another region
  - You can then make your application highly available by setting the failover to another region.

Different Regions to High Availability (HA)

Each region has 2-5 availability zones.

Colorado State University

CS370 Operating Systems

Colorado State University
Yashwant K Malaiya

Security

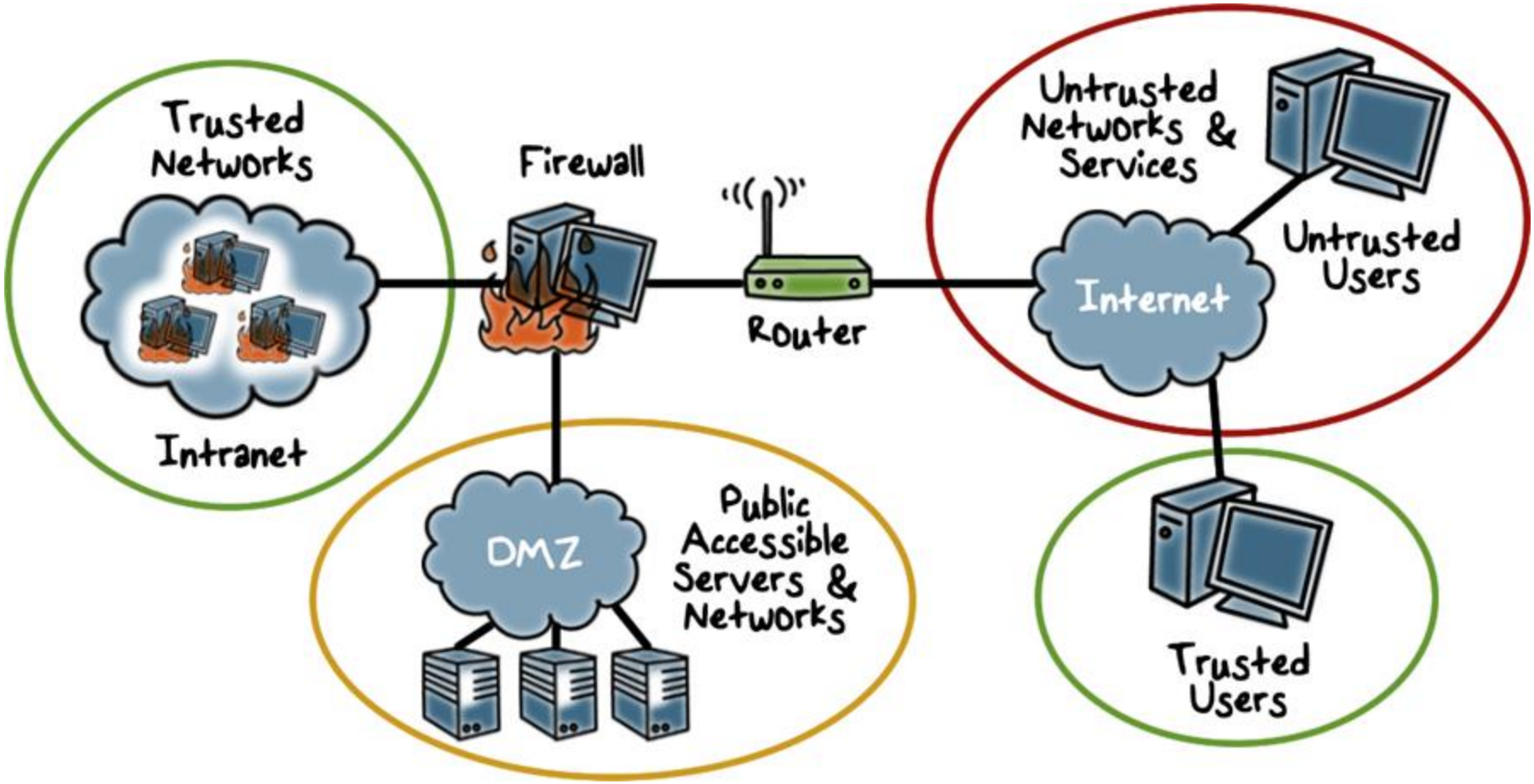
Slides based on

- Various sources

Security System Architecture

- Networked systems
  - Use of firewalls: Organization wide and system level
  - Address translation
  - Isolation of systems
- Single computing System: OS
  - Multiple levels of priviledges
  - Isolation of
    - processes,
    - cgroups,
    - virtual machines

Firewalls



DMZ: “Demilitarized zone”, distributed firewalls, From Georgia Tech
Note multiple levels of trust.

Firewalls

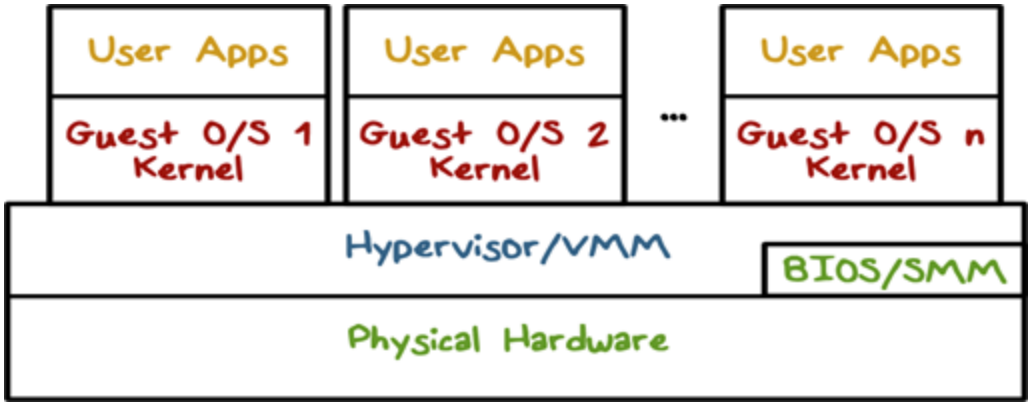
- A firewall checks traffic (packets or sessions) passing through it
- Can be programmed to check address ranges (IP addresses, ports), protocols, applications and content types.
- Can provide address translation, encryption

OS - trusted computing base

- The operating system serves as as trusted computing base (TCB) that controls access to protected resources.
  - Must establish the source of a request for a resource (authentication is how we do it)
  - Authorization or access control
  - Mechanisms that allow various policies to be supported
- How
  - Hardware support for memory protection
  - Processor execution modes (system and user modes)
  - Privileged instructions - can only be executed in system mode
  - System calls - transfer control between user and system code

Isolation in a system

- OS isolates address spaces of different processes using address translation. Also data vs code isolation.
  - Page tables governed by OS.
- In virtualization, hypervisor isolates virtual machines.
- Containers (Docker): Linux cgroups isolate process groups.



Assets, Risk, Threat, Vulnerability

System Resource (Asset): what needs protection by the defenders.

Risk: A measure of the adverse impacts and the likelihood of occurrence.

Threat: potential attempts by an adversary.

Vulnerability: Weakness in an information system that may be exploited.

Note of caution: In pre-cyber-security days, classical risk literature used the term vulnerability with a different meaning.

RFC 2828, Internet Security Glossary

Assets and threats

	Availability	Confidentiality	Integrity
Hardware	Equipment is stolen or disabled, thus denying service.	An unencrypted CD-ROM or DVD is stolen.	
Software	Programs are deleted, denying access to users.	An unauthorized copy of software is made.	A working program is modified, either to cause it to fail during execution or to cause it to do some unintended task.
Data	Files are deleted, denying access to users.	An unauthorized read of data is performed. An analysis of statistical data reveals underlying data.	Existing files are modified or new files are fabricated.
Communication Lines and Networks	Messages are destroyed or deleted. Communication lines or networks are rendered unavailable.	Messages are read. The traffic pattern of messages is observed.	Messages are modified, delayed, reordered, or duplicated. False messages are fabricated.

Question: where does ransomwere fit? Viruses?

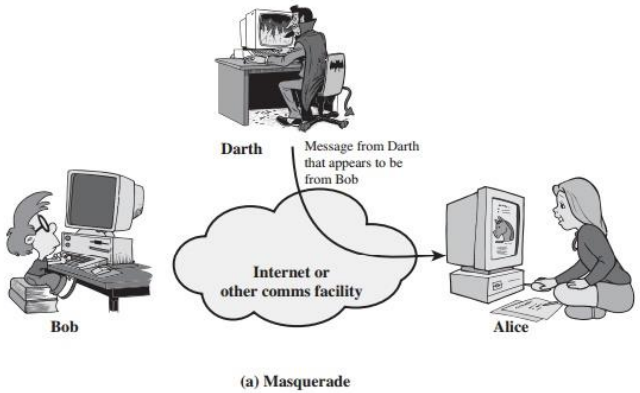
Computer security : Principles and Practice, William Stallings, Lawrie Brown

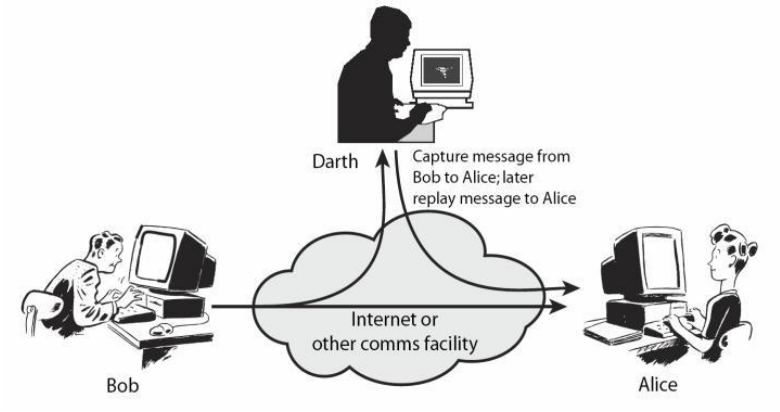
Attacks

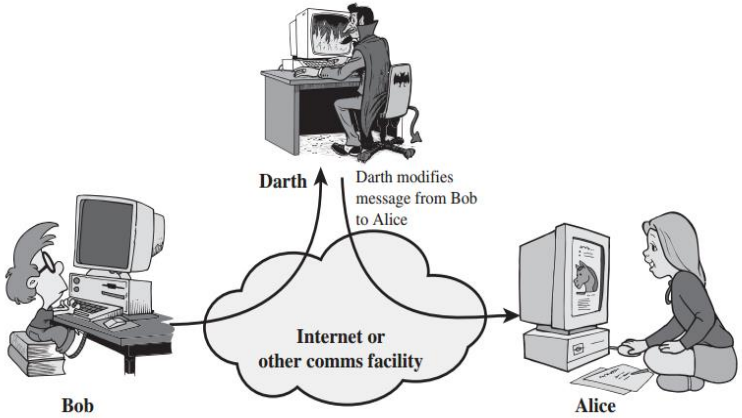
Passive Attack	Active Attack

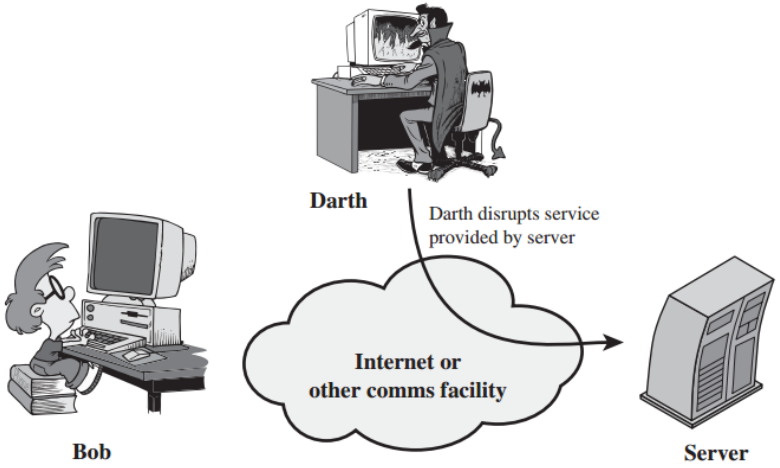
• Attempts to learn or make use of information from the system but does not affect system resources
• Eavesdropping on, or monitoring of, transmissions to obtain information that is being transmitted
• Two types:
  • Release of message contents
  • Traffic analysis
	• Attempts to alter system resources or affect their operation
	• Involve some modification of the data stream or the creation of a false stream
	• Four categories:
	  • Replay
	  • Masquerade
	  • Modification of messages
	  • Denial of service

Alice and Bob Diagrams





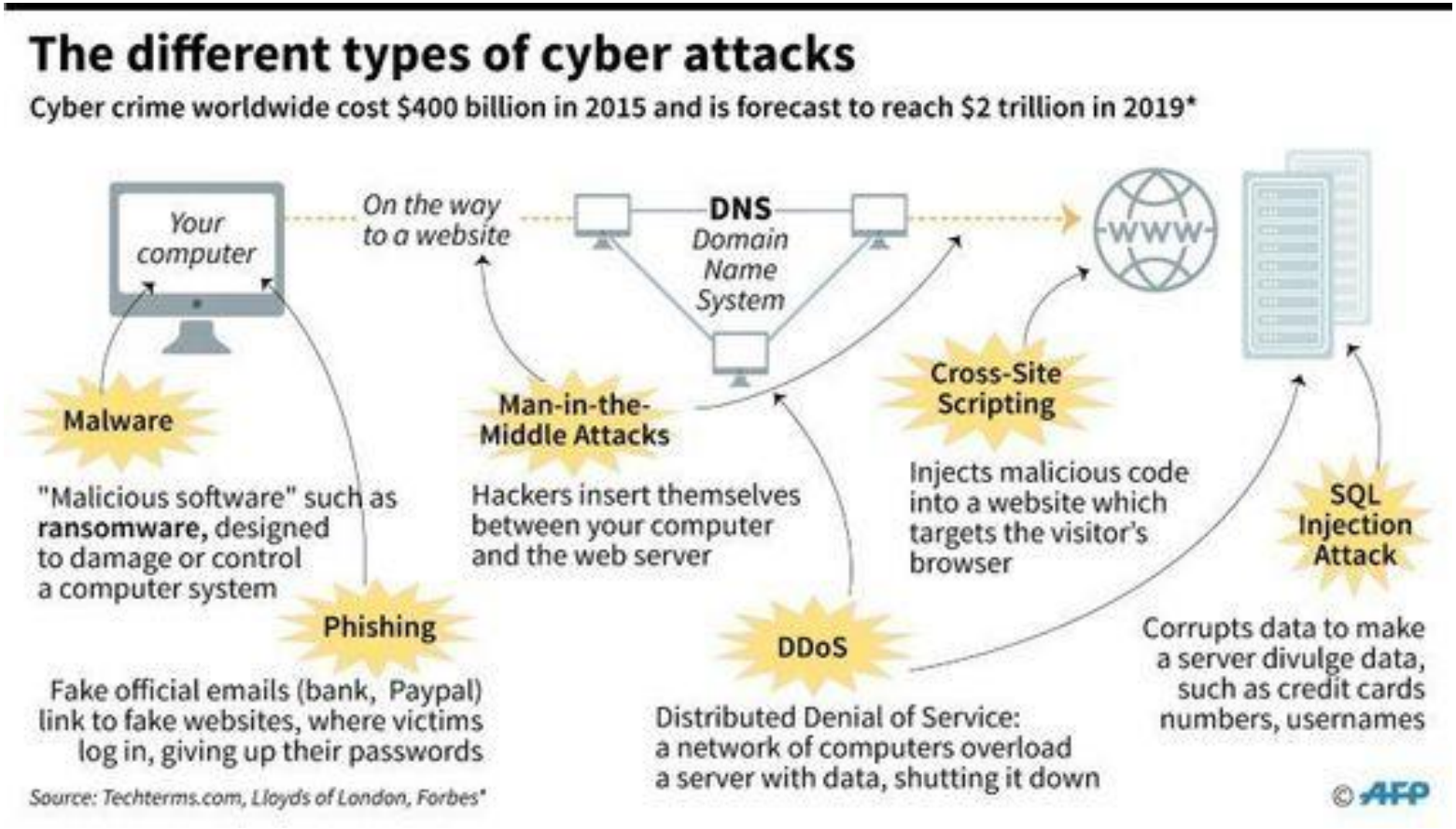




History: Rivest, Shamir, and Adleman's 1978 article "A method for obtaining digital signatures and public-key cryptosystems". a. masquerade, b. replay, c. modification of messages, d. denial-of-service

Colorado State University

Cyber attack types



Attack Surfaces

Surfaces: where the “holes” might be.

Network attack surface: vulnerabilities over an enterprise network, wide-area network, or the Internet.

- Including network protocol vulnerabilities, such as those used for a denial-of-service attack, disruption of communications links, and various forms of intruder attacks.

Software attack surface: vulnerabilities in application, utility, or operating system code.

- Web server, browser, Operating System.

Human attack surface: vulnerabilities in the personnel behavior

- social engineering, human error, and trusted insiders

Malware

- Malware ("malicious software"):
  - a catch-all term for any type of malicious software,
  - regardless of how it works, its intent, or how it's distributed.
- Virus
  - a specific type of malware that self-replicates by inserting its code into other programs. Types:
  - The **file infector** can burrow into executable files and spread through a network. A file infector can overwrite a computer's operating system or even reformat its drive.
  - The **macro virus** takes advantage of programs that support macros. Macro viruses usually arrive as Word or Excel documents attached to a spam email, or as a zipped attachment.
  - **Polymorphic viruses** modify their own code. The virus replicates and encrypts itself, changing its code just enough to evade detection by antivirus programs.

https://www.mcafee.com/enterprise/en-in/security-awareness/ransomware/malware-vs-viruses.html

Malware: Functional types

- **Worm:** a standalone program that can self-replicate and spread over a network. Unlike a virus, a worm spreads by exploiting a vulnerability in the infected system or through email as an attachment masquerading as a legitimate file.
- **Ransomware**: demands that users pay a ransom—usually in bitcoin or other cryptocurrency—to regain access to their computer.
- **Scareware**: attempts to frighten the victim into buying unnecessary software or providing their financial data.
- **Adware and spyware**: Adware pushes unwanted advertisements at users and spyware secretly collects information about the user. Spyware may record the websites the user visits, information about the user's computer system and vulnerabilities for a future attack, or the user's keystrokes.
  - Spyware that records keystrokes is called a keylogger.
- **Fileless malware**: Unlike traditional malware, fileless malware does not download code onto a computer, so there is no malware signature for a virus scanner to detect. Instead, fileless malware operates in the computer's memory and may evade detection by hiding in a trusted utility, productivity tool, or security application.

https://www.mcafee.com/enterprise/en-in/security-awareness/ransomware/malware-vs-viruses.html

CS370 Operating Systems

Colorado State University
Yashwant K Malaiya
Fall 2022

Access control

Slides based on

- Various sources

Access Control

Definition according to RFC 4949:

> "a process by which use of system resources is regulated according to a security policy and is permitted only by **authorized entities** (*users, programs, processes, or other systems*) according to that policy"

RFC 4949 defines security as

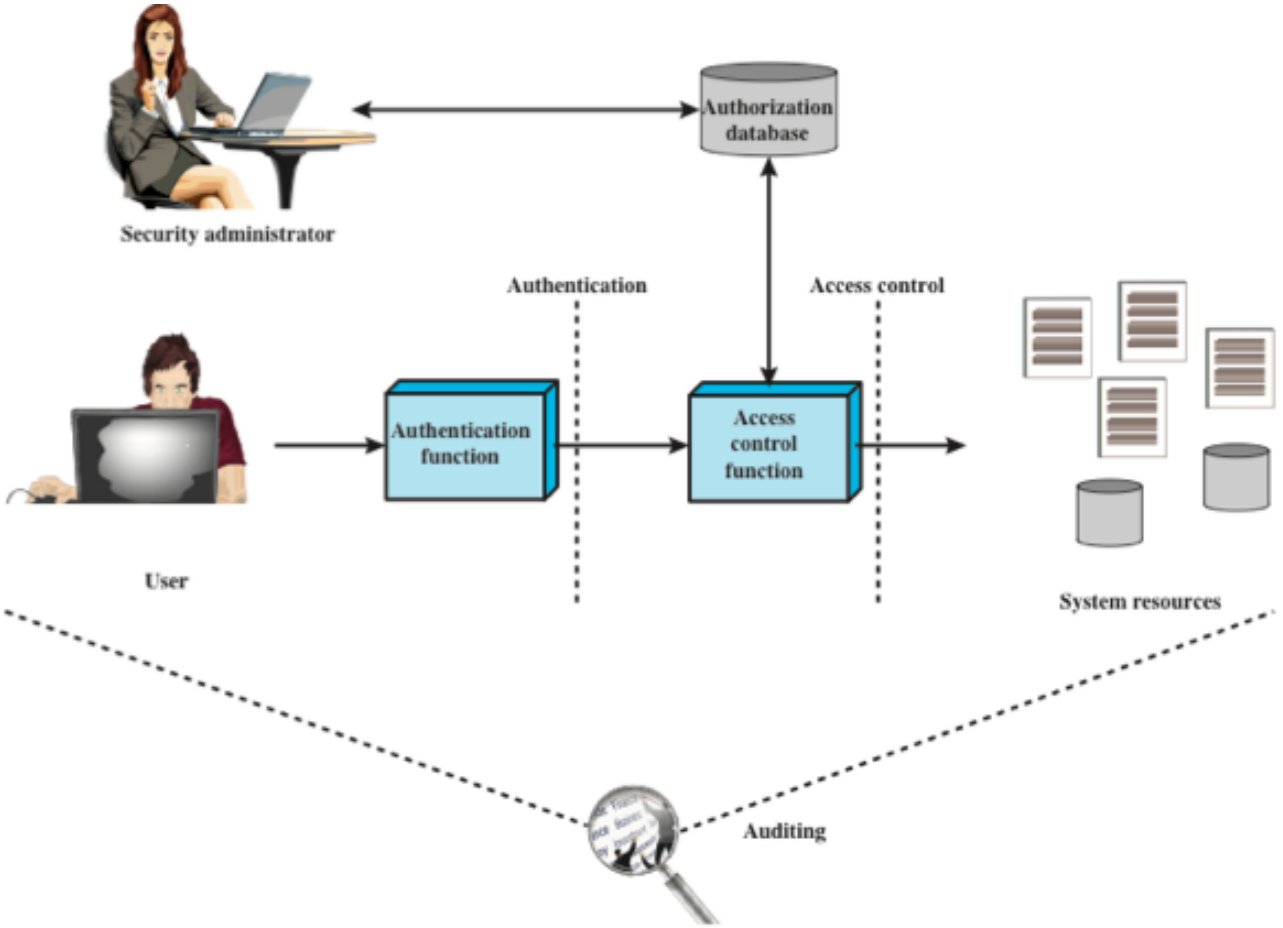
> "measures that implement and assure security services in a computer system, particularly those that assure access control service"

Thus all of computer security is concerned with access control.

Enforced by the Trusted Computing Base (OS) which performs

- authentication
- authorization

Access Control as a Security Function



Principles of Access Control

- Guiding principle – **principle of least privilege**
  - Programs, users and systems should be given just enough **privileges** to perform their tasks
- **Compartmentalization** a derivative concept regarding access to data
    - Process of protecting each individual system component through the use of specific permissions and access restrictions
- Can be
  - static (during life of system, during life of process)
  - Or dynamic (changed by process as needed) – access **domain switching**, **privilege escalation**

Principles of Access Control

- Rough-grained or Fine-grained management
  - Rough-grained privilege management easier, simpler, but least privilege now done in large chunks
    - For example, traditional Unix processes either have abilities of the associated user, or of root
  - Fine-grained management more complex, more overhead, but more protective
    - File ACL lists, RBAC
- Domain can be user, process, procedure
- **Audit trail** – recording all protection-orientated activities, important to understanding what happened, why, and catching things that shouldn't
- No single principle is a panacea for security vulnerabilities – need **defense in depth**

Subjects, Objects, and Access Rights

- A **subject** is an entity capable of accessing objects.
  - represented by a process. A user/application gains access to an object by means of a process that represents that user/application. The process takes on the attributes of the user, such as access rights.
  - Held accountable for the actions.
  - Classes: Owner (u in linux), Group (g), World (o), people with specific roles
- An **object** is a resource to which access is controlled.
  - an entity used to contain and/or receive information.
  - Examples: pages/segments, files/directories/programs, ports, devices etc.

Subjects, Objects, and Access Rights

An access right describes the way in which a subject may access an object.

- **Read**: User may view information in a system resource (e.g., a file, selected records in a file, selected fields within a record, or some combination). Read access includes the ability to copy or print.
  - Directory: ability to list the directory.
- **Write**: User may add, modify, or delete data in system resource (e.g., files, records, programs). Write access includes read access.
  - Directory: create new files
- **Execute:** User may execute specified programs.
  - Directory: enter it to access the files within it.
- Delete: User may delete certain system resources, such as files or records.
- Create: User may create new files, records, or fields.
- Search: User may list the files in a directory or otherwise search the directory.

Access Control Schemes

Discretionary Access Control (DAC): Scheme in which an entity may be granted access rights that permit the **owner** entity, by its own violation, to enable another entity to access some resources.

- Provided using an access matrix

Mandatory Access Control: **Centralized authority** sets security policy for all resources

- Example: SELinux

Example: Access Control Matrix

		OBJECTS			
		File 1	File 2	File 3	File 4
SUBJECTS	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

(a) Access matrix

Access Control List (ACL): Every object has an ACL that identifies what operations subjects can perform. Each access to object is checked against object's ACL.

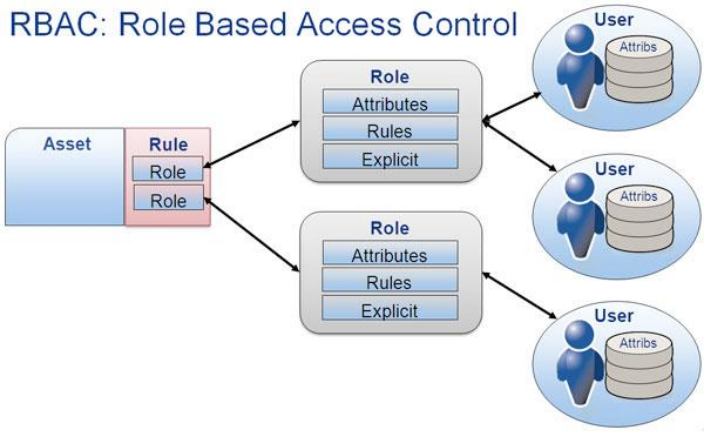
May be kept in a relational database. Access recorded in file metadata (inode).

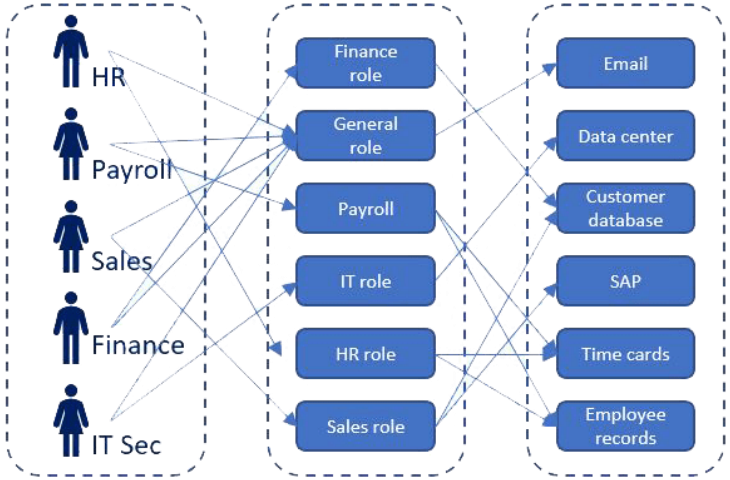
Unix Access Control

- Subjects (Who?)
  - Users
- Objects (What?)
  - Files, directories
  - Files: sockets, pipes, hardware devices, kernel objects, process data
- Access Operations
  - Read, Write, Execute
  - Set by root or owner of the object
- Linux is an example of *discretionary access control*.
  - Resource owners can set the security policy for objects they own
- Superuser (root) allowed to do anything.
  - System administrators assume superuser role to perform privileged actions – Good practice to assume superuser role only when necessary

Role Based Access Control (RBAC)

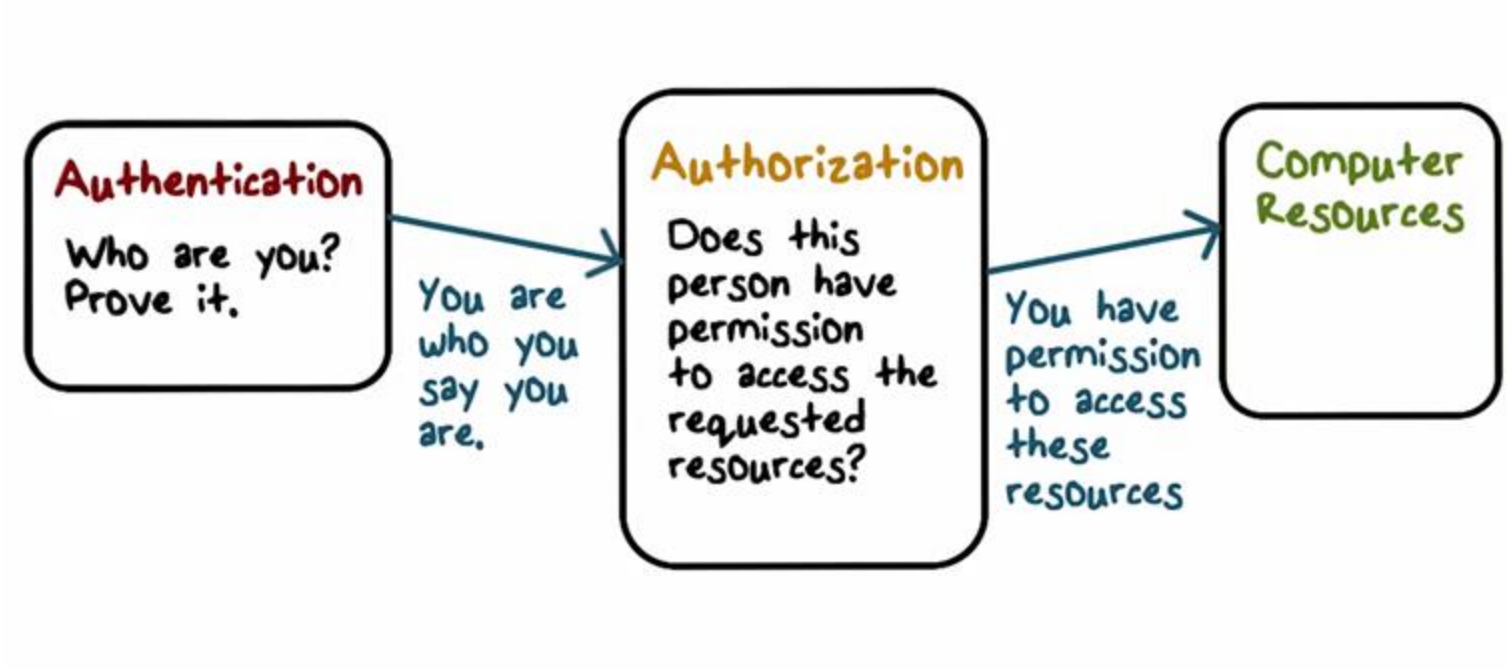
- **Role-based access control (RBAC):** based on the roles that users have within the system and on rules specifying the accesses are allowed to users in given roles.
- Widely used commercially in larger organizations.





Colorado State University

Authentication



Georgia Tech

Authentication

- OS (Trusted Computing Base) needs to know who makes a request for a protected resource
- A process that makes the request does it on behalf of a certain user
- Authentication handles the question: on whose behalf the requesting process runs?
- Involves
  - claims about an identity and
  - verification of the claimed identity
- Goals
  - No false negatives
  - No false positives (major consideration)

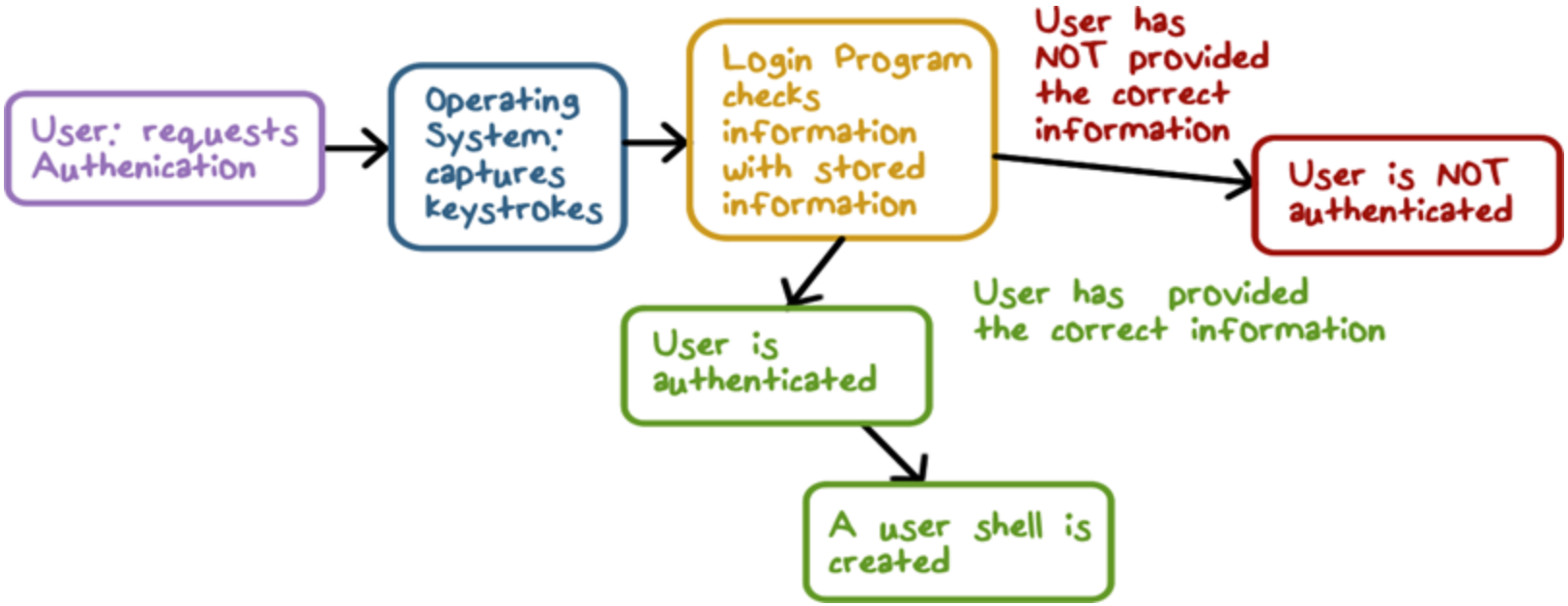
Authentication Methods

Three existing and two new.

- Something a user knows
  - Password, answers to questions
- Something a user has
  - Ex. Id card, Phone
- Something a user is
  - Biometric (face, iris, fingerprint)
- Somewhere you are geographically
- Something you do based on recognizable patterns of behavior

- Can be multifactor to reduce false positives
- After-access confirmation

Implementation: Password based Authentication



The system must provide a trusted path from keyboard to the OS.

Georgia Tech

Password authentication

Possible approaches

1. Store a list of passwords, one for each user in the system file, readable only by the root/admin account
   - Why the admin need to know the passwords?
   - If security is breached, the passwords are available to an attacker. No longer used.
2. Do not store passwords, but store something that is derived from them
   - Use a hash function and store the result
     - Dictionaries of hashed passwords exist. “Salt” is added to make cracking harder.

Security Challenges

- Password guessing
  - List of bad passwords. 123456, password, ..
- Brute force guessing
  - more later
- Good passwords are the ones harder to remember
- May be stolen using
  - keyloggers,
  - compromised websites where same password was used
  - eavesdropping *(Alice, Bob and Eve?)*
- Solution: multi-factor authentication

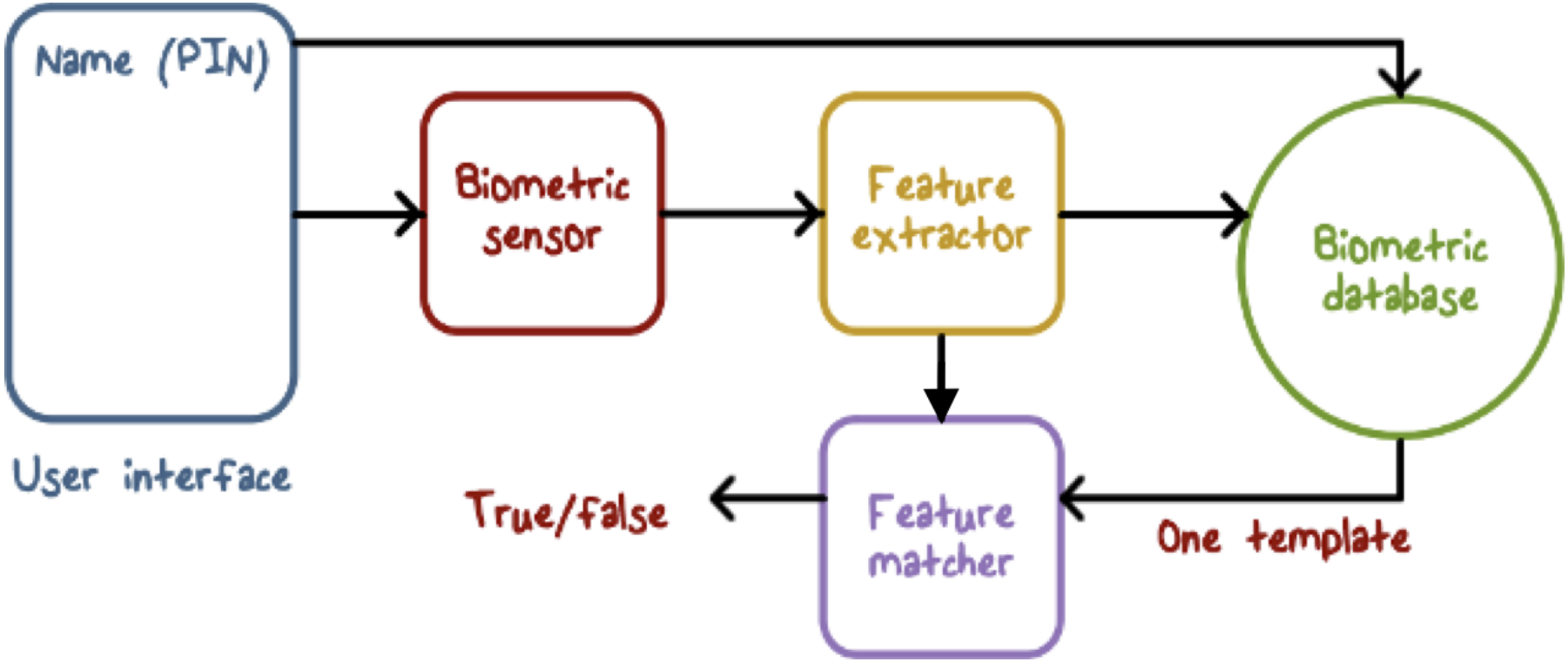
Biometric Authentication

- Fingerprints (finger swipes)
- Keystroke dynamics
- Voice
- Retina scans

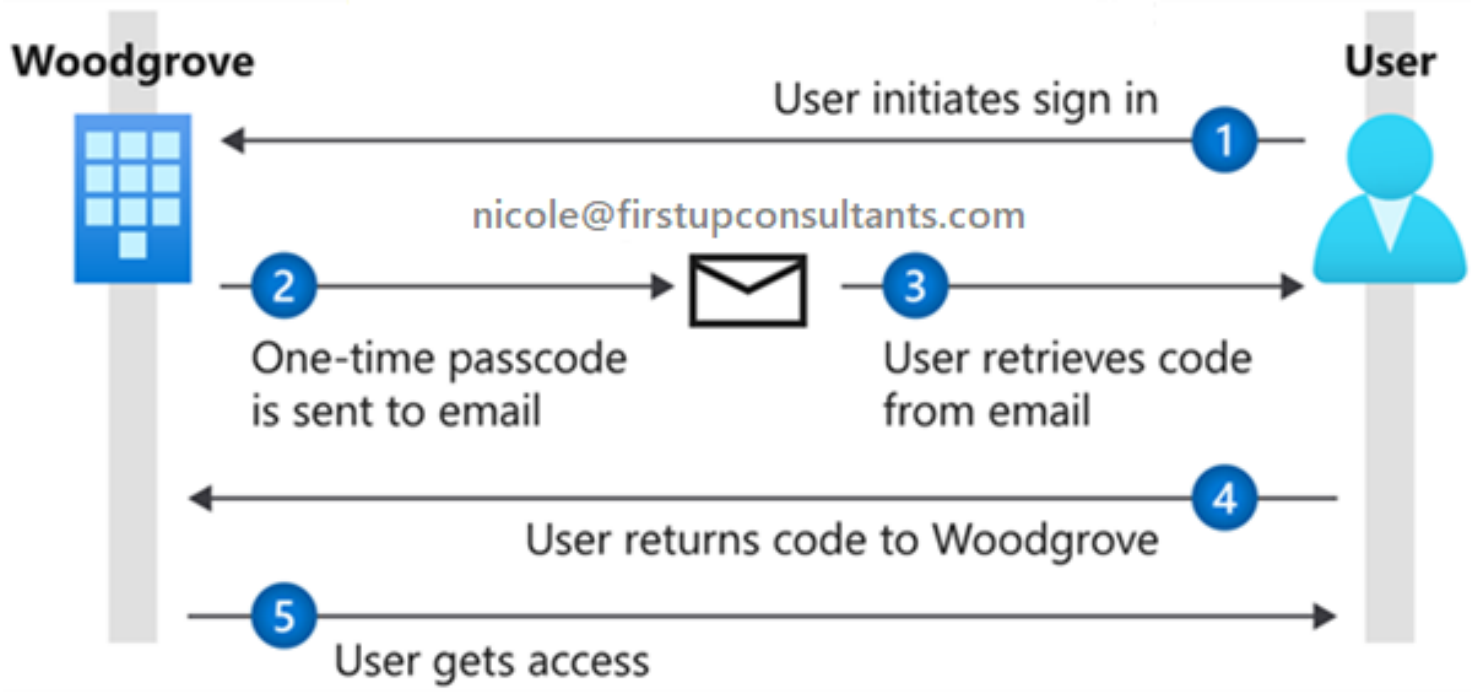
Issues

- Feature value distribution or a range
- False positives and negatives

Implementing Biometric Authentication



Smartphone OTP Authentication



OTP (one-time passcode authentication) texted or email to unique phone number/email address.