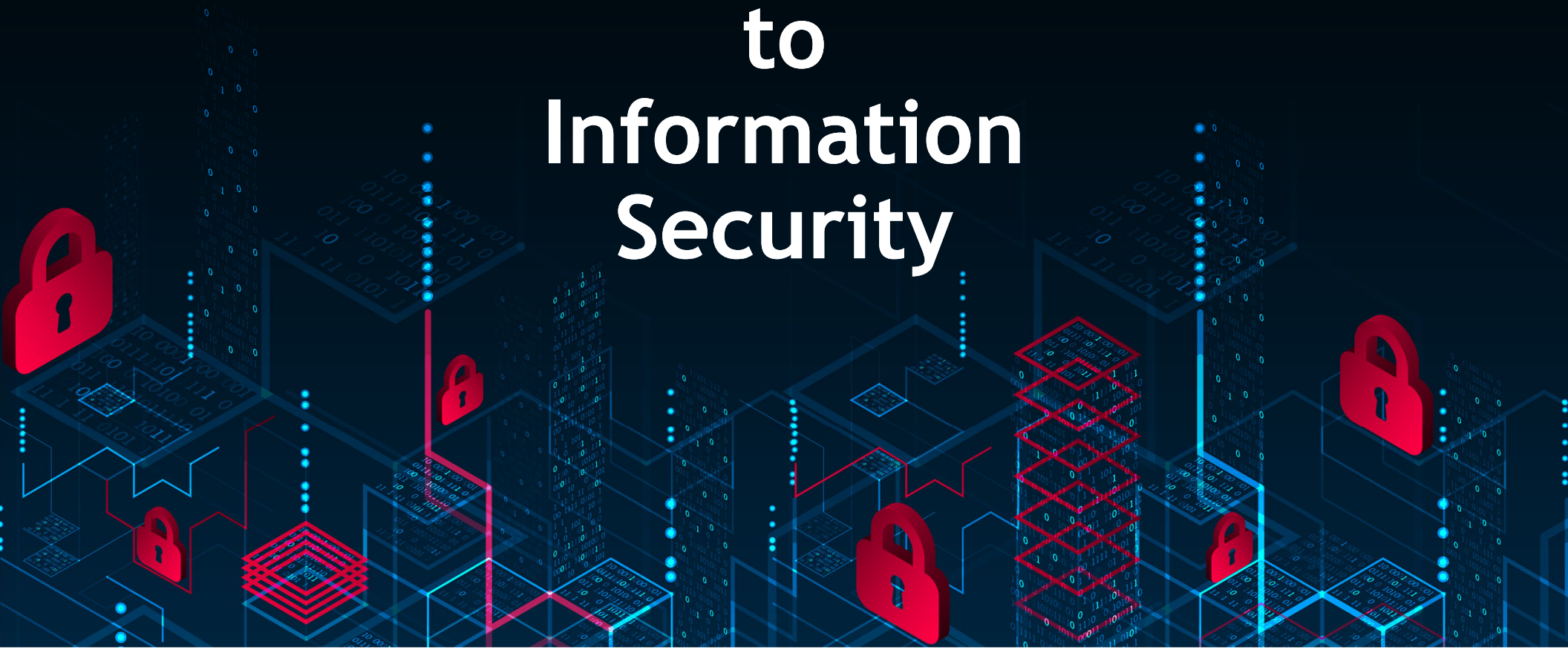
CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

Introduction to Information Security



Module Objectives:

- Introduction to Information Security
  - By the end of this module you will be able to:
    - Differentiate between Confidentiality, Integrity, and Availability
    - Understand technical areas that must underpin any effective security strategy
    - Differentiate between threats, attacks and assets

Introduction to Information Security

CIA Triad

Protecting Assets

Security Concepts

Attacks
(Active & Passive)

Counter Measures

Security Aspects

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

Introduction to Information Security

Module:

- Introduction to Information Security

  - By the end of this module you will be able to:

    - Differentiate between Confidentiality, Integrity, and Availability
    - Understand technical areas that must underpin any effective security strategy
    - Differentiate between threats, attacks and assets

The NIST Internal/Interagency Report NISTIR 7298 defines the term *computer security* as follows:

> “Measures and controls that ensure confidentiality, integrity, and availability of information system assets including hardware, software, firmware, and information being processed, stored, and communicated.”

NIST: National Institute of Standards and Technology (www.nist.gov)

Key Security Objectives

- Confidentiality
  - Data confidentiality: assure confidential information not made available to unauthorized individuals
  - Privacy: assure individuals can control what information related to them is collected, stored, distributed
- Integrity
  - Data integrity: assure information and programs are changed only in a authorized manner
  - System integrity: assure system performs intended function
- Availability
  - Assure that systems work promptly and service is not denied to authorized users

CIA Triad

Extended CIA Triad

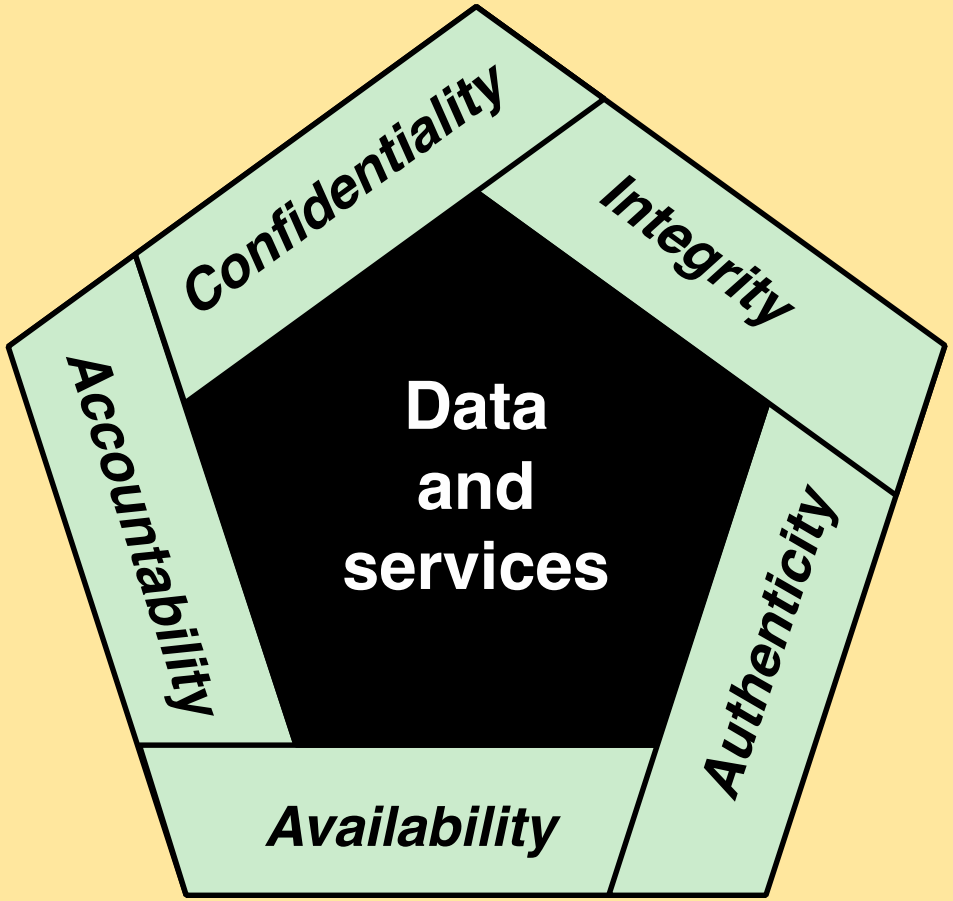


Figure 1.1 Essential Network and Computer Security Requirements

Extended CIA Triad

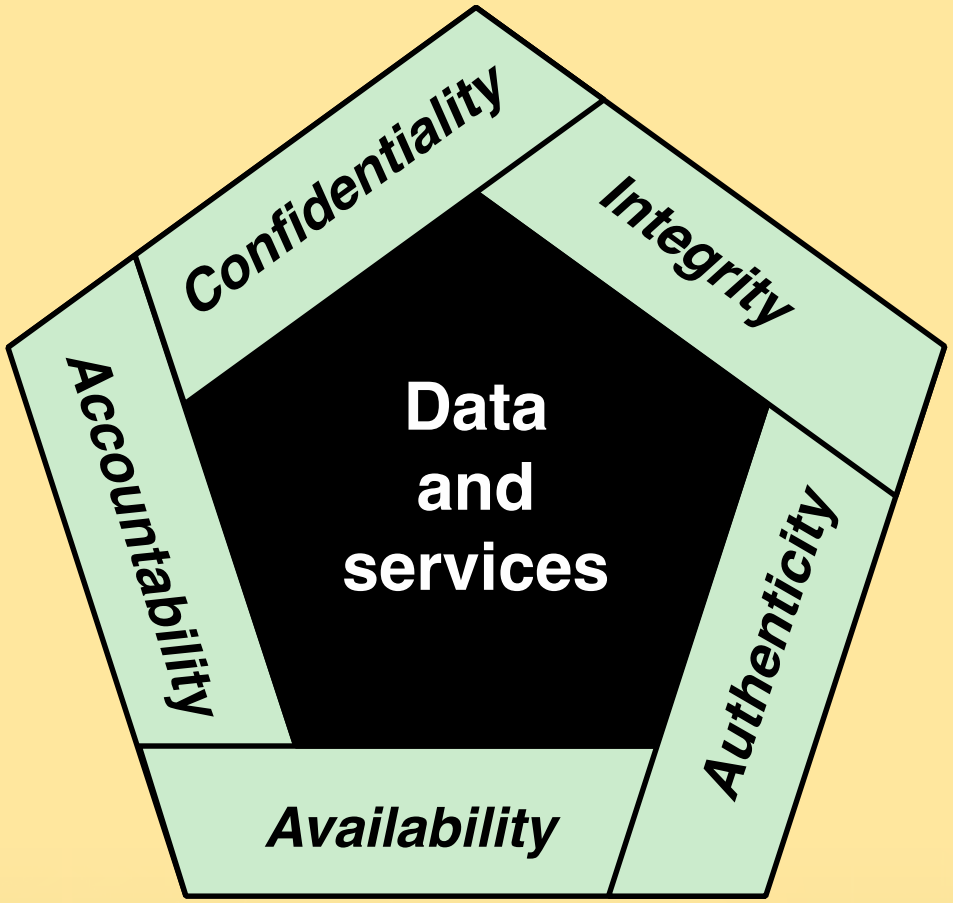


Figure 1.1 Essential Network and Computer Security Requirements

Extended CIA Triad

- Authenticity
  - Users and system inputs are genuine and can be verified and trusted
    - Data authentication
    - Source authentication

- Accountability
  - Actions of an entity can be traced uniquely to that entity
    - Supports: non-repudiation, deterrence, fault isolation, intrusion detection and prevention, after-action recovery

Key Security Concepts

- Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information

- Guarding against improper information modification or destruction, including ensuring information nonrepudiation and authenticity

- Ensuring timely and reliable access to and use of information

Levels of Impact

- We use three levels of impact on organizations or individuals should there be a breach of security (i.e., a loss of confidentiality, integrity, or availability).

- These levels are defined in FIPS 199:

  - Low
  - Moderate
  - High

FIPS: Federal Information Processing Standards (part of NIST)

Levels of Impact

Low

The loss could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals

Moderate

The loss could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals

High

The loss could be expected to have a severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals

Computer Security Challenges

1. Computer security is not as simple as it might first appear to the novice
2. In developing a particular security mechanism or algorithm, one must always consider potential attacks on those security features
3. Procedures used to provide particular services are often counterintuitive
4. Physical and logical placement of security mechanisms needs to be determined
5. Security mechanisms typically involve more than a particular algorithm or protocol and also require that participants be in possession of some secret information which raises questions about the creation, distribution, and protection of that secret information
6. Attackers only need to find a single weakness, while the designer must find and eliminate all weaknesses to achieve perfect security
7. Security is still too often an afterthought to be incorporated into a system after the design is complete, rather than being an integral part of the design process
8. Security requires regular and constant monitoring
9. There is a natural tendency on the part of users and system managers to perceive little benefit from security investment until a security failure occurs
10. Many users and even security administrators view strong security as an obstacle to efficient and user-friendly operation of an information system or use of information

Module:

- Introduction to Information Security

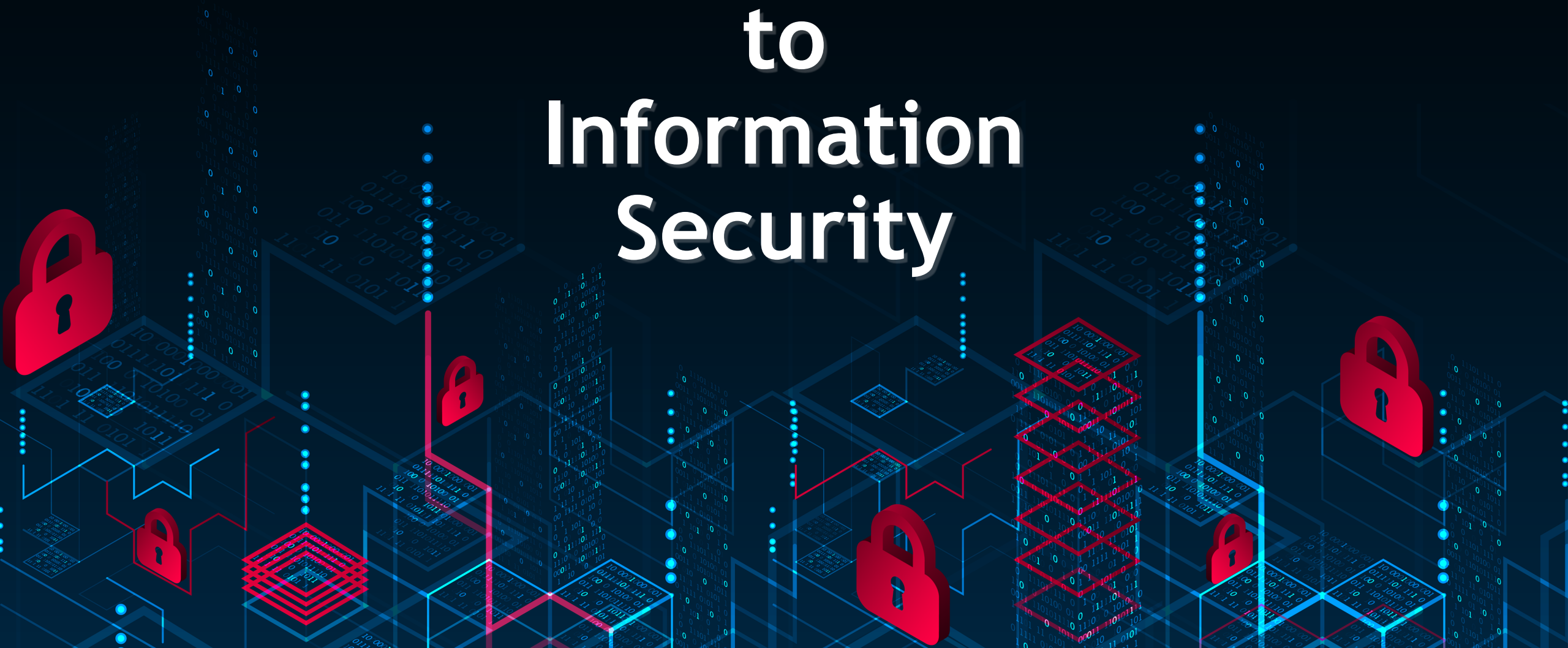
  - By the end of this module you will be able to:

    - Differentiate between Confidentiality, Integrity, and Availability
    - Understand technical areas that must underpin any effective security strategy
    - Differentiate between threats, attacks and assets

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

Introduction to Information Security



Module:

- Introduction to Information Security

  - By the end of this module you will be able to:

    - Differentiate between Confidentiality, Integrity, and Availability
    - Understand technical areas that must underpin any effective security strategy
    - Differentiate between threats, attacks and assets

Computer Security Concepts

- Assets
- Vulnerabilities
- Security Policies
- Threats
- Attacks
- Countermeasure

Assets of a Computer System
(things that we want to protect)

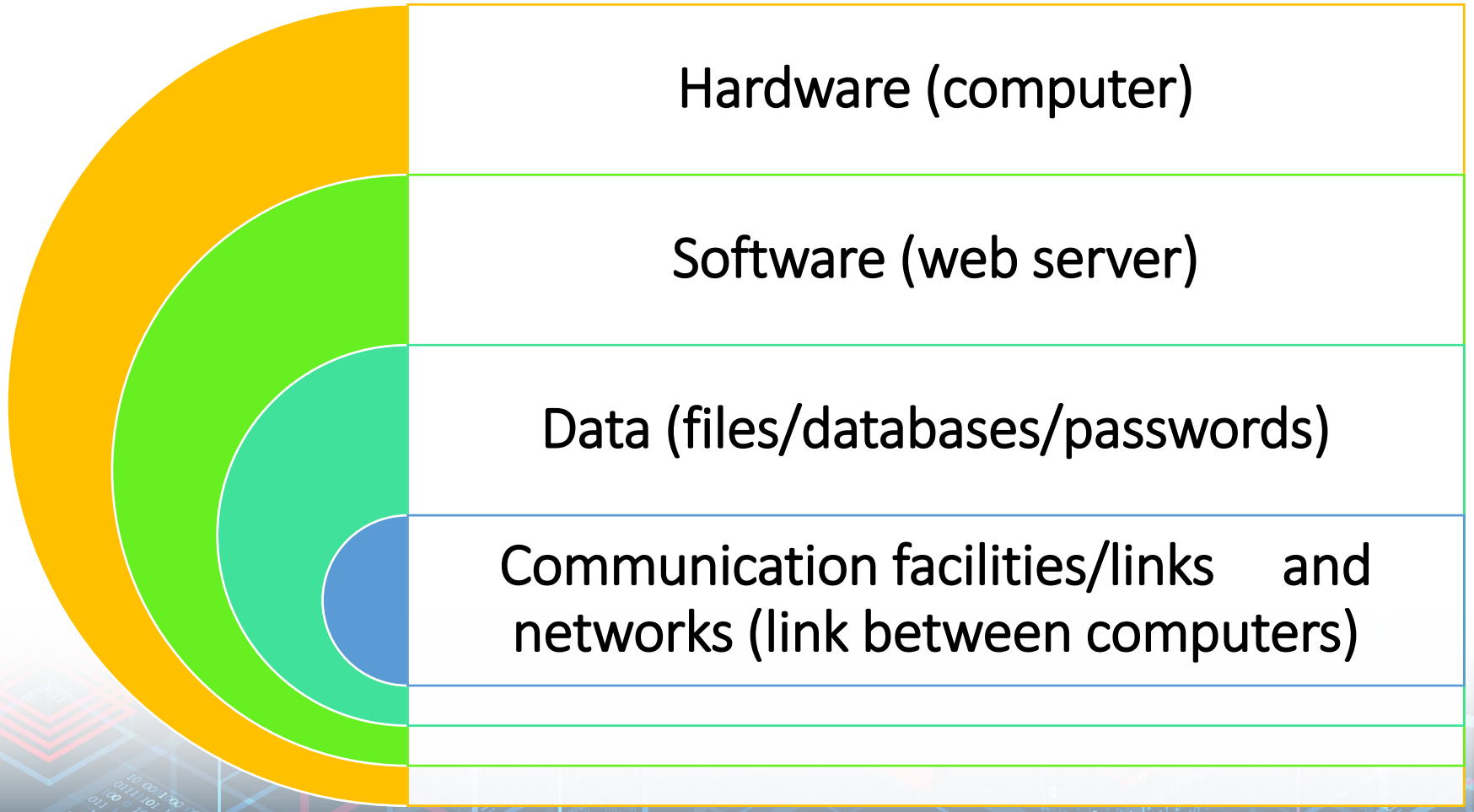


Table 1.3

Computer and Network Assets, with Examples of 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.

Vulnerabilities, Threats and Attacks

- Categories of vulnerabilities
  - Corrupted (loss of integrity – ex: asset doesn't do its function)
  - Leaky (loss of confidentiality – ex: leaks out information)
  - Unavailable or very slow (loss of availability - ex: users can't access the asset)

- Note that it is complex to write a software without a bug
- it is complex to build a hardware without flaws,
- and it is complex to keep track of data

There are often vulnerabilities… try to avoid them

Vulnerabilities, Threats and Attacks

- Security Policy
  - Set of rules and practices that specifies how a certain organization/company provides security services to protect assets
  - Example in the university there is a policy for who can access student's data (Confidentiality).
  - Can I access your grades in another course?
  - The organization must implement certain techniques to implement those policies

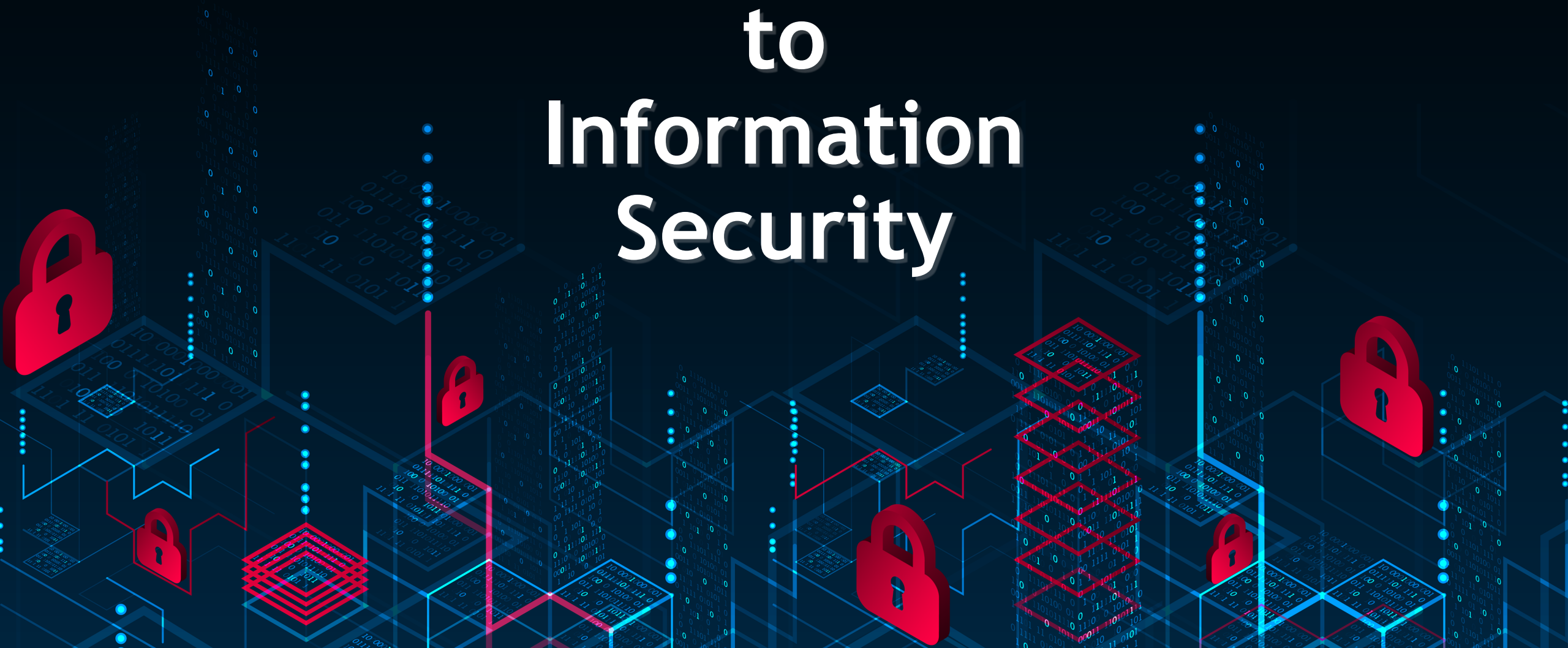
Computer Security Concepts

- Assets
- Vulnerabilities
- Security Policies
- Threats
- Attacks
- Countermeasure

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

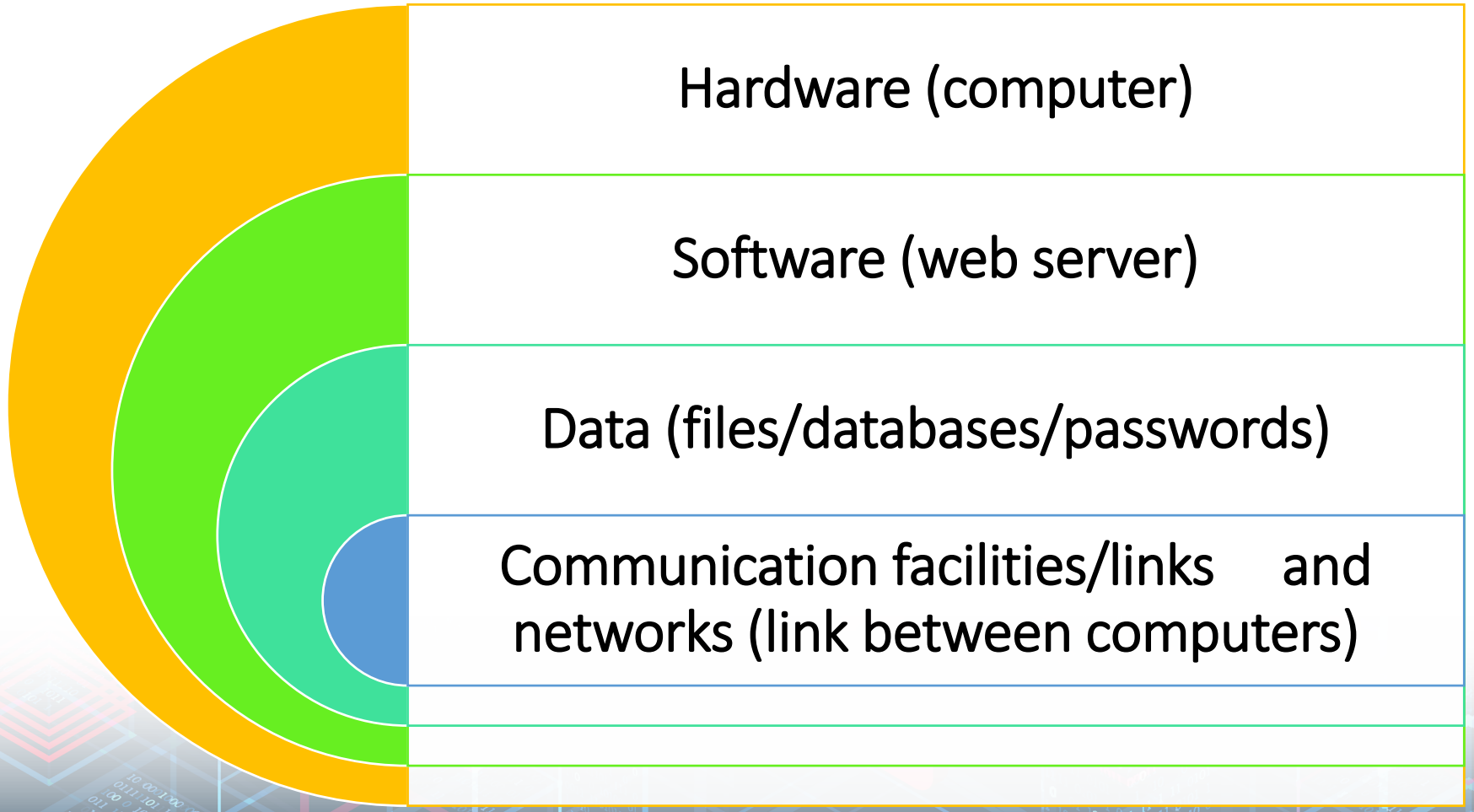
Introduction to Information Security



Computer Security Concepts

- Assets
- Security Policies
- Vulnerabilities
- Threats
- Attacks
- Countermeasure

Assets of a Computer System
(things that we want to protect)



Vulnerabilities, Threats and Attacks

- Categories of vulnerabilities
  - Corrupted (loss of integrity – ex: asset doesn't do its function)
  - Leaky (loss of confidentiality – ex: leaks out information)
  - Unavailable or very slow (loss of availability - ex: users can't access the asset)

- Note that it is complex to write a software without a bug
- it is complex to build a hardware without flaws,
- and it is complex to keep track of data

There are often vulnerabilities… try to avoid them

Vulnerabilities, Threats and Attacks

- Security Policy
  - Set of rules and practices that specifies how a certain organization/company provides security services to protect assets
  - Example in the university there is a policy for who can access student's data (Confidentiality).
  - Can I access your grades in another course?
  - The organization must implement certain techniques to implement those policies

Vulnerabilities, Threats and Attacks

- Threats
  - Potential violation of security policy by exploiting a vulnerability
  - Represent potential security harm to an asset
  - If we have a policy that a student can't access the grades of another student. Threat is if something allow a student to potential access another student's grades.

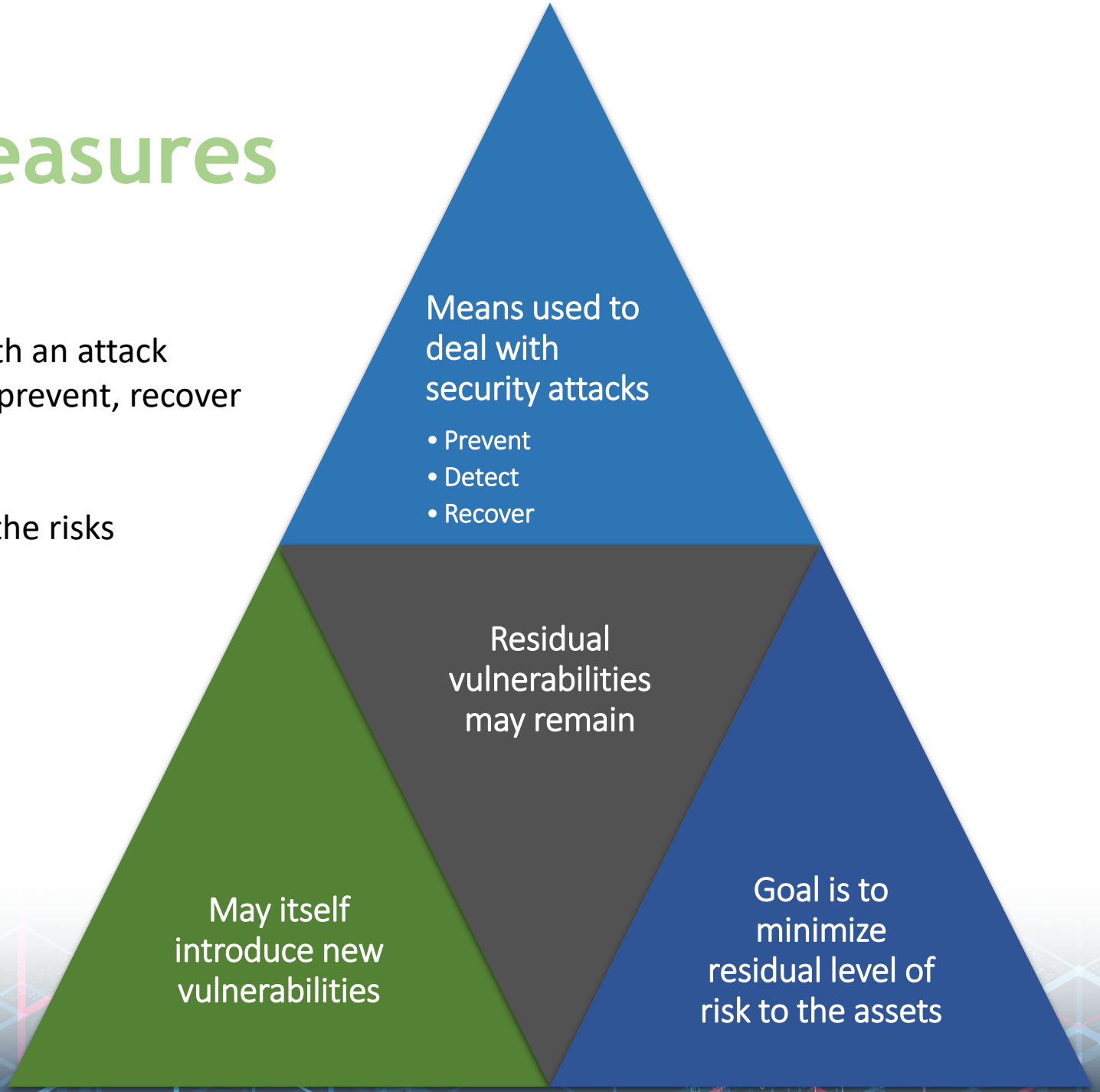
Vulnerabilities, Threats and Attacks

- Attacks
  - A threat that is carried out; a successful attack leads to violation of security policy
  - Passive – attempt to learn or make use of information from the system that does not affect system resources
  - Active – attempt to alter system resources or affect their operation
  - Insider – initiated by an entity inside the security parameter
  - Outsider – initiated from outside the perimeter

Countermeasures

A way to deal with an attack
Detect, respond, prevent, recover

Aim to minimize the risks



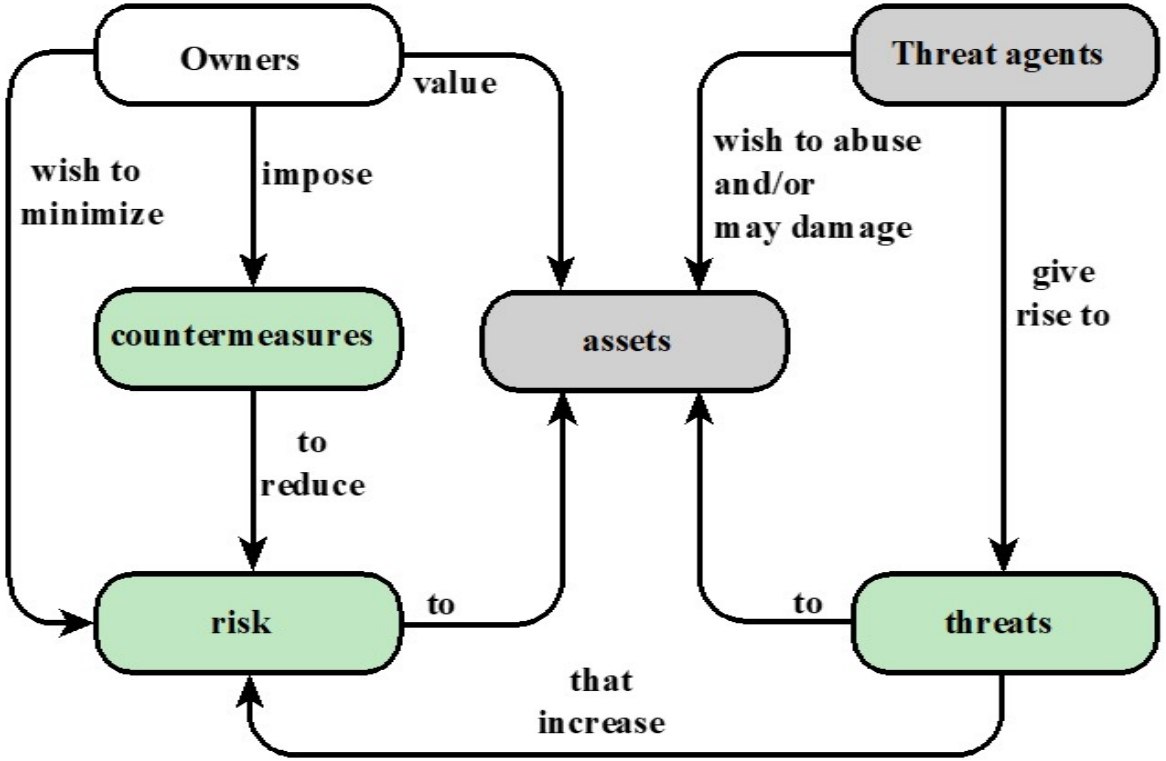


Figure 1.2 Security Concepts and Relationships

Threat Consequences and Attacks

- Threat Action: an attack
- Threat Agent: Entity that attacks, or is threat to system (hacker, attacker (don't have to be bad - law enforcement), malicious user)
- Threat Consequence: A security violation that results from threat action

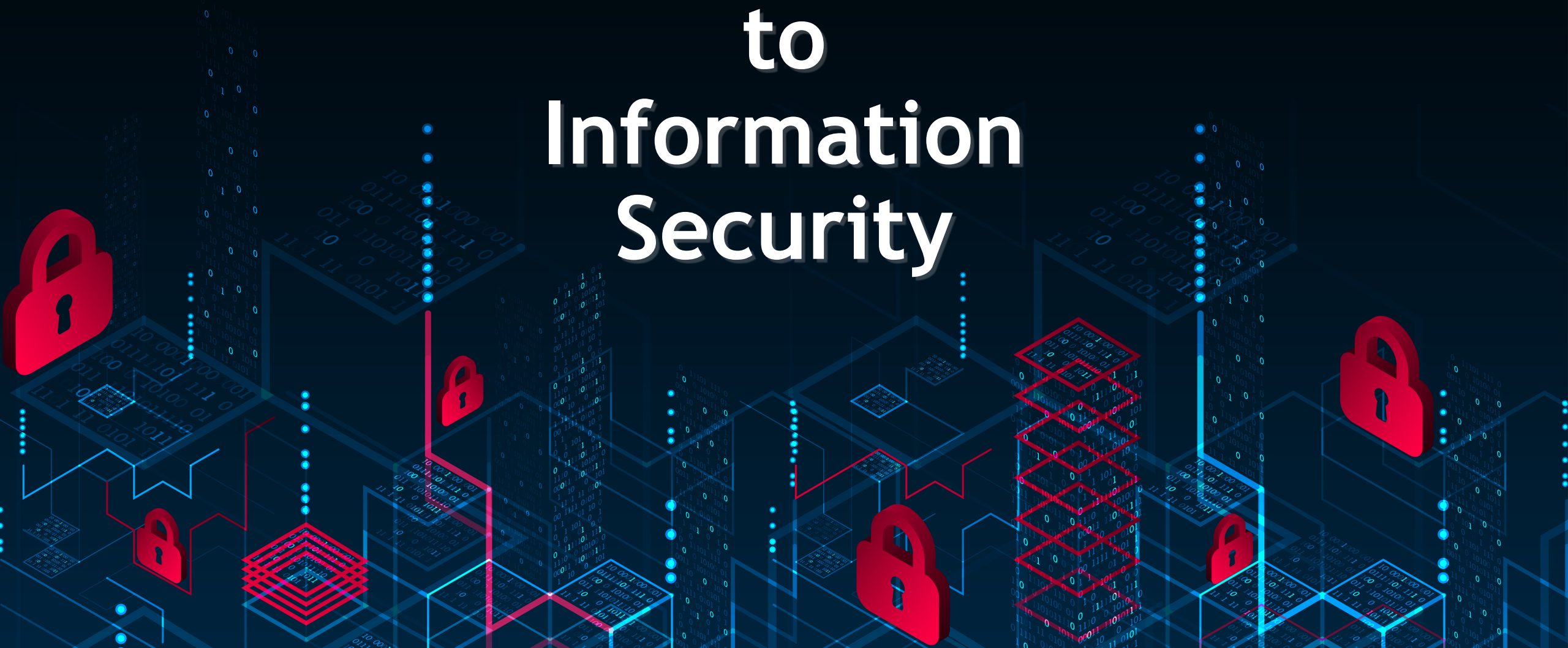
Computer Security Concepts

- Assets
- Security Policies
- Vulnerabilities
- Threats
- Attacks
- Countermeasure

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

Introduction to Information Security



Types of Attacks

Passive attacks

Active attacks

Passive and Active Attacks

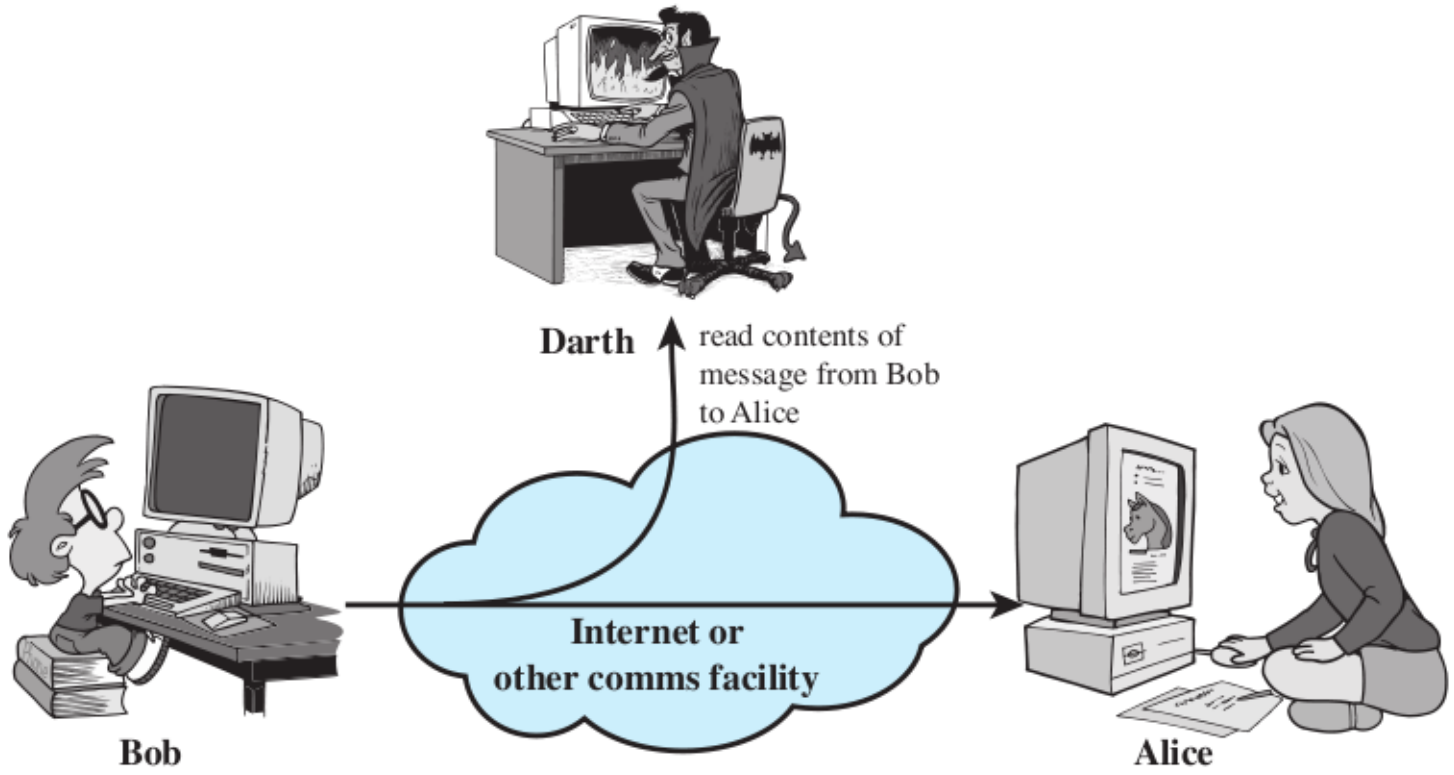
Passive Attack

- Attempts to learn or make use of information from the system but does not affect system resources
- Eavesdropping on, or monitoring of, transmissions
- Goal of attacker is to obtain information that is being transmitted
- Two types:
  - Release of message contents
  - Traffic analysis

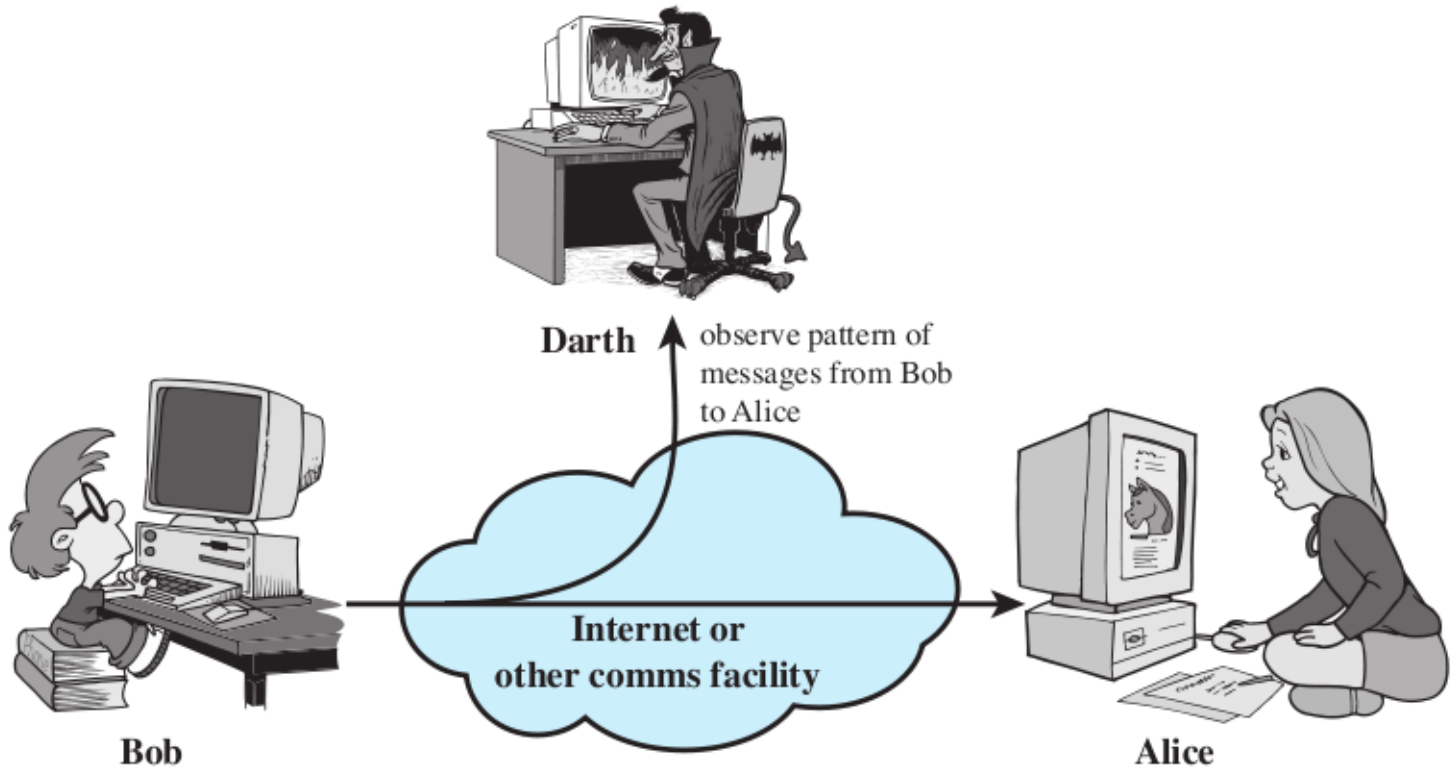
Active Attack

- 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

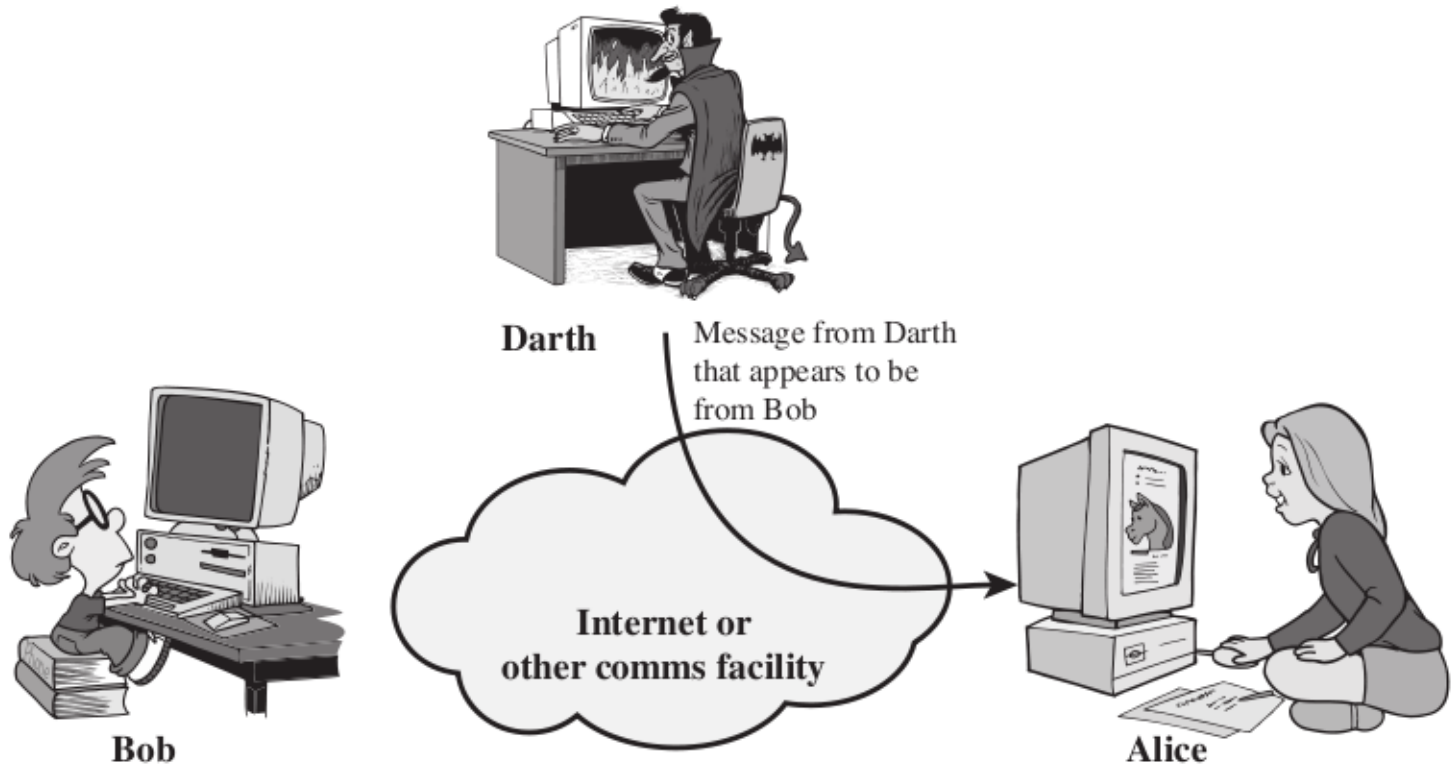
Release Message Contents



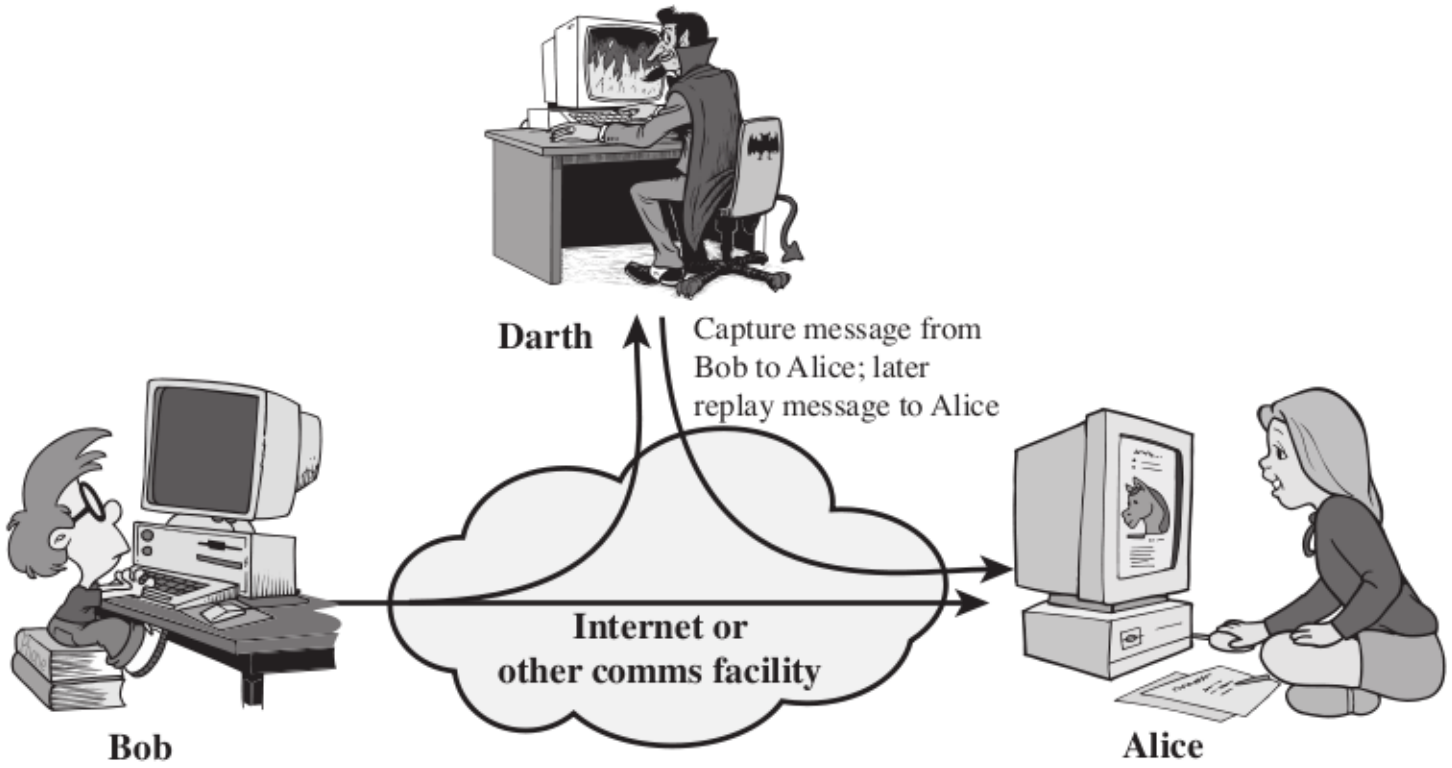
Traffic Analysis



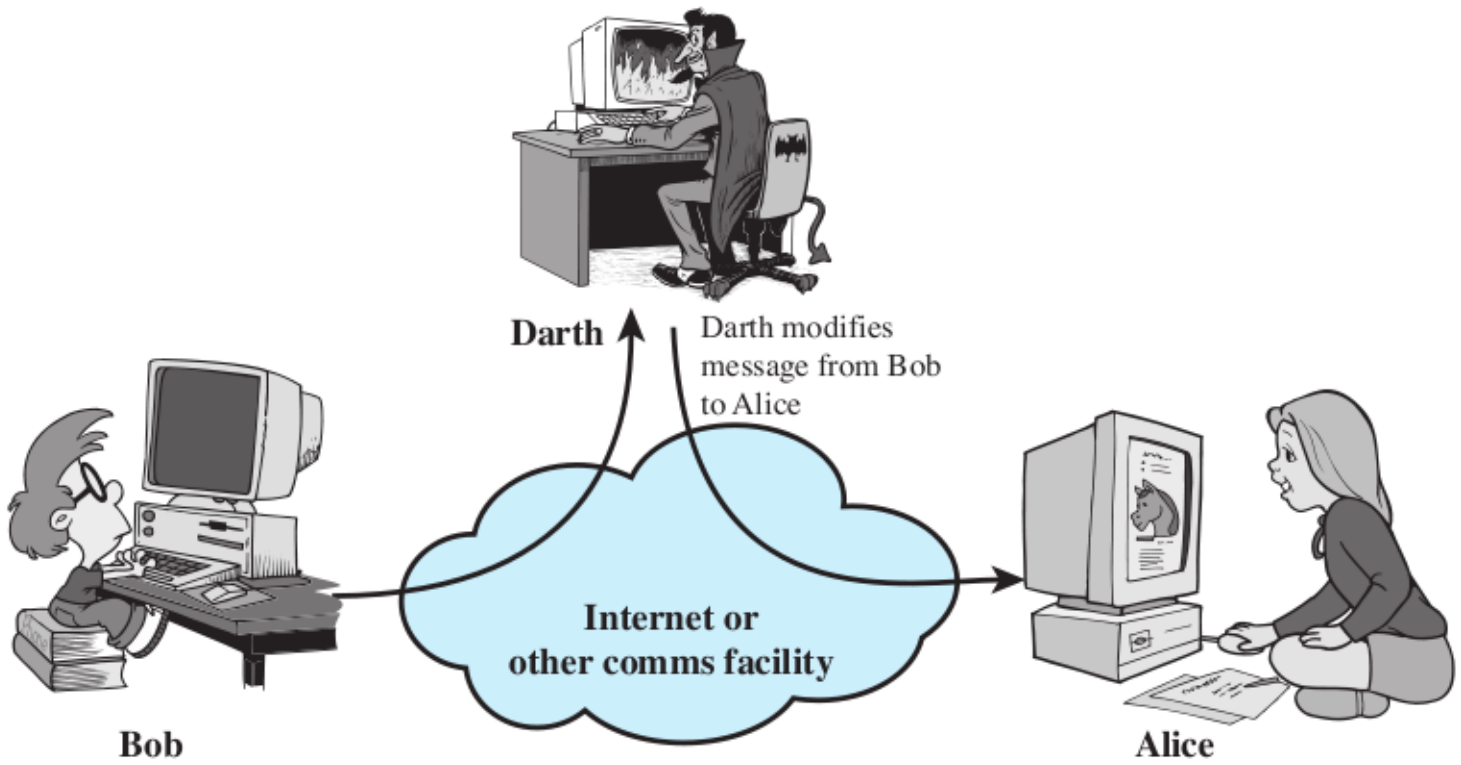
Masquerade Attack



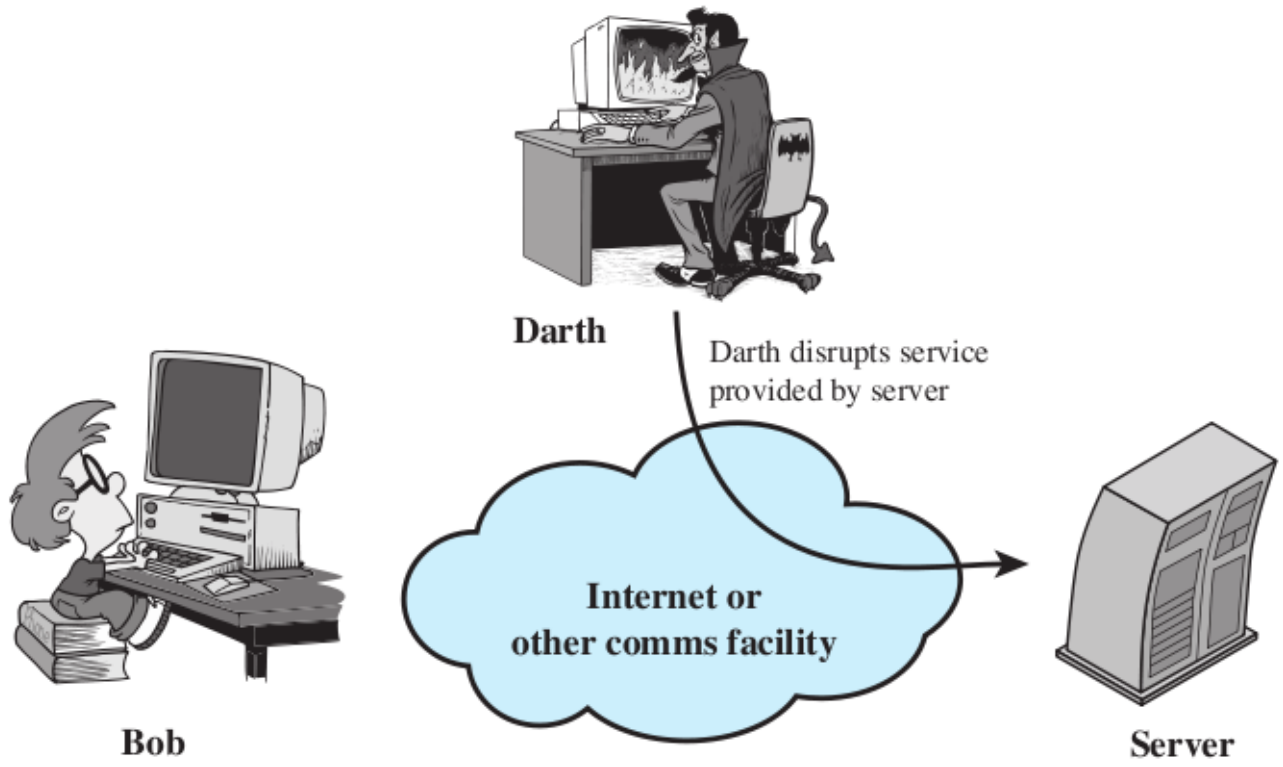
Replay Attack



Modification Attack



Denial of Service Attack



Types of Attacks

Passive attacks: are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the attacker is to obtain information that is being transmitted.
Two types of passive attacks are:

a) release of message contents
b) traffic analysis.

Active attacks: involve some modification of the data stream or the creation of a false stream.
Fours types of active attacks are:

a) Replay attack
b) Masquerade attack
c) Modification of messages attack
d) Denial of service.

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

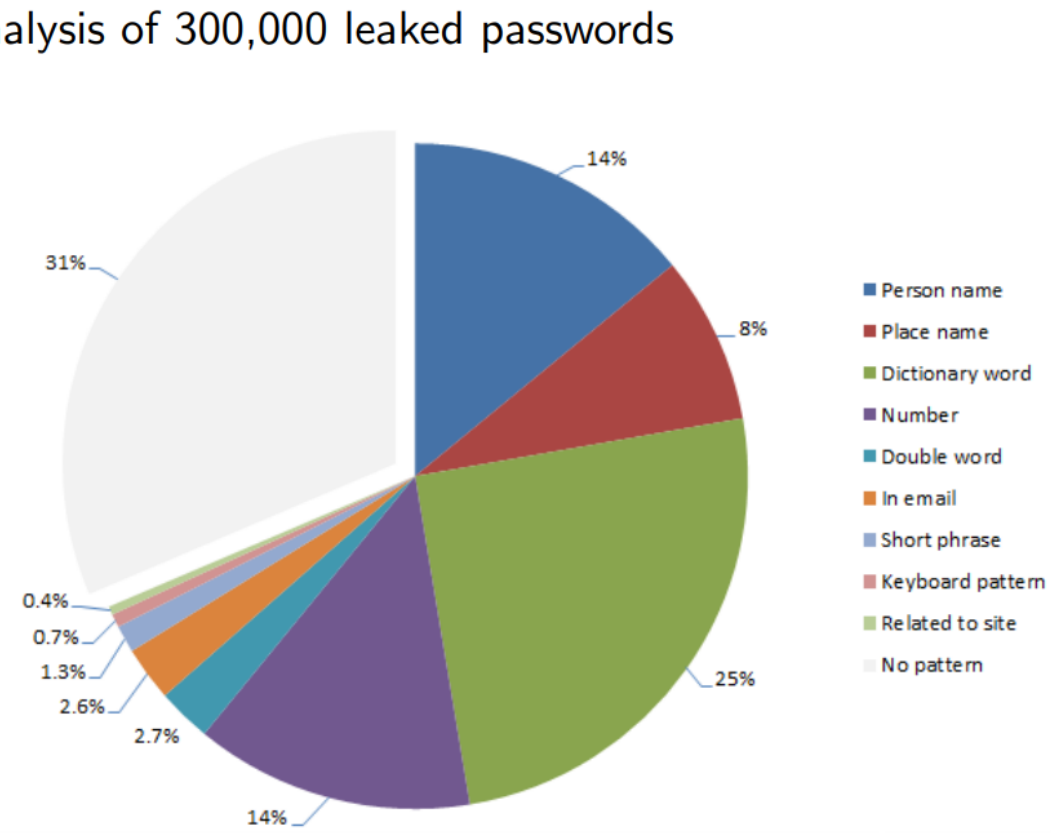
User Authentication, Access Control, and Operating System

Storing Passwords

Video Summary

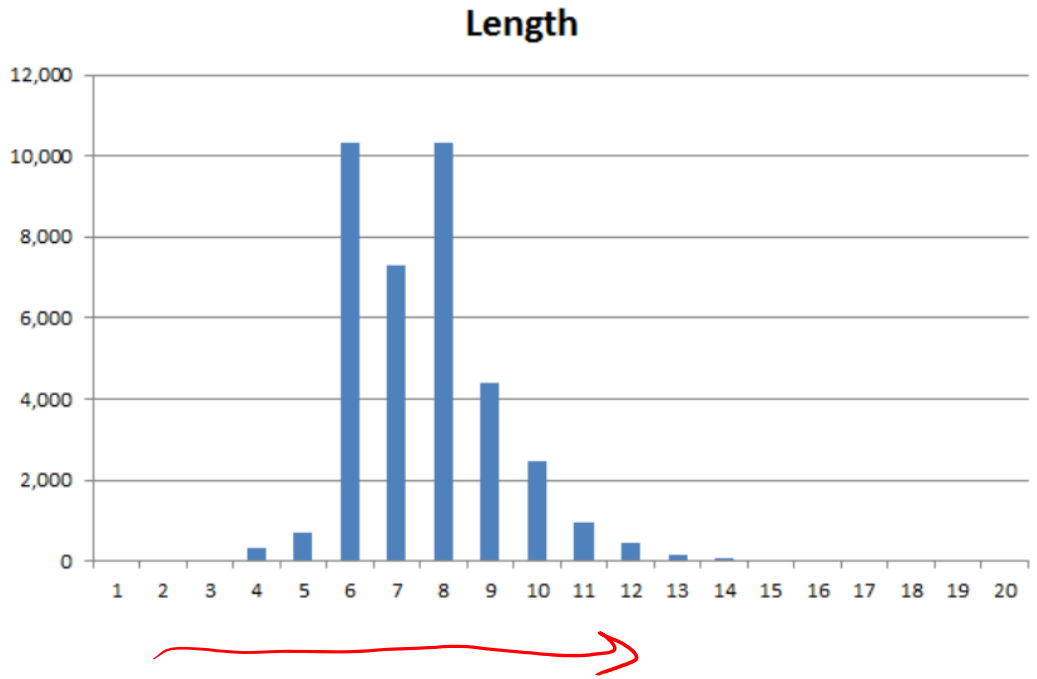
- How to Select A Password
- How to Store A Password
- Hashing Passwords

Selecting Passwords



How Long Are Passwords?

Analysis of 37,000 leaked passwords



Credit: Troy Hunt, *A brief Sony password analysis*, www.troyhunt.com, CCBY3.0

Other Common Characteristics of Passwords

poopoo
maximus
genius
cool
vampire
lacrosse
asd123
aaaa
christin
kimberly
speedy
sharon
carmen
111222
kristina
sammy
racing
ou812
sabrina
horses
0987654321
qwerty1
pimpin
baby
stalker
enigma
147147
star
poohbear
boobies
147258
simple
bollocks
12345q
marcus
brian
1987
qweasdzxc
drowssap
hahaha
caroline
barbara
dave
viper
drummer
action
einstein
bitches
genesis
hello1
scotty
friend
forest
010203

- Most use only alphanumeric characters
- Most are in (password) dictionaries
- Many users re-use passwords across systems
- Some very common passwords: 123456, password, 12345678, qwerty, abc123, letmein, iloveyou, . . .
- When forced to change passwords, most users change a single character

Storing Passwords

- Upon initial usage, user ID and password are registered with system
- ID, password (or information based on it), and optionally other user information stored on system, e.g. in file or database
- To access system, user submits ID and password, compared against stored values
- How should passwords be stored?

Storing Passwords

ID, P

Insider attack: normal user reads the database and learns other users passwords

- Countermeasure: access control on password database

Insider attack: admin user reads the database and learns other users passwords

- Countermeasure: none—admin users must be trusted!

Outsider attack: attacker gains unauthorised access to database and learns all passwords

- Countermeasure: do not store passwords in the clear

Encrypting Passwords

$$ID, E(K, P)$$

- Encrypted passwords are stored
- When user submits password, it is encrypted and compared to the stored value
- Drawback: Secret key, K, must be stored (on file or memory); if attacker can read database, then likely they can also read K

Hashing the Passwords

$$ID, H(P)$$

- Hashes of passwords are stored
- When user submits password, it is hashed and compared to the stored value
- Practical properties of hash functions:
  - Variable sized input; produce a fixed length, small output
  - No collisions
  - One-way function
- If attacker gains database, practically impossible to take a hash value and directly determine the original password

Hashing the Passwords

username	password
john	mysecret
sandy	ld9a%23f
daniel	mysecret
...	...
steve	h31p_m3?

Hashing the Passwords

username	password
john	mysecret
sandy	ld9a%23f
daniel	mysecret
...	...
steve	h31p_m3?

username	H(password)
john	06c219e5bc8378f3a8a3f83b4b7e4649
sandy	5fc2bb44573c7736badc8382b43fbeae
daniel	06c219e5bc8378f3a8a3f83b4b7e4649
...	...
steve	75127c78fd791c3f92a086c59c71ece0

Hash

Video Summary

- How to Select A Password
- How to Store A Password
- Hashing Passwords

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

User Authentication, Access Control, and Operating System

Cracking Passwords

Video Summary

- Brute Force Attack on Hashed Passwords
- Hashing Speed
- Preventing Hashing Attacks
- Rainbow Tables

Brute Force Attack on Hashed Passwords

- Aim: given one (or more) target hash value, find the original password
- Start with large set of possible passwords (e.g. from dictionary, all possible n-character combinations)
- Calculate hash of possible password, compare with target hash
  - if match, original password is found
  - else, try next possible password
- Attack duration depends on size of possible password set

Hashing the Passwords

username	H(password)
john	06c219e5bc8378f3a8a3f83b4b7e4649
sandy	5fc2bb44573c7736badc8382b43fbeae
daniel	06c219e5bc8378f3a8a3f83b4b7e4649
...	...
steve	75127c78fd791c3f92a086c59c71ece0

- Brute force on n-bit hash value: 2^n attempts
- For MD5 128 bit: 2^{128} attempts (how long does this take?)
- How many hashes your computer calculate per second?

4×10^6 hashes/sec

Hashing the Passwords

username	H(password)
john	06c219e5bc8378f3a8a3f83b4b7e4649
sandy	5fc2bb44573c7736badc8382b43fbeae
daniel	06c219e5bc8378f3a8a3f83b4b7e4649
...	...
steve	75127c78fd791c3f92a086c59c71ece0

- Brute force on n-bit hash value: 2^n attempts
- For MD5 128 bit: 2^{128} attempts (how long does this take?)
- How many hashes your computer calculate per second?
- @ 4×10^6 hashes/sec

$2^{128} / (4 \times 10^6 * 60 * 60 * 24) = 9.85 \times 10^{26}$ days

Hashing the Passwords

username	H(password)
john	06c219e5bc8378f3a8a3f83b4b7e4649
sandy	5fc2bb44573c7736badc8382b43fbeae
daniel	06c219e5bc8378f3a8a3f83b4b7e4649
...	...
steve	75127c78fd791c3f92a086c59c71ece0

- Brute force on n-bit hash value: 2^n attempts
- For MD5 128 bit: 2^{128} attempts (how long does this take?)
- How many hashes your computer calculate per second?
- @ $4x10^6$ hashes/sec
- How long ➔ 2^{128} /(4000000*60*60*24)=$9.85x10^{26}$ days = $2.7x10^{24}$ years

Hashing the Passwords

- What if we used a GPU? (gaming computers or mining hardware)
- 10^6 hashes/sec: still TOO LONG

Hashing the Passwords

- What if we used a GPU? (gaming computers or mining hardware)
- 10^6 hashes/sec: still TOO LONG
- What about using GPU and parallel computing?➔ 10^{10} hashes/sec
- How many passwords the user can choose from given that you have a maximum of 8 characters?

$1 \rightarrow 94$

$2 \rightarrow (94)^2$

$3 \rightarrow (94)^3$

$8 \rightarrow (94)^8$

Hashing the Passwords

- Worst case: $94^8 + 94^7 + 94^6 + 94^5 + 94^4 + 94^3 + 94^2 + 94^1 = 6.16 \times 10^{15}$ possible passwords

$$94^8 \simeq 6.16 \times 10^{15}$$

- If we are used a GPU (10^{10} hashes/sec).. How long it will take us to calculate the hashes of all passwords?

$$6.16 \times 10^{15} / (10^{10} \times 60 \times 60 \times 24)$$

7 days

Hashing the Passwords

- Worst case: $94^8 + 94^7 + 94^6 + 94^5 + 94^4 + 94^3 + 94^2 + 94^1 = 6.16x10^{15}$ possible passwords
- If we are used a GPU (10^{10} hashes/sec).. How long it will take us to calculate the hashes of all passwords?
- $6.16x10^{15} / (10^{10} *60*60*24) = 7$ days!!
- How to prevent such an attach?

Hashing the Passwords

- Worst case: $94^8 + 94^7 + 94^6 + 94^5 + 94^4 + 94^3 + 94^2 + 94^1 = 6.16 \times 10^{15}$ possible passwords
- If we are used a GPU (10^{10} hashes/sec).. How long it will take us to calculate the hashes of all passwords?
- $6.16 \times 10^{15} / (10^{10} * 60 * 60 * 24) = 7$ days!!
- How to prevent such an attach? ➔ Use a slower hash function

Hash Type	PC1	PC2
MD5	8581 Mh/s	2753 Mh/s
SHA1	3037 Mh/s	655 Mh/s
SHA256	1122 Mh/s	355 Mh/s
SHA512	414 Mh/s	104 Mh/s
SHA-3(Keccak)	179 Mh/s	92 Mh/s

Hashing the Passwords

- Worst case: 94^8 $94^7 + 94^6 + 94^5 + 94^4 + 94^3 + 94^2 + 94^1 = 6.16x10^{15}$ possible passwords
- If we are used a GPU (10^{10} hashes/sec).. How long it will take us to calculate the hashes of all passwords?
- $6.16x10^{15} / (10^{10} *60*60*24) = 7$ days!!
- How to prevent such an attach?
  - ✓ More characters in the password (for example 9 digits)

$(94)^9$

Cracking Passwords

- Store passwords and hash values in advance (instead of generating them)
- The question is how big is it?

  Password is 8 Bytes + hash is 128 bits (if using MD5)

  (8 Byte + 16 Byte) x 94^8 = 1.4x1017 Bytes = 146 TB (approx.)

- Instead of generating this huge amount of data we can use

Rainbow Tables

Cracking Passwords

➢Rainbow Tables

MD5 Rainbow Tables

Table ID	Charset	Plaintext Length	Key Space	Success Rate	Table Size
md5_ascii-32-95#1-7	ascii-32-95	1 to 7	70,576,641,626,495	99.9 %	52 GB 64 GB
md5_ascii-32-95#1-8	ascii-32-95	1 to 8	6,704,780,954,517,120	96.8 %	460 GB 576 GB
md5_mixalpha-numeric#1-8	mixalpha-numeric	1 to 8	221,919,451,578,090	99.9 %	127 GB 160 GB
md5_mixalpha-numeric#1-9	mixalpha-numeric	1 to 9	13,759,005,997,841,642	96.8 %	690 GB 864 GB

- Lookup on 0.5 TB Rainbow Table will take only hours to find the password

http://project-rainbowcrack.com/table.htm

Pre-calculated Hashes & Rainbow Tables

- How big is such a database of pre-calculated hashes?
  - In raw form, generally too big to be practical (100's, 1000's of TB)
  - Using specialised data structures (e.g. Rainbow tables), can obtain manageable size, e.g. 1 TB
- Trade-off: reduce search time, but increase storage space
- Countermeasures:
  - Longer passwords
  - Slower hash algorithms
  - Salting the password before hashing

Video Summary

- Brute Force Attack on Hashed Passwords
- Hashing Speed
- Preventing Hashing Attacks
- Rainbow Tables

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

User Authentication, Access Control, and Operating System

Rainbow Tables and Salting

Video Summary

- Rainbow Tables
- Salting Passwords
- Storing Passwords with Salt
- Modern Approaches

Cracking Passwords

- Store passwords and hash values in advance (instead of generating them)
- The question is how big is it?

Password is 8 Bytes + hash is 128 bits (if using MD5)

(8 Byte + 16 Byte) x 94^8 = 1.4x1017 Bytes = 146 TB (approx.)

Cracking Passwords

- Store passwords and hash values in advance (instead of generating them)
- The question is how big is it?

  Password is 8 Bytes + hash is 128 bits (if using MD5)

  (8 Byte + 16 Byte) x 94^8 = 1.4x1017 Bytes = 146 TB (approx.)

- Instead of generating this huge amount of data we can use

Rainbow Tables

Rainbow Tables

- A rainbow table is a precomputed dictionary of plaintext passwords and their corresponding hash values that can be used to find out what plaintext password produces a particular hash.
- Unlike brute-forcing, performing the hash function isn't the problem here. With all of the values already computed, it's simplified to just a simple search-and-compare operation on the table.

Cracking Passwords

➢Rainbow Tables

MD5 Rainbow Tables

Table ID	Charset	Plaintext Length	Key Space	Success Rate	Table Size	Files	Performance
md5_ascii-32-95#1-7	ascii-32-95	1 to 7	70,576,641,626,495	99.9 %	52 GB 64 GB	Perfect Non-perfect	Perfect Non-perfect
md5_ascii-32-95#1-8	ascii-32-95	1 to 8	6,704,780,954,517,120	96.8 %	460 GB 576 GB	Perfect Non-perfect	Perfect Non-perfect
md5_mixalpha-numeric#1-8	mixalpha-numeric	1 to 8	221,919,451,578,090	99.9 %	127 GB 160 GB	Perfect Non-perfect	Perfect Non-perfect
md5_mixalpha-numeric#1-9	mixalpha-numeric	1 to 9	13,759,005,997,841,642	96.8 %	690 GB 864 GB	Perfect Non-perfect	Perfect Non-perfect
md5_loweralpha-numeric#1-9	loweralpha-numeric	1 to 9	104,461,669,716,084	99.9 %	65 GB 80 GB	Perfect Non-perfect	Perfect Non-perfect
md5_loweralpha-numeric#1-10	loweralpha-numeric	1 to 10	3,760,620,109,779,060	96.8 %	316 GB 396 GB	Perfect Non-perfect	Perfect Non-perfect

- Lookup on 0.5 TB Rainbow Table will take only hours to find the password

http://project-rainbowcrack.com/table.htm

Pre-calculated Hashes & Rainbow Tables

- How big is such a database of pre-calculated hashes?
  - In raw form, generally too big to be practical (100's, 1000's of TB)
  - Using specialised data structures (e.g. Rainbow tables), can obtain manageable size, e.g. 1 TB
- Trade-off: reduce search time, but increase storage space
- Countermeasures:
  - Longer passwords
  - Slower hash algorithms
  - Salting the password before hashing

Salting Passwords

$$ID, Salt, H(P||Salt)$$

- When ID and password initially created, generate random s-bit value (salt), concatenate with password and then hash
- When user submits password, salt from password database is concatenated, hashed and compared
- If attacker gains database, they know the salt; same effort to find password as brute force attack
- BUT pre-calculated values (e.g. Rainbow tables) are no longer feasible
  - Space required increased by factor of 2^s

Salting Passwords

username	salt	H(password || salt)
john	a4H*1	ba586dcb7fe85064d7da80ea6361ddb6
sandy	U9(-f	816a425628d5dee17839fffeafb67144
daniel	5<as4	11842ced4203d4067ed6a6667f3f18d9
...	...	...
steve	LqM4^	184b7f9c6126c568ee50cd3364257973

- The attacker now knows the user name, the salt, and the hash
- How he is going to get the original password?
- How long this will take (worst case) ?

Salting Passwords

username	salt	H(password || salt)
john	a4H*1	ba586dcb7fe85064d7da80ea6361ddb6
sandy	U9(-f	816a425628d5dee17839fffeafb67144
daniel	5<as4	11842ced4203d4067ed6a6667f3f18d9
...	...	...
steve	LqM4^	184b7f9c6126c568ee50cd3364257973

- The attacker now knows the user name, the salt, and the hash
- How he is going to get the original password?
- How long this will take (worst case) ?
- What is the benefit of using Salt?

Salting Passwords

- If we used a 16-bits salt then the number of salts available are 2^{16}
- To use the Rainbow Table you have to generate a Rainbow table for each possible salt value
- 65536 Rainbow Tables x 0.5 TB per table = 23768 TB
- Another benefit of using salt is that if two users have the same password, they will have different hash values

username	password
john	mysecret
sandy	ld9a%23f
daniel	mysecret
...	...
steve	h31p_m3?

username	H(password)
john	06c219e5bc8378f3a8a3f83b4b7e4649
sandy	5fc2bb44573c7736badc8382b43fbeae
daniel	06c219e5bc8378f3a8a3f83b4b7e4649
...	...
steve	75127c78fd791c3f92a086c59c71ece0

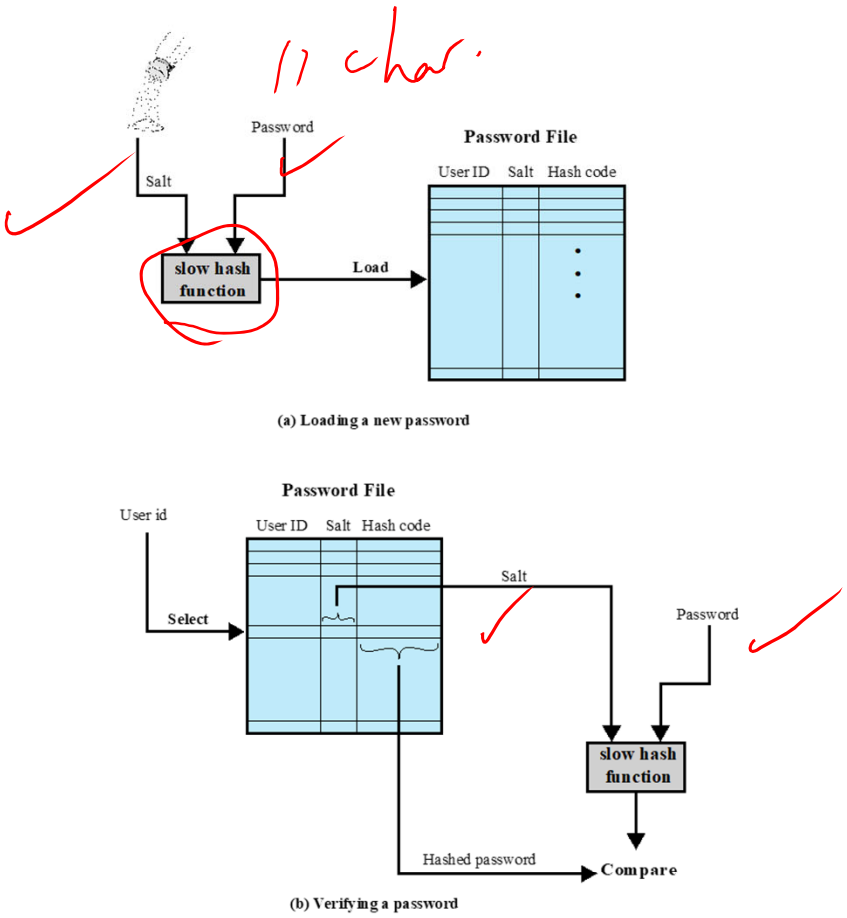


Figure 3.3 UNIX Password Scheme

Password Storage: Best Practice

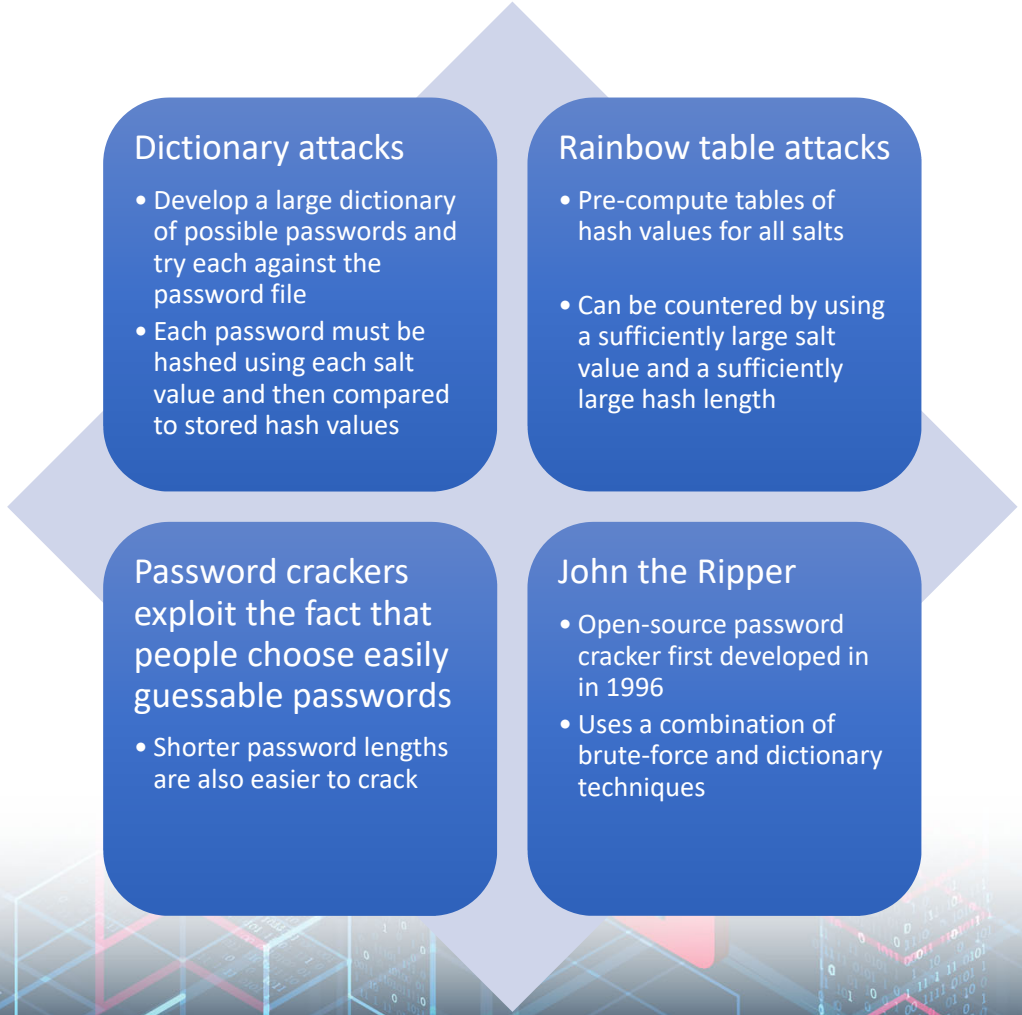
When storing user login information, always store a hash of a salted password

$$ID, Salt, H(P||Salt)$$

- Salt: random, generated when ID/password first stored; 32 bits or longer
- Hash function: slow, adaptive speed (work factor), e.g. bcrypt/scrypt, PBKDF2

Design for failure: assume password database will eventually be compromised

Password Cracking

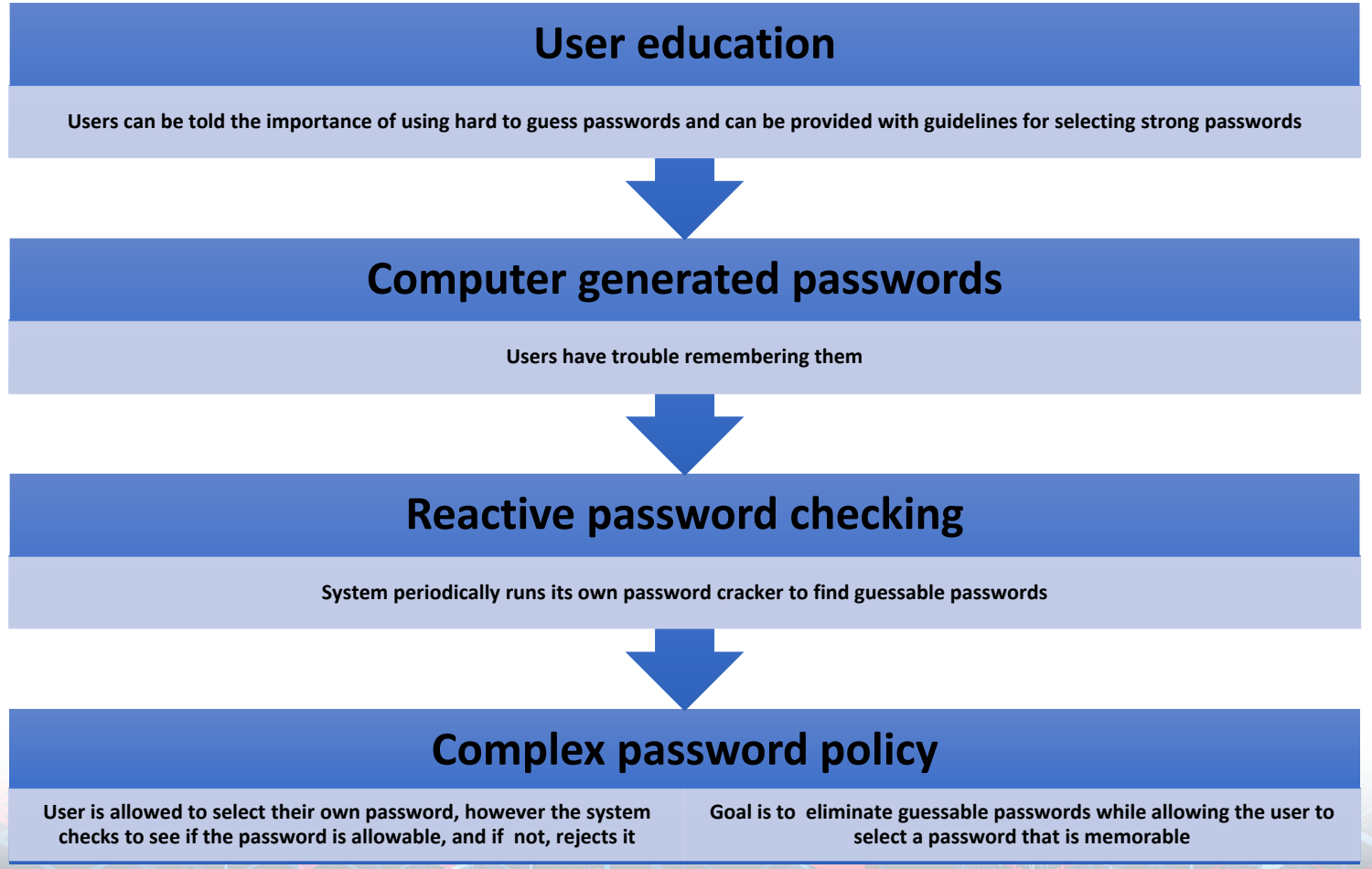


Modern Approaches

- Complex password policy
  - Forcing users to pick stronger passwords

- However password-cracking techniques have also improved
  - The processing capacity available for password cracking has increased dramatically
  - The use of sophisticated algorithms to generate potential passwords
  - Studying examples and structures of actual passwords in use

Password Selection Strategies



Video Summary

- Rainbow Tables
- Salting Passwords
- Storing Passwords with Salt
- Modern Approaches

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

User Authentication, Access Control, and Operating System

Token-based Authentication

Video Summary

- Token-based Authentication
- Biometric Authentication

Token-Based Authentication

Objects that a user possesses for purpose of user authentication are called tokens

Card Type	Defining Feature	Example
Embossed	Raised characters only, on front	Old credit card
Magnetic stripe	Magnetic bar on back, characters on front	Bank card
Memory	Electronic memory inside	Prepaid phone card
Smart Contact Contactless	Electronic memory and processor inside Electrical contacts exposed on surface Radio antenna embedded inside	Biometric ID card

Memory Cards

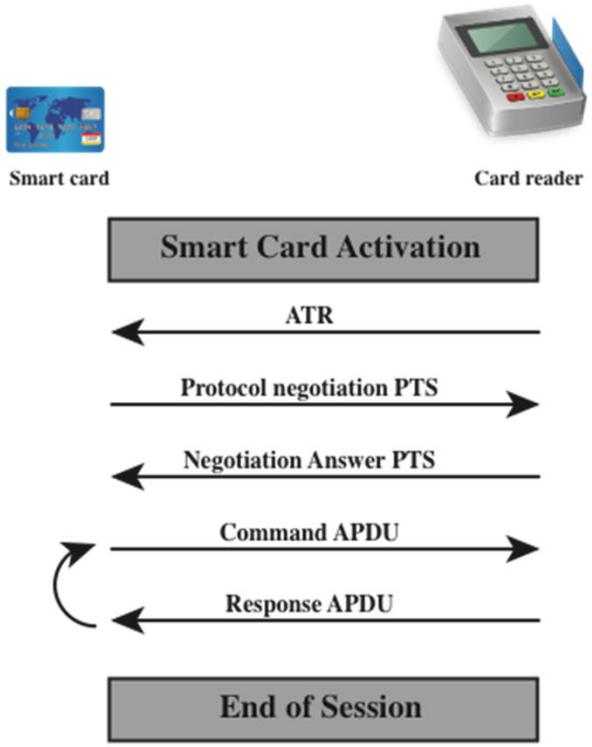
- Can store but do not process data
- The most common is the magnetic stripe card
- Can include an internal electronic memory
- Can be used alone for physical access
  - Hotel room
  - ATM
- Provides significantly greater security when combined with a password or PIN
- Drawbacks of memory cards include:
  - Requires a special reader
  - Loss of token
  - User dissatisfaction

Smart Tokens

- **Physical characteristics:**
  - Include an embedded microprocessor
  - A smart token that looks like a bank card
  - Can look like calculators, keys, small portable objects
- **User interface:**
  - Manual interfaces include a keypad and display for human/token interaction
- **Electronic interface**
  - A smart card or other token requires an electronic interface to communicate with a compatible reader/writer
  - Contact and contactless interfaces
- **Authentication protocol:**
  - Classified into three categories:
    - Static
    - Dynamic password generator
    - Challenge-response

Smart Cards

- Most important category of smart token
  - Has the appearance of a credit card
  - Has an electronic interface
  - May use any of the smart token protocols
- Contain:
  - An entire microprocessor
    - Processor
    - Memory
    - I/O ports
- Typically include three types of memory:
  - Read-only memory (ROM)
    - Stores data that does not change during the card's life
  - Electrically erasable programmable ROM (EEPROM)
    - Holds application data and programs
  - Random access memory (RAM)
    - Holds temporary data generated when applications are executed



APDU = application protocol data unit
ATR = Answer to reset
PTS = Protocol type selection

Figure 3.6 Smart Card/Reader Exchange

Electronic Identity Cards (eID)

Use of a smart card as a national identity card for citizens

Can serve the same purposes as other national ID cards, and similar cards such as a driver's license, for access to government and commercial services

Can provide stronger proof of identity and can be used in a wider variety of applications

In effect, is a smart card that has been verified by the national government as valid and authentic

Most advanced deployment is the German card *neuer Personalausweis*

Has human-readable data printed on its surface

- Personal data
- Document number
- Card access number (CAN)
- Machine readable zone (MRZ)

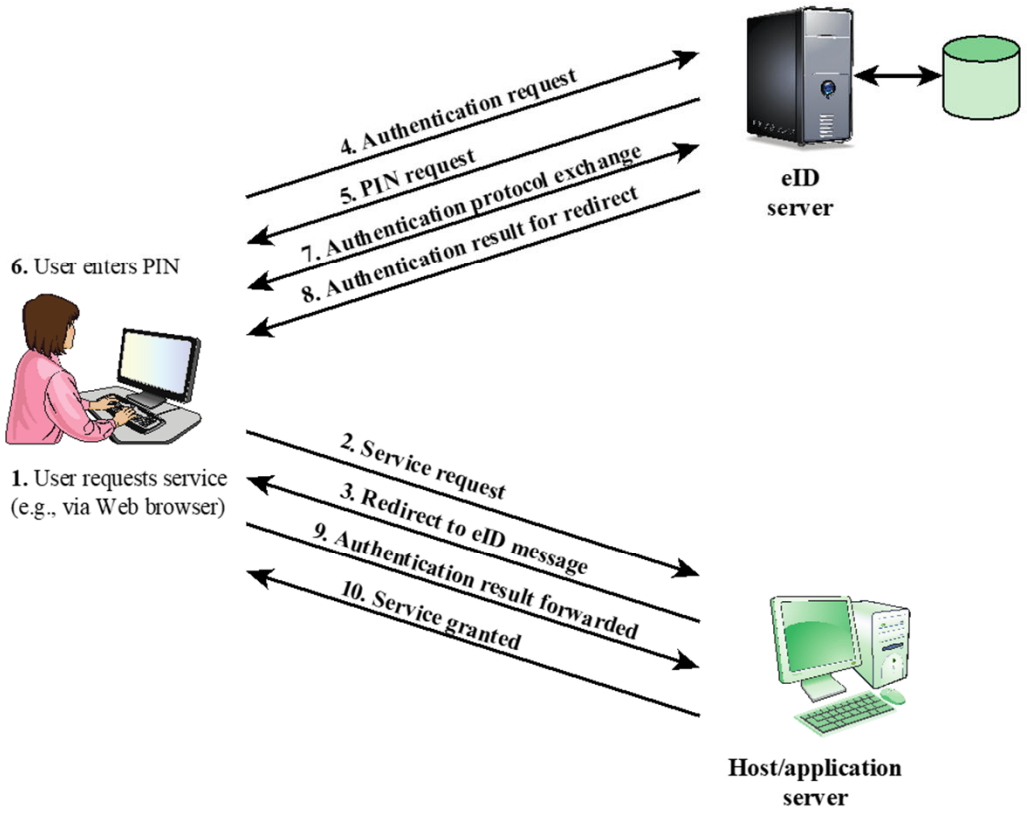


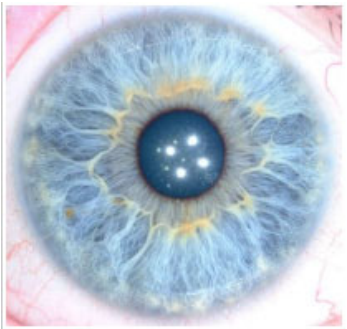
Figure 3.7 User Authentication with eID

Biometric Authentication

- Attempts to authenticate an individual based on unique physical characteristics
- Based on pattern recognition
- Is technically complex and expensive when compared to passwords and tokens
- Physical characteristics used include:
  - Facial characteristics
  - Fingerprints
  - Hand geometry
  - Retinal pattern (blood vessels in eyeball)
  - Iris (color pattern of your eye)
  - Signature
  - Voice

Retina vs. Iris

The Retina



The Iris

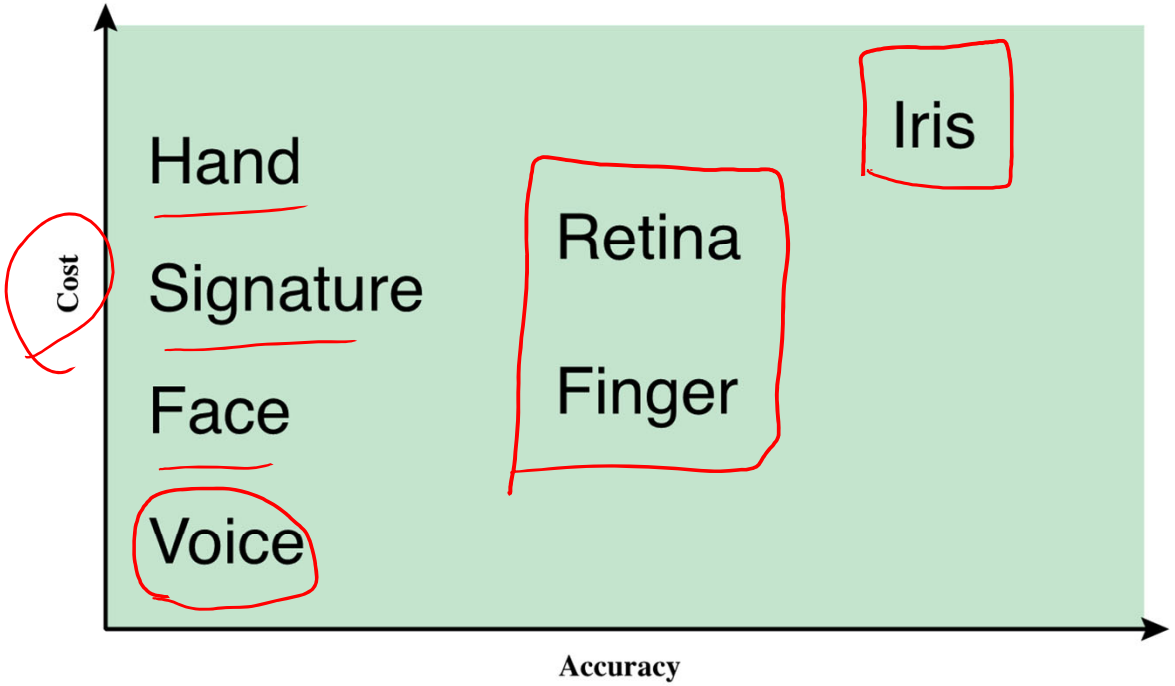


Figure 3.8 Cost Versus Accuracy of Various Biometric Characteristics in User Authentication Schemes.

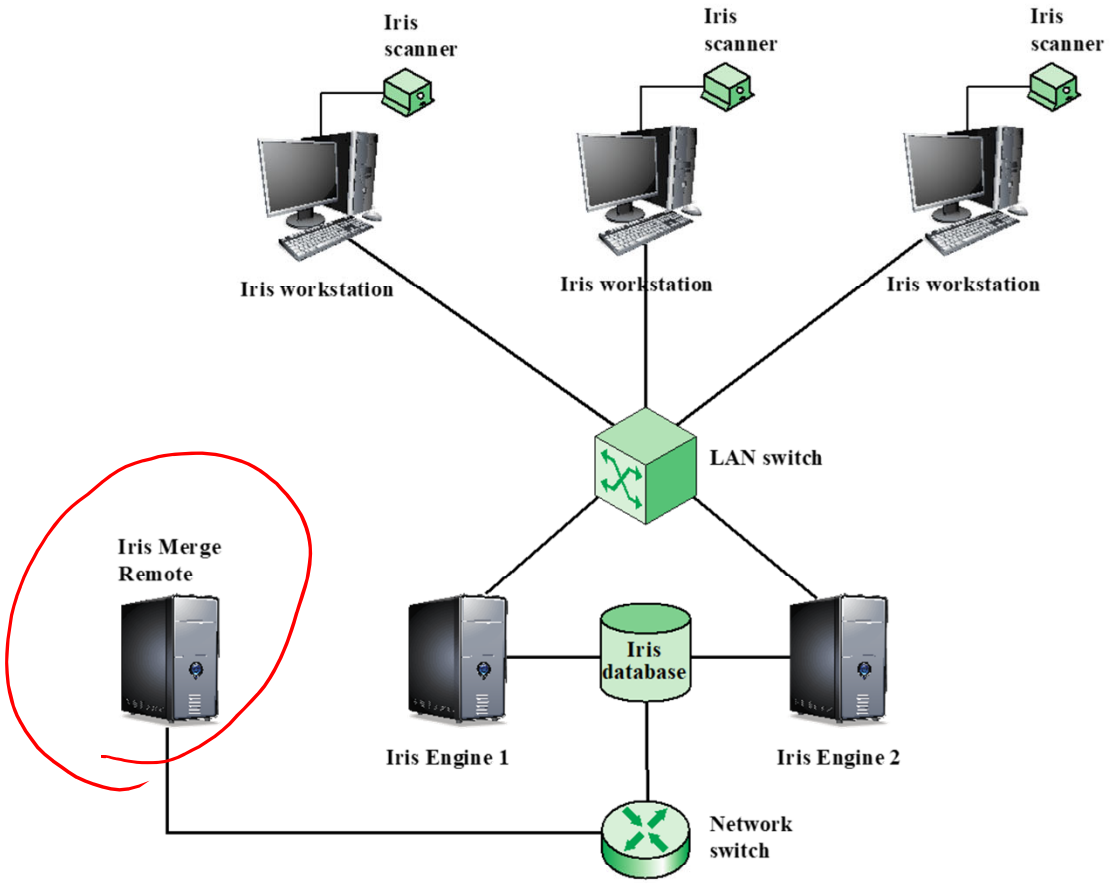


Figure 3.14 General Iris Scan Site Architecture for UAE System

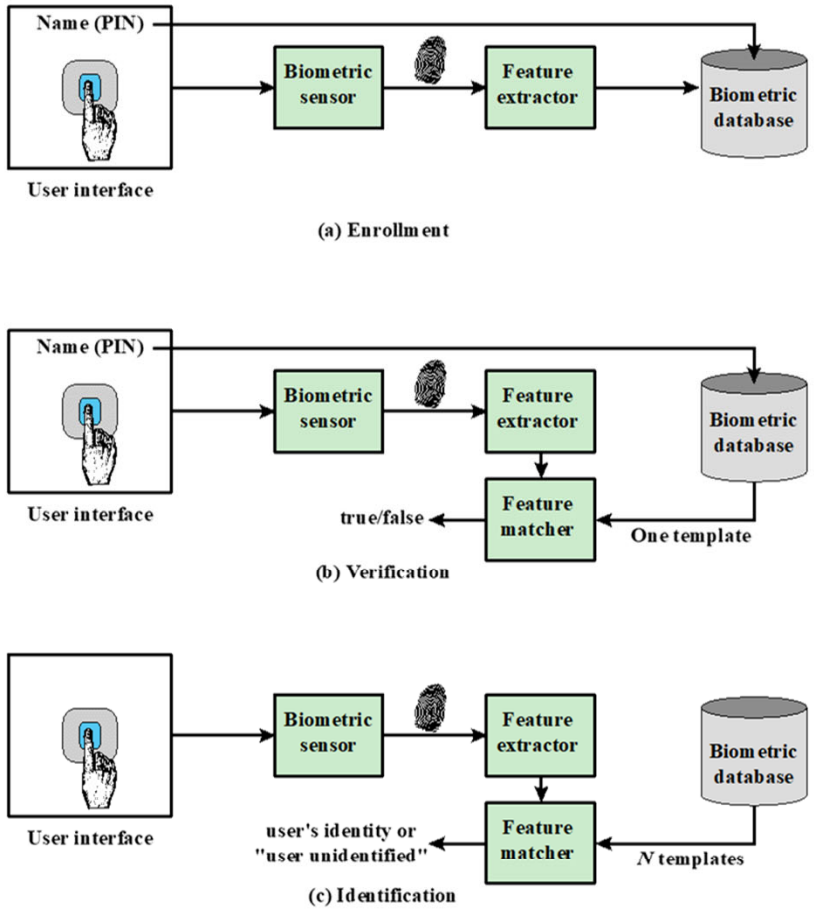


Figure 3.9 A Generic Biometric System. Enrollment creates an association between a user and the user's biometric characteristics. Depending on the application, user authentication either involves verifying that a claimed user is the actual user or identifying an unknown user.

Video Summary

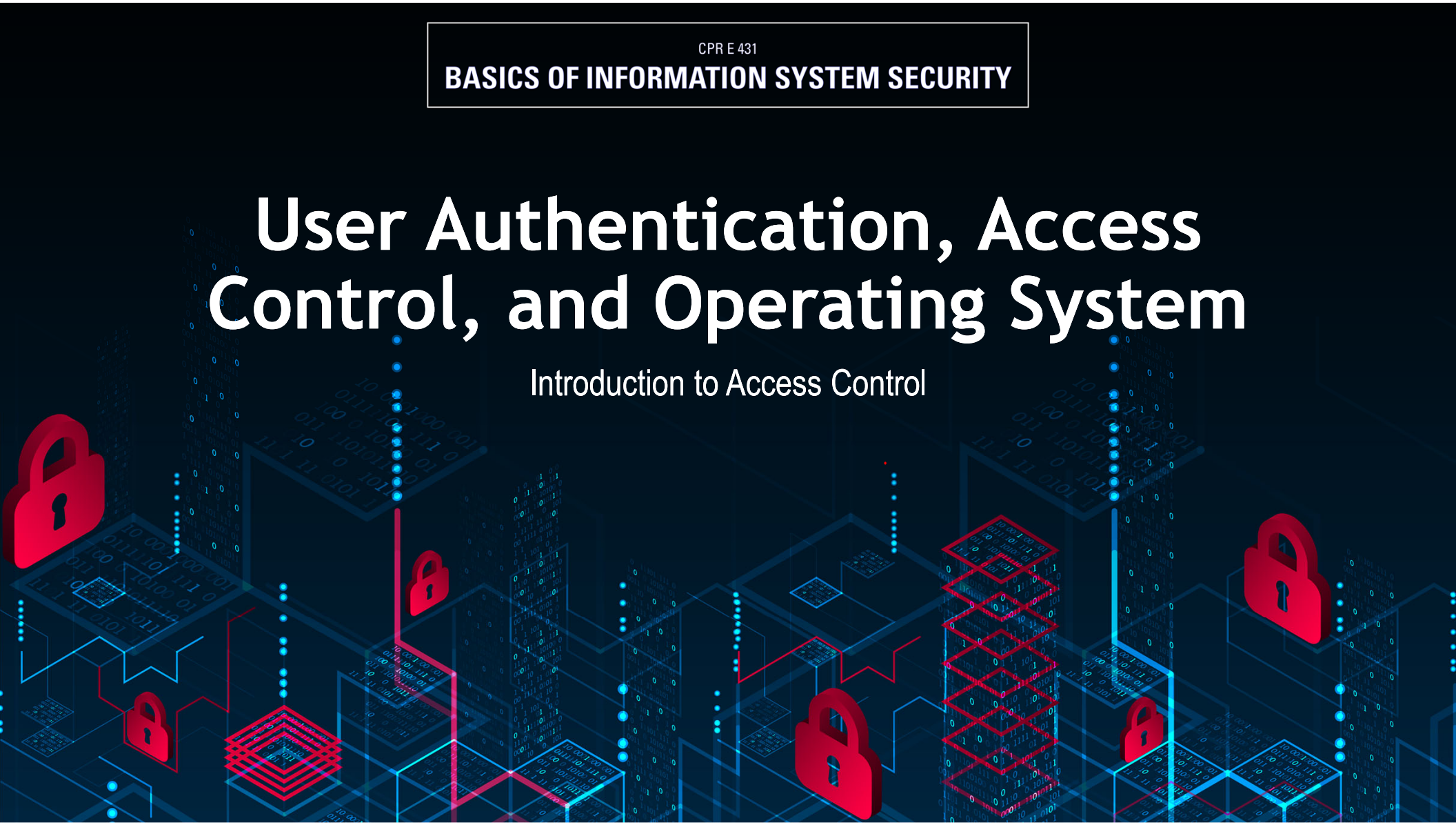
- Token-based Authentication
- Biometric Authentication

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

User Authentication, Access Control, and Operating System

Introduction to Access Control



Video Summary

- What is Access Control (AC)
- Subjects, Objects, and Access Rights
- General Requirements of Access Control

Access Control Definitions 1/2

NIST 7298 defines access control as:

"The process of granting or denying specific requests to:

(1) obtain and use information and related information processing services
(2) enter specific physical facilities"

Access Control Definitions 2/2

RFC 4949 defines access control as:

> "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"

Access Control Principles

- In a broad sense, all of computer security is concerned with access control

- RFC 4949 defines computer security as:

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

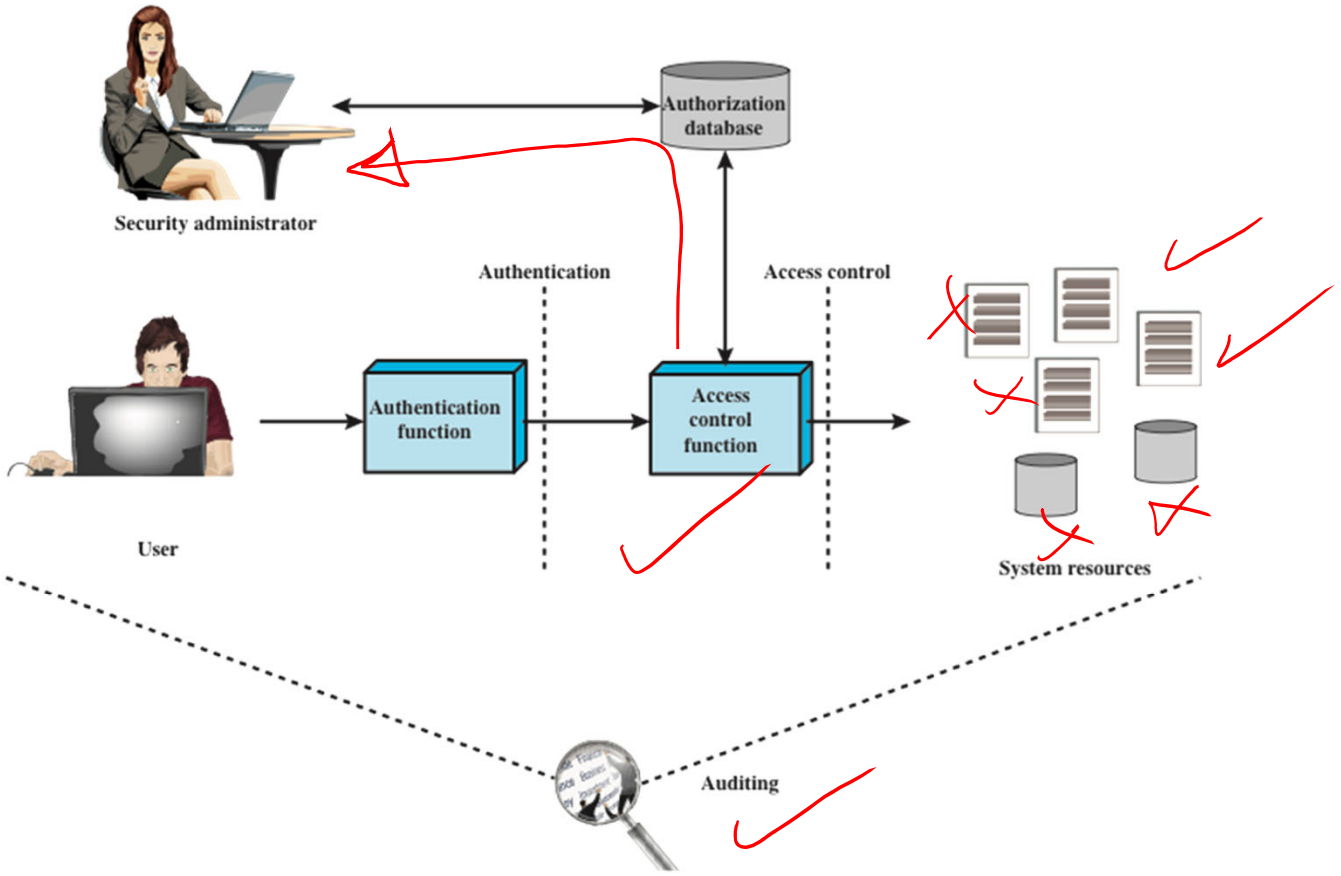
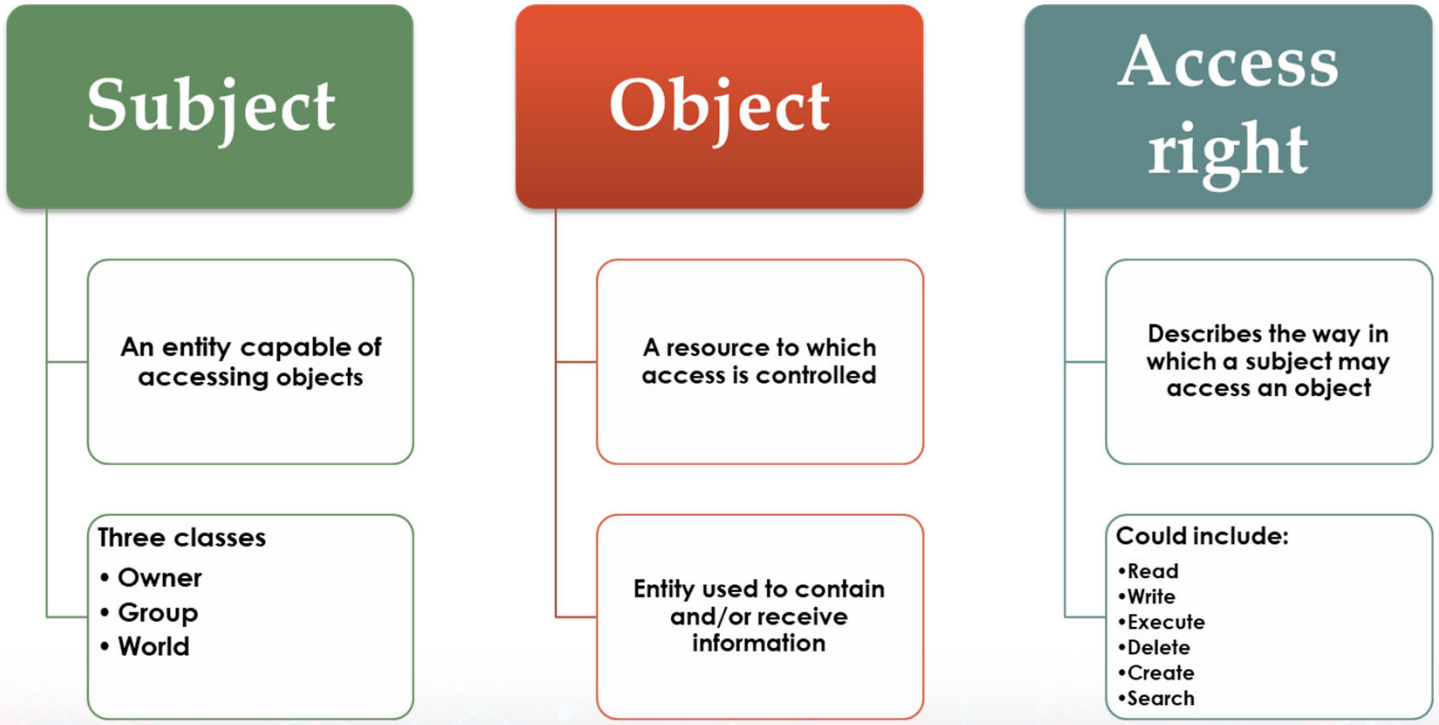


Figure 4.1 Relationship Among Access Control and Other Security Functions

Access Control & Security Functions

Authentication	verification that the credentials of a user or other entity are valid
Authorization	granting of a right or permission to a system entity to access a resource
Audit	independent review of system records and activities in order to test for adequacy of system control, ensure compliance to policy, detect breaches and recommend changes

Subjects, Objects, and Access Rights



General Requirements of Access Control

- Reliable input
- Fine and coarse specifications
- Least privilege
- Separation of duty
- Open and closed policies
- Policy combinations and conflict resolution
- Administrative policies
- Dual control

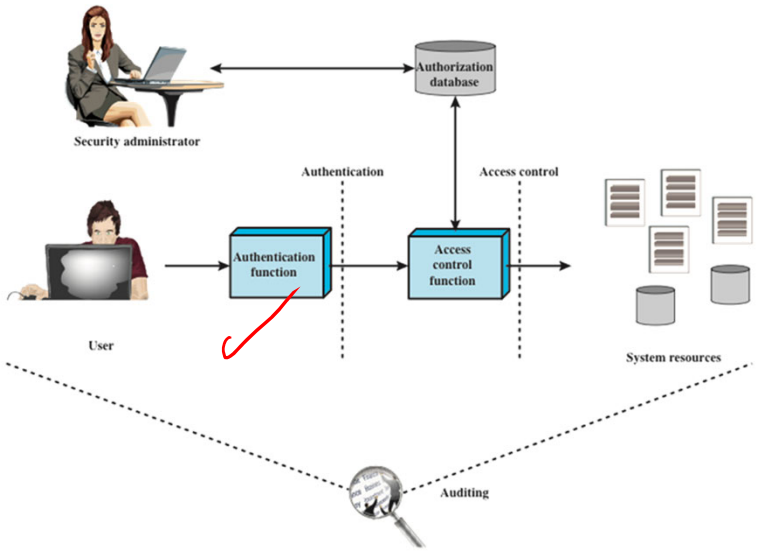


Figure 4.1 Relationship Among Access Control and Other Security Functions

Access Control Policies

- Discretionary access control (DAC)
  - Controls access based on the identity of the requestor and on access rules (authorizations) stating what requestors are (or are not) allowed to do
- Mandatory access control (MAC)
  - Controls access based on comparing security labels with security clearances
- Role-based access control (RBAC)
  - Controls access based on the roles that users have within the system and on rules stating what accesses are allowed to users in given roles

Video Summary

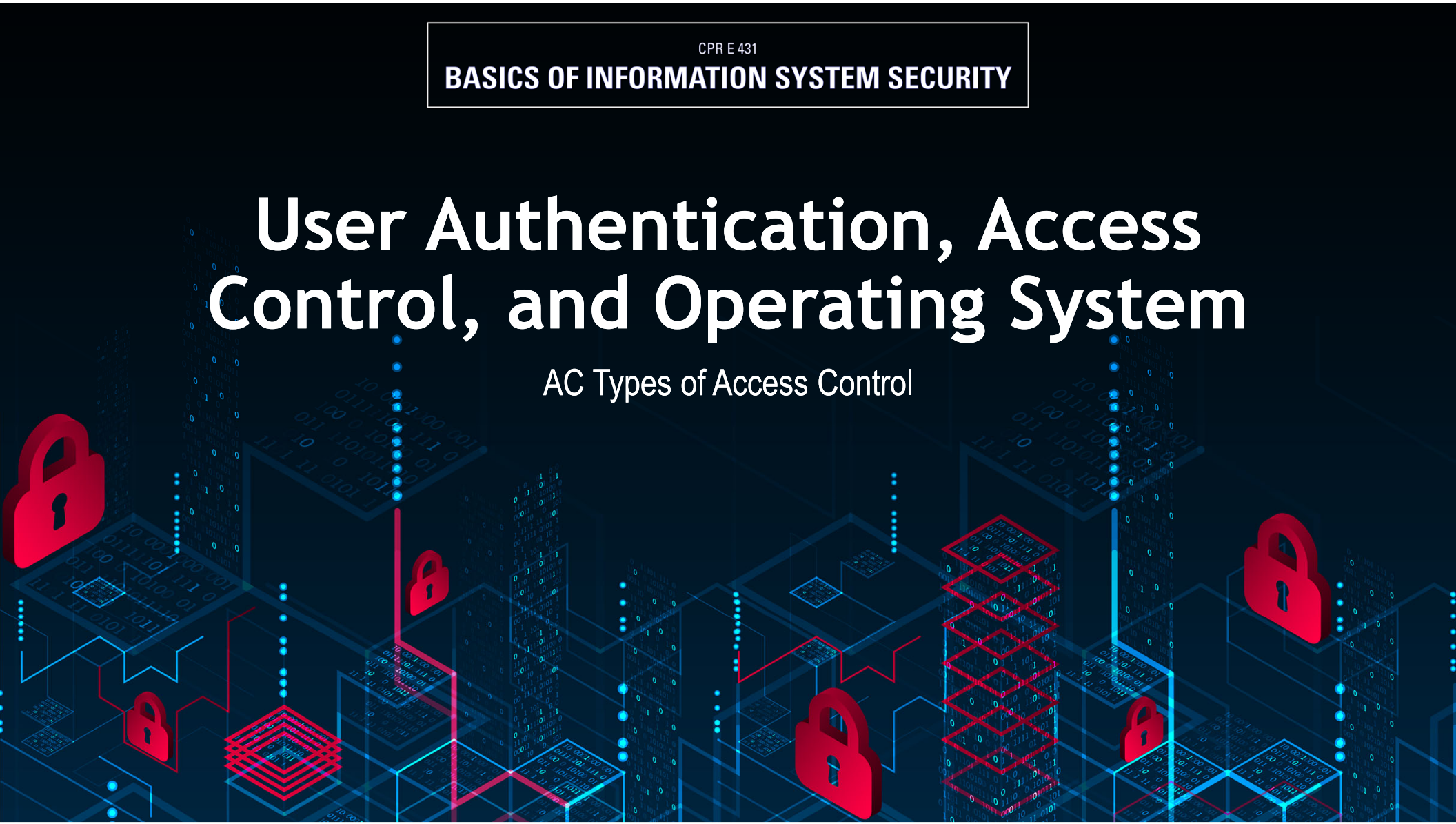
- What is Access Control (AC)
- Subjects, Objects, and Access Rights
- General Requirements of Access Control

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

User Authentication, Access Control, and Operating System

AC Types of Access Control



Video Summary

- What is Discretionary Access Control (DAC)
- What is Role-based Access Control (RBAC)
- What are the limitations of RBAC
- What is Attribute-based Access Control (ABAC)
- What is Mandatory Access Control (MAC)

Discretionary Access Control (DAC)

- DAC: an entity may be granted access rights that permit the entity, if they choose so, to enable another entity to access a resource
- Common access control scheme in operating systems and database management systems
- Access Matrix specifies access rights of subjects on objects

Discretionary Access Control (DAC)

- In practice, access matrix is sparse, so implement as either:

  Access Control Lists (ACL) For each object, list subjects and their access rights

  Capability Lists For each subject, list objects and the rights the subject have on that object

- Alternative implementation: authorization table listing subject, access mode and object; easily implemented in database

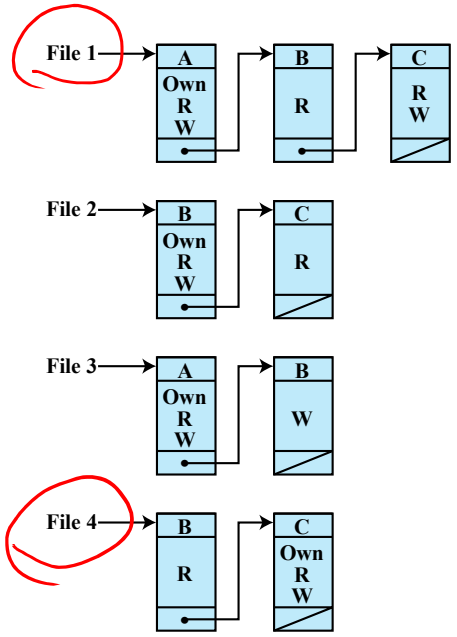
Example of DAC: Access Matrix

SUBJECTS	OBJECTS: File 1	File 2	File 3	File 4
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

Figure 4.2 Example of Access Control Structures

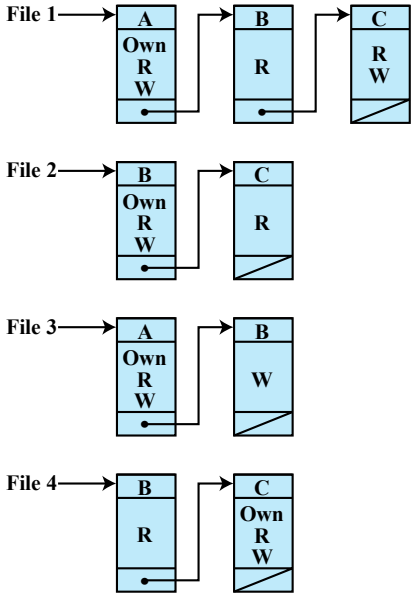
Example of DAC: Access Control List



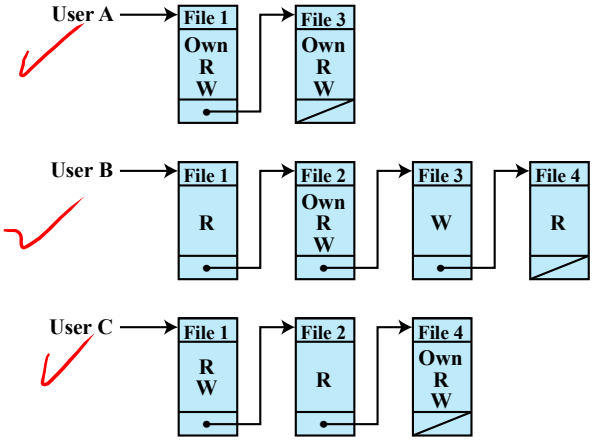
(b) Access control lists for files of part (a)

Figure 4.2 Example of Access Control Structures

Example of DAC: Capability lists



(b) Access control lists for files of part (a)



(c) Capability lists for files of part (a)

Figure 4.2 Example of Access Control Structures

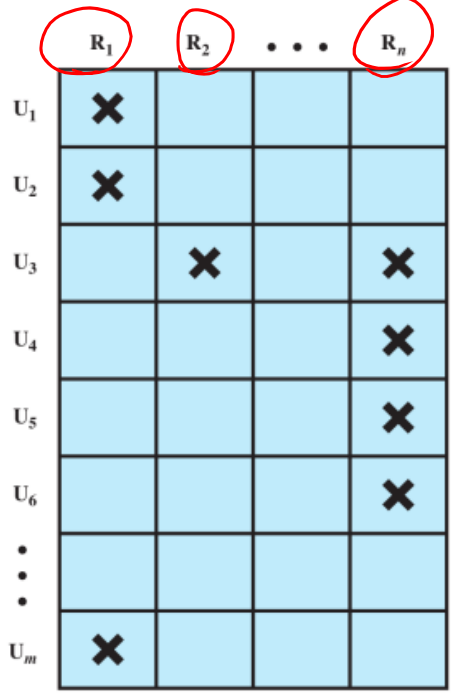
Example of Authorization Table

Subject	Access Mode	Object
A	Own	File 1
A	Read	File 1
A	Write	File 1
A	Own	File 3
A	Read	File 3
A	Write	File 3
B	Read	File 1
B	Own	File 2
B	Read	File 2
B	Write	File 2
B	Write	File 3
B	Read	File 4
C	Read	File 1
C	Write	File 1
C	Read	File 2
C	Own	File 4
C	Read	File 4
C	Write	File 4

Role-Based Access Control

- RBAC: users are assigned to roles; access rights are assigned to roles
- Roles typically job functions and positions within organisation, e.g. senior financial analyst in a bank, doctor in a hospital
- Users may be assigned multiple roles; static or dynamic
- Sessions are temporary assignments of user to role(s)
- Access control matrix can map users to roles and roles to objects

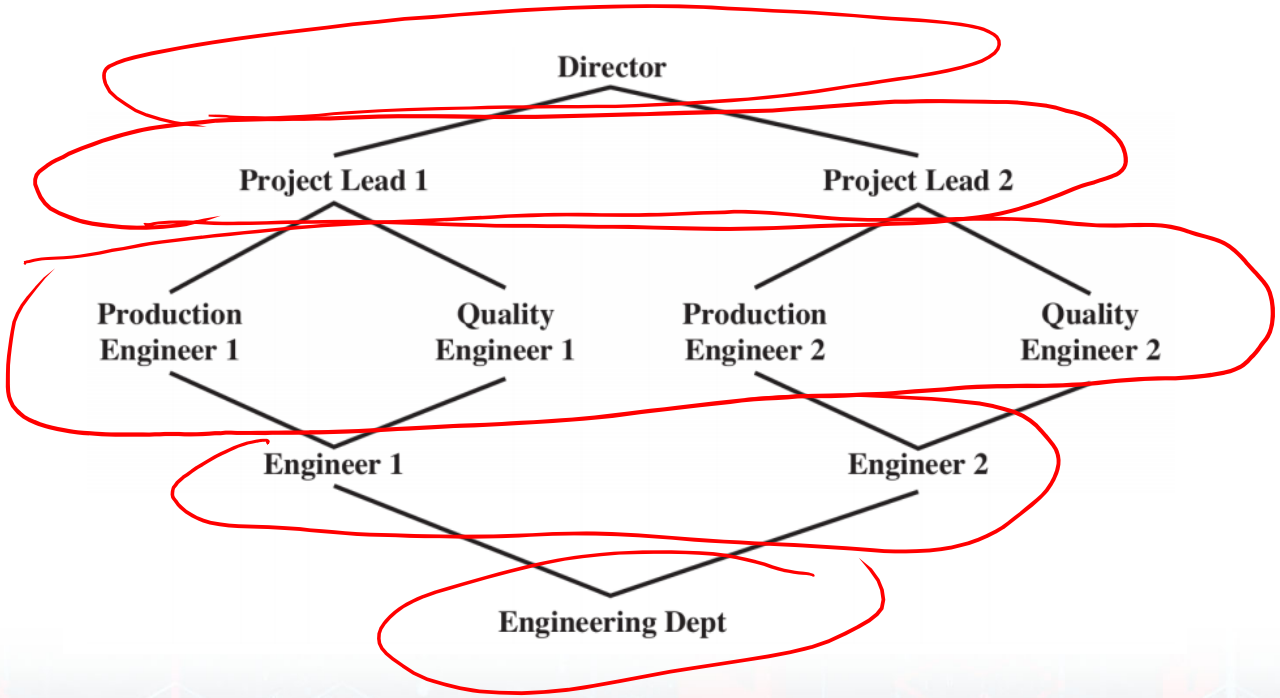
Role-Based Access Control Matrix



ROLES	OBJECTS								
	R1	R2	Rn	F1	F1	P1	P2	D1	D2
R1	control	owner	owner control	read *	read owner	wakeup	wakeup	seek	owner
R2		control		write *	execute			owner	seek *
Rn			control		write	stop			

Hierarchies in RBAC

- Hierarchy of an organisation can be reflected in roles
- A higher role includes all access rights of lower role



Constraints in RBAC

- Constraints define relationships between roles or conditions on roles
- A higher role includes all access rights of lower role
- Mutually exclusive roles: user can only be assigned to one role in the set

Constraints in RBAC

- Constraints define relationships between roles or conditions on roles
- A higher role includes all access rights of lower role
- Mutually exclusive roles: user can only be assigned to one role in the set
- Cardinality: maximum number with respect to roles, e.g.
  - maximum number of users assigned to a role
  - maximum number of roles a user can be assigned to
  - maximum number of roles that can be granted particular access rights

Constraints in RBAC

- Constraints define relationships between roles or conditions on roles
- A higher role includes all access rights of lower role
- Mutually exclusive roles: user can only be assigned to one role in the set
- Cardinality: maximum number with respect to roles, e.g.
  - maximum number of users assigned to a role
  - maximum number of roles a user can be assigned to
  - maximum number of roles that can be granted particular access rights
- Prerequisite: condition upon which user can be assigned a role, e.g.
  - user can only be assigned a senior role if already assigned a junior role

Video Summary

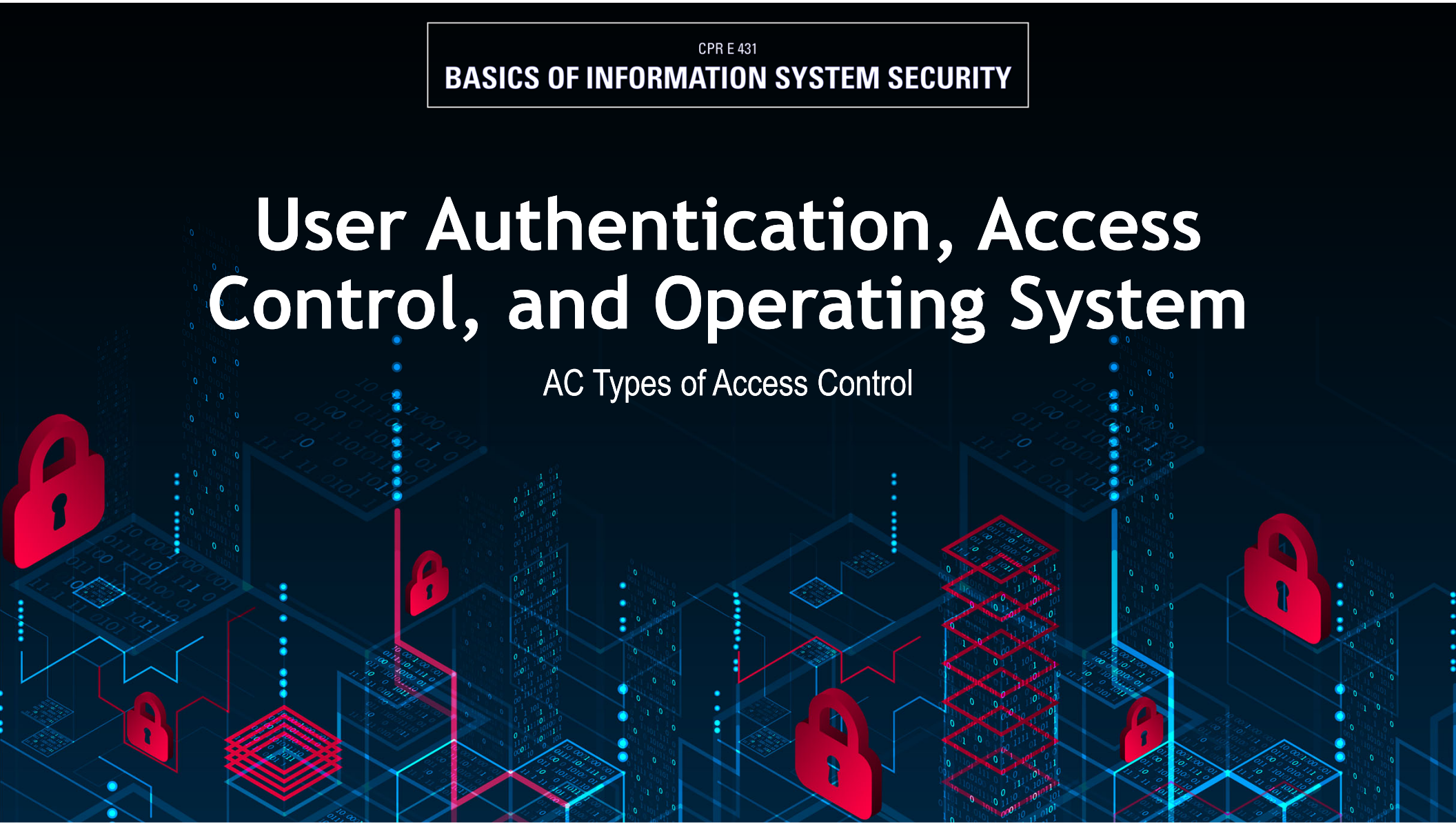
- What is Discretionary Access Control (DAC)
- What is Role-based Access Control (RBAC)
- What are the limitations of RBAC
- What is Attribute-based Access Control (ABAC)
- What is Mandatory Access Control (MAC)

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

User Authentication, Access Control, and Operating System

AC Types of Access Control



Video Summary

- What is Discretionary Access Control (DAC)
- What is Role-based Access Control (RBAC)
- What are the limitations of RBAC
- What is Attribute-based Access Control (ABAC)
- What is Mandatory Access Control (MAC)

How does RBAC work?

- Administrators assign access permissions to roles
- Then, roles can be assigned to individual users
  - Users may have one or several roles (each with different access rights)
- Administrators can simply update roles or access permissions
  - By assigning users (or removing users from) to the appropriate roles

The Limitations of RBAC

- RBAC provides static access control configurations.
- It fails to provide a flexible mechanism by which users/entities can express their requirements.

- Limitation #1: Role Explosion
  - RBAC is limited to defining access permissions by role
  - An ever-increasing number of users requires an exponentially increasing number of roles to accommodate various permission combinations

The Limitations of RBAC

- Limitation #2: Toxic Combinations
  - Various roles assigned to a given user could contain conflicting data.
    - One user may have a role allowing him to create a purchase order, and another allowing him to approve it.

The Limitations of RBAC

- Limitation #3: Management Nightmares
  - Between growing numbers of users, and exponentially more roles
  - Administrators have to constantly be on top of changes to users and to roles, and ensure that role assignment combinations are current, accurate, and not conflicting with other roles a user might be assigned.

The Limitations of RBAC

- Limitation #4: Lack of Context
  - Due to the static nature of Role Based Access Control, RBAC is unable to model policies that depend on contextual details:
    - Time-of-day, location, relationship between users, etc.
  - RBAC has no way of determining the relationships between users and using that information to make policy decisions.
  - At its best, RBAC was originally designed to answer just one question:

What access does a user have based on their assigned role(s)?

The Limitations of RBAC

- Today, defining authorization policies based on a user's role is not good enough.
- The context surrounding that user, their data, and the interaction between the two are also important to provide access to
  - the right user,
  - at the right time,
  - in the right location,
  - and by meeting regulatory compliance.
- That means evolving an existing Role Based Access Control model to an Attribute Based Access Control (ABAC) model

Evolving RBAC with ABAC

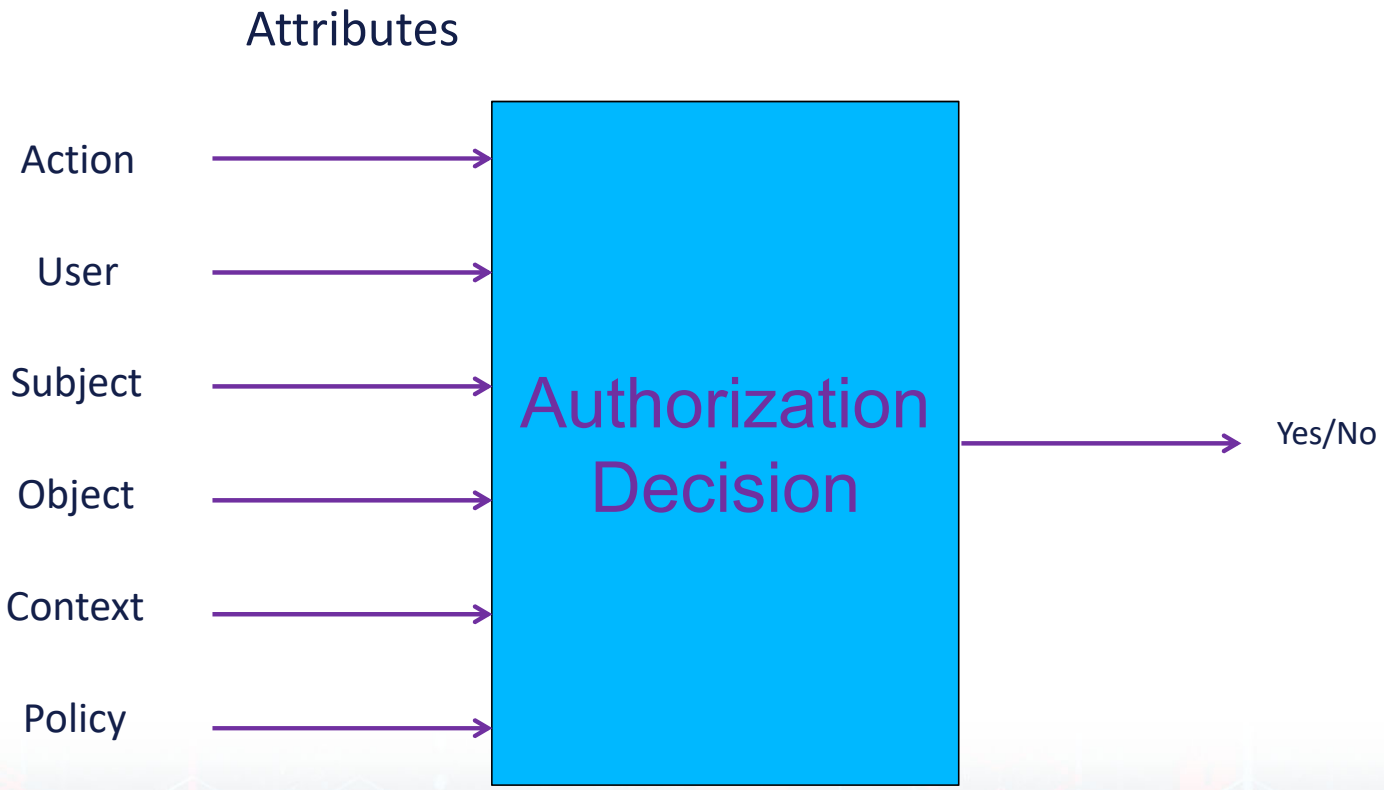
- Attribute Based Access Control allows an enterprise to extend existing roles using attributes and policies.
- By adding context, authorization decisions can be made based on:
  - Role of the user
  - Who or what that user is related to
  - What that user needs access to
  - Where that user needs access from
  - When that user needs access
  - How that user is accessing that information
- For example, a policy may be written as follows:
  - "Doctors can view medical records of any patient in their department and update any patient record that is directly assigned to them, during working hours and from an approved device."

Attribute based access control

- Similar to RBAC in the sense that it also adopts a policy driven approach.
- Uses attributes of subjects, objects, and the environment (instead of roles).

More suitable in adapting to dynamic access requirements in e-Health

Attribute based access control



Mandatory Access Control (MAC)

- Based on multilevel security (MLS)

  top secret $>$ secret $>$ confidential $>$ restricted $>$ unclassified

- Subject has security clearance of a given level
- Object has security classification of a given level

Mandatory Access Control (MAC)

- Based on multilevel security (MLS)

  top secret > secret > confidential > restricted > unclassified

- Subject has security clearance of a given level
- Object has security classification of a given level
- Two required properties for confidentiality:

  No read up Subject can only read an object of less or equal security level

  No write down Subject can only write into object of greater or equal security level

- Clearance and classification is determine by administrator; users cannot override security policy

Video Summary

- What is Discretionary Access Control (DAC)
- What is Role-based Access Control (RBAC)
- What are the limitations of RBAC
- What is Attribute-based Access Control (ABAC)
- What is Mandatory Access Control (MAC)

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

User Authentication, Access Control, and Operating System

AC UNIX/LINUX Access Control

Video Summary

- How UNIX/LINUX Files are Administered?
- What is inodes?
- UNIX/LINUX File Access Control
- DEMO

UNIX File Access Control

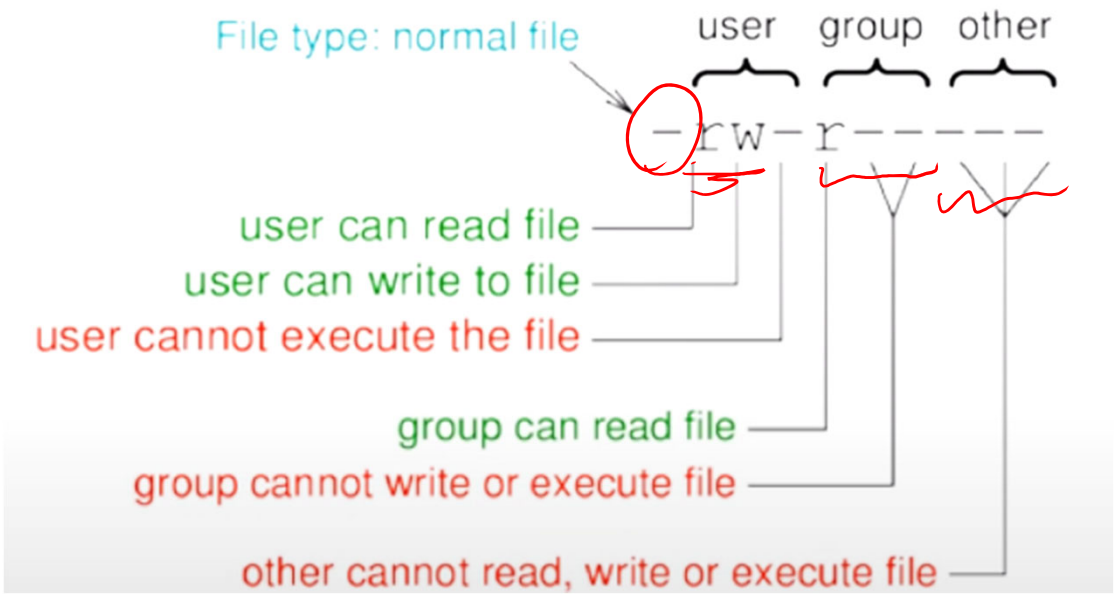
UNIX files are administered using inodes (index nodes)

- Control structures with key information needed for a particular file
- Several file names may be associated with a single inode
- An active inode is associated with exactly one file
- File attributes, permissions and control information are sorted in the inode
- On the disk there is an inode table, or inode list, that contains the inodes of all the files in the file system
- When a file is opened its inode is brought into main memory and stored in a memory resident inode table

Directories are structured in a hierarchical tree

- May contain files and/or other directories
- Contains file names plus pointers to associated inodes

Protection bits in an inode



inode

- Files and directories administered by operating system using inodes
- inode is data structure that stores important information about a file or directory
  - mode
  - owner information
  - size
  - timestamps
  - pointers to data blocks (data blocks contain the actual file)
- OS maintains list of inodes in inode table

inode Contents

mode 16 bits

- 12 protection bits: permissions
- 4 bit file type: regular file, directory, ...

owner id 16 bit user ID

group id 16 bit group ID

size size of file in bytes

timestamps last time, in seconds since epoch:

- atime: inode accessed
- ctime: inode changed
- mtime: file data modified

and other fields ...

Permissions and Users

Permissions

- read the file; list the contents of the directory
- write to the file; create and remove files in the directory
- execute the file; access files in the directory

Categories of Users

- user that owns the file
- users in the file's group
- other users
- (all users, i.e. the above three)

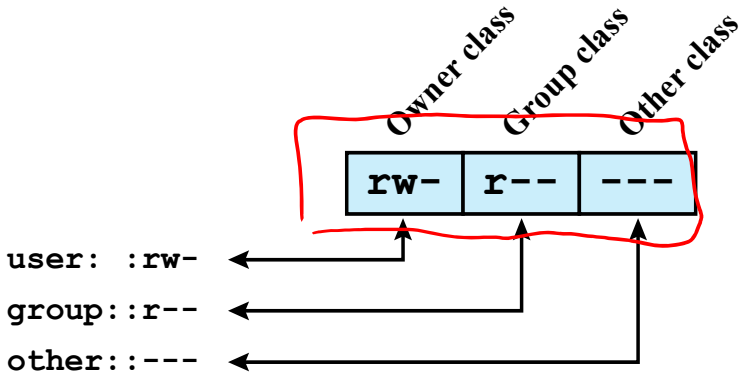
Permissions and Users

Special Permissions

- setuid bit: Set the process's effective user ID to that of the file
  - Directory: files created in that directory are given same user owner as the directory
- setgid bit: Set the process's effective group ID to that of the file
  - Directory: files created in that directory are given same group owner as the directory
- sticky bit: prevent users from removing or renaming a file unless they are user owner

UNIX File Access Control (DEMO)

- Unique user identification number (user ID)
- Member of a primary group identified by a group ID
- Belongs to a specific group
- 12 protection bits
  - Specify read, write, and execute permission for the owner of the file, members of the group and all other users
- The owner ID, group ID, and protection bits are part of the file's inode



(a) Traditional UNIX approach (minimal access control list)

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

User Authentication, Access Control, and Operating System

Operating System Security

Video Summary

- Operating System Strategy
- System Security Planning
- System Security Planning Process
- Operating System Hardening

Strategies

- The 2010 Australian Signals Directorate (ASD) lists the "Top 35 Mitigation Strategies"
- Over 85% of the targeted cyber intrusions investigated by ASD in 2009 could have been prevented
- The top four strategies for prevention are:
  - White-list approved applications
  - Patch third-party applications and operating system vulnerabilities
  - Restrict administrative privileges
  - Create a defense-in-depth system (layered defense mechanisms which increases the security of a system as a whole)

Strategies

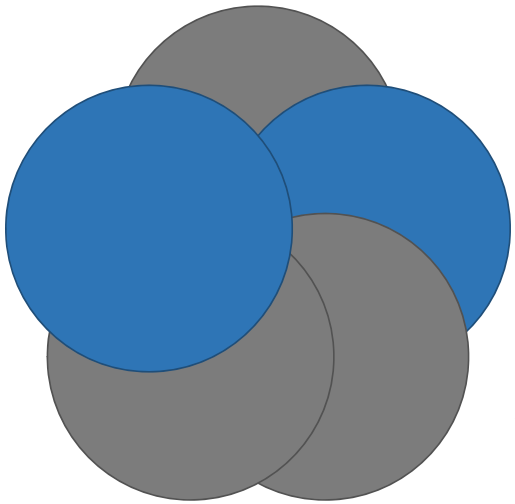
- Since 2013 these top four strategies are mandatory for all Australian government agencies.

- These strategies largely align with those in the “20 Critical Controls” developed by DHS, NSA, the Department of Energy, SANS, and others in the United States

System Security Planning

The first step in deploying a new system is planning

Plan needs to identify appropriate personnel and training to install and manage the system

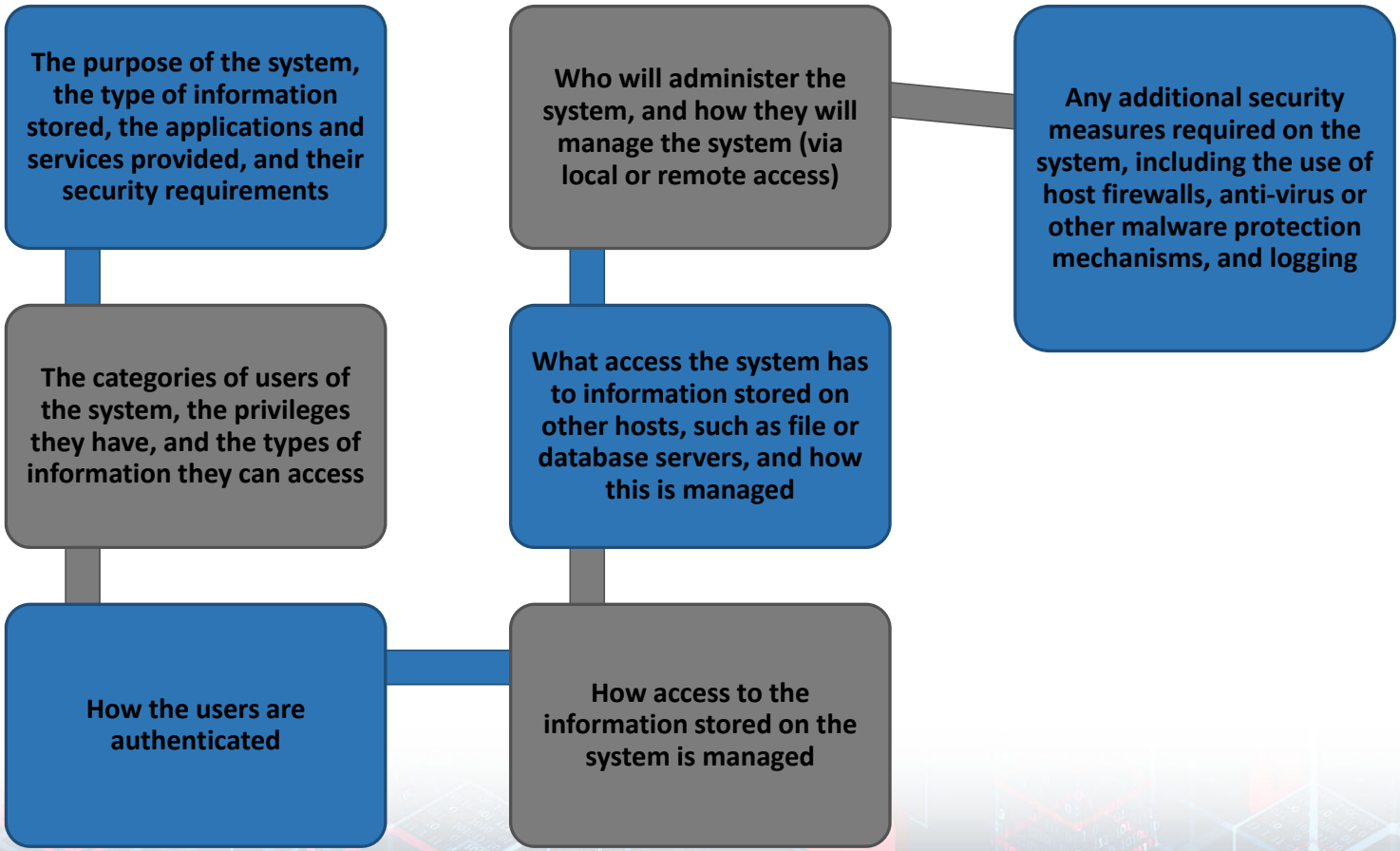


Planning should include a wide security assessment of the organization

Planning process needs to determine security requirements for the system, applications, data, and users

Aim is to maximize security while minimizing costs

System Security Planning Process



Operating Systems Hardening

- While the details of how to secure each specific operating system differ, the broad approach is similar.

- Appropriate security configuration guides and checklists exist for most common operating systems, and these should be consulted by the specific needs of each organization and their systems.

Operating Systems Hardening

- First critical step in securing a system is to secure the base operating system
- Basic steps
  - Install and patch the operating system
  - Harden and configure the operating system to adequately address the indentified security needs of the system by:
    - Removing unnecessary services, applications, and protocols
    - Configuring users, groups, and permissions
    - Configuring resource controls

Operating Systems Hardening

- Install and configure additional security controls, such as anti-virus, host-based firewalls, and intrusion detection system (IDS)

- Test the security of the basic operating system to ensure that the steps taken adequately address its security needs

Video Summary

- Operating System Strategy
- System Security Planning
- System Security Planning Process
- Operating System Hardening

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

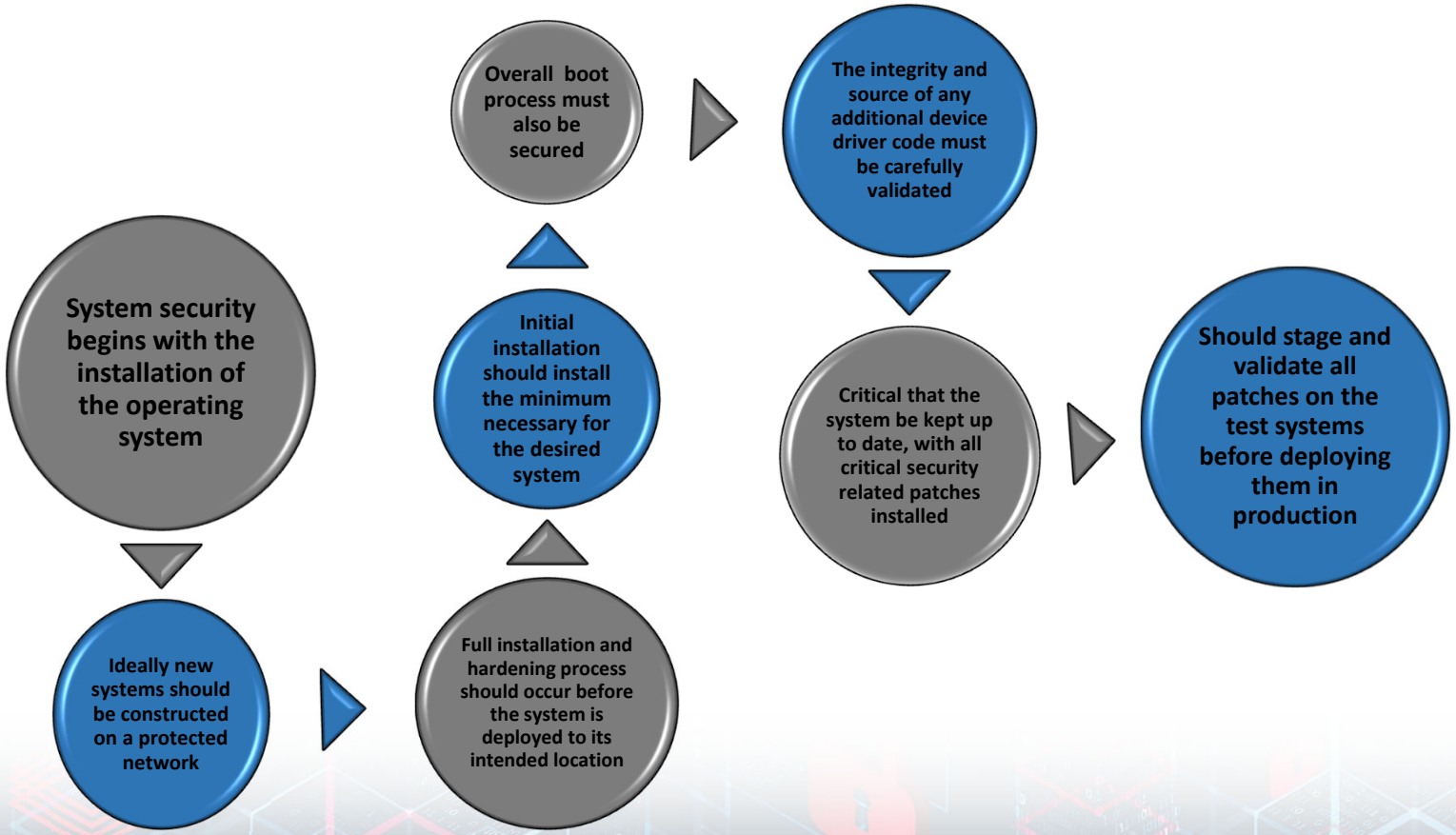
User Authentication, Access Control, and Operating System

Operating System Security

Video Summary

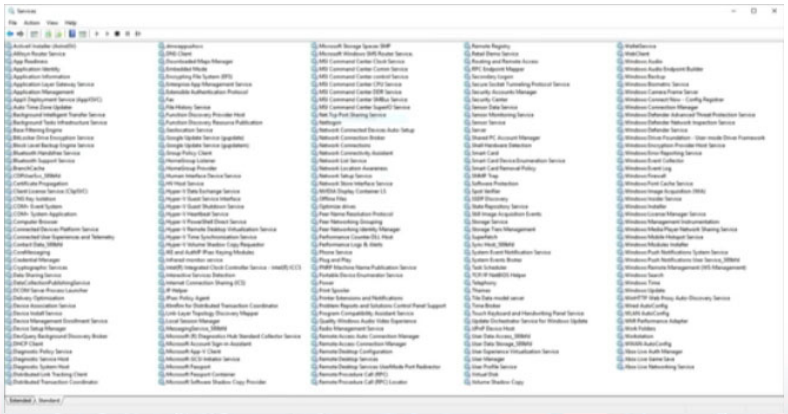
- OS Initial Setup and Patching
- OS Application Configuration
- Encryption Technology
- Security Maintenance
- Logging and Backup

Initial Setup and Patching

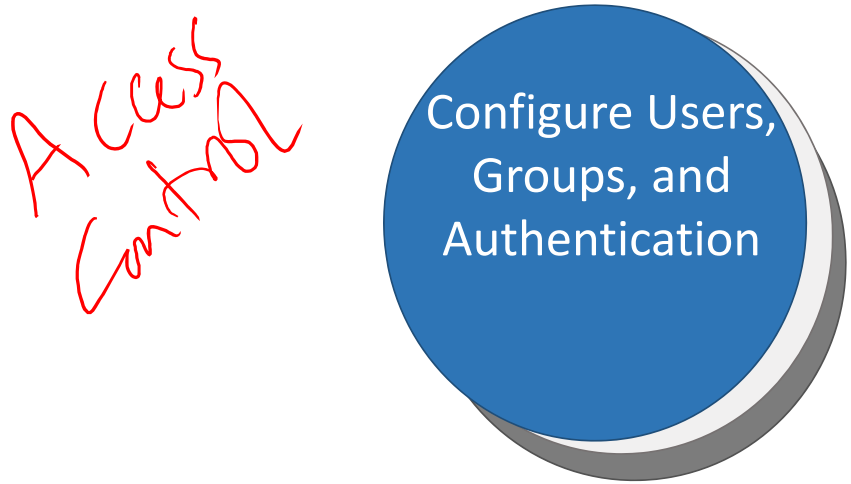


Remove Unnecessary Services, Applications, Protocols

- If **fewer software packages** are available to run the risk is reduced



- When performing the initial installation the supplied **defaults should not be used**
  - Default configuration is set to maximize ease of use and functionality rather than security
  - Windows 10 has over 240 services by default
  - If additional packages are needed later they can be installed when they are required



- Not all users with access to a system will have the same access to all data and resources on that system
- **Elevated privileges** should be restricted to only those users that require them, and then only when they are needed to perform a task

- **System planning process should consider:**
  - Categories of users on the system
  - Privileges they have
  - Types of information they can access
  - How and where they are defined and authenticated
- **Default accounts included as part of the system installation should be secured**
  - Those that are not required should be either removed or disabled
  - Policies that apply to authentication credentials configured

- Once the users and groups are defined, appropriate **permissions can be set on data and resources**
- Many of the security hardening guides provide lists of recommended changes to the default access configuration

- Further security possible by installing and configuring additional security tools:
  - Anti-virus software
  - Host-based firewalls
  - IDS or IPS software
  - Application white-listing

- Final step in the process of initially securing the base operating system is security testing
- Goal:
  - Ensure the previous security configuration steps are correctly implemented
  - Identify any possible vulnerabilities

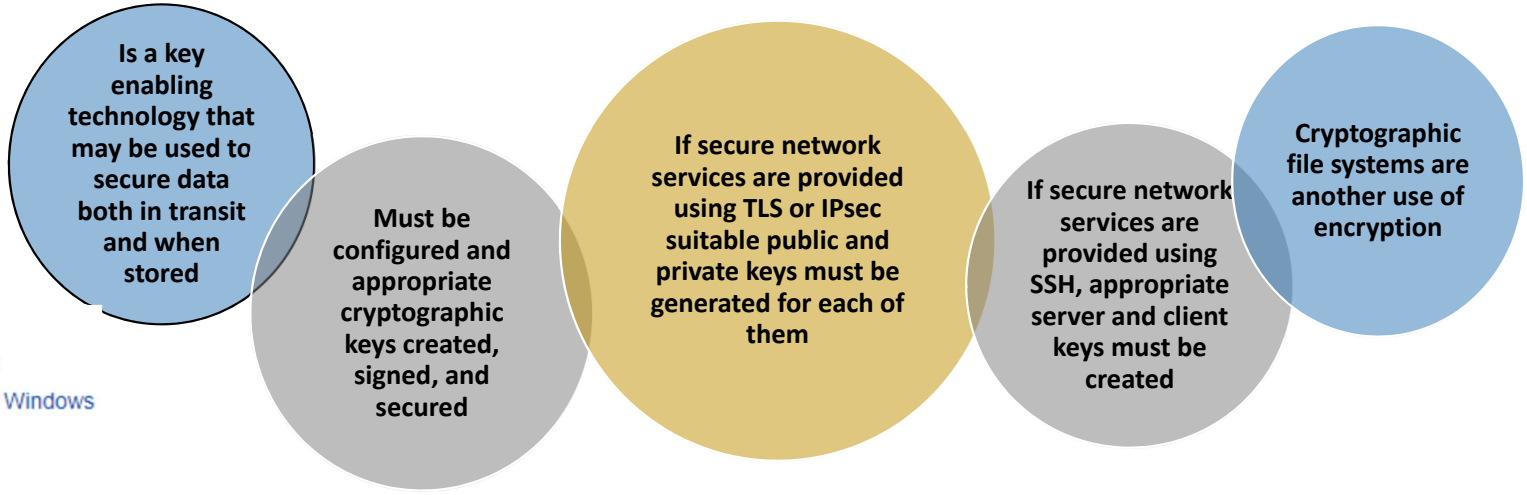
- Checklists are included in security hardening guides
- There are programs specifically designed to:
  - Review a system to ensure that a system meets the basic security requirements
  - Scan for known vulnerabilities and poor configuration practices
- Should be done following the initial hardening of the system
- Repeated periodically as part of the security maintenance process

Application Configuration

- Security policy can control app execution
  - Whitelisting and blacklisting
  - Whitelisting nothing run unless it's approved (very restrictive)
  - Blacklisting everything run except those in the blacklist
  - This can be achieved using app hash of certificates (trusted publishers)

- Of particular concern with remotely accessed services such as Web and file transfer services
  - Risk from this form of attack is reduced by ensuring that most of the files can only be read, but not written, by the server

Encryption Technology



- AdvFS on Digital Tru64 UNIX
- Novell Storage Services on Novell NetWare and Linux
- NTFS with Encrypting File System (EFS) for Microsoft Windows
- ZFS since Pool Version 30
- Ext4, added in Linux kernel 4.1[1] on June 2015
- F2FS, added in Linux 4.2[2]
- APFS, macOS High Sierra (10.13) and later.

Security Maintenance

- Process of maintaining security is continuous
- Security maintenance includes:
  - Monitoring and analyzing logging information
  - Performing regular backups
  - Recovering from security compromises
  - Regularly testing system security
  - Using appropriate software maintenance processes to patch and update all critical software, and to monitor and revise configuration as needed

Security Maintenance

- Process of maintaining security is continuous
- Example secure configuration policies:
  - Stay updated with the latest patches
  - Compromised systems are re-imaged (not cleaned)

Logging

Can only inform you about bad things that have already happened

In the event of a system breach or failure, system administrators can more quickly identify what happened

Key is to ensure you capture the correct data and then appropriately monitor and analyze this data

Information can be generated by the system, network and applications

Range of data acquired should be determined during the system planning stage

Generates significant volumes of information and it is important that sufficient space is allocated for them

Automated analysis is preferred

Data Backup and Archive

Performing regular backups of data is a critical control that assists with maintaining the integrity of the system and user data

- May be legal or operational requirements for the retention of data

Backup

- The process of making copies of data at regular intervals

Archive

- The process of retaining copies of data over extended periods of time in order to meet legal and operational requirements to access past data

Needs and policy relating to backup and archive should be determined during the system planning stage

- Kept online or offline
- Stored locally or transported to a remote site
  - Trade-offs include ease of implementation and cost versus greater security and robustness against different threats

Video Summary

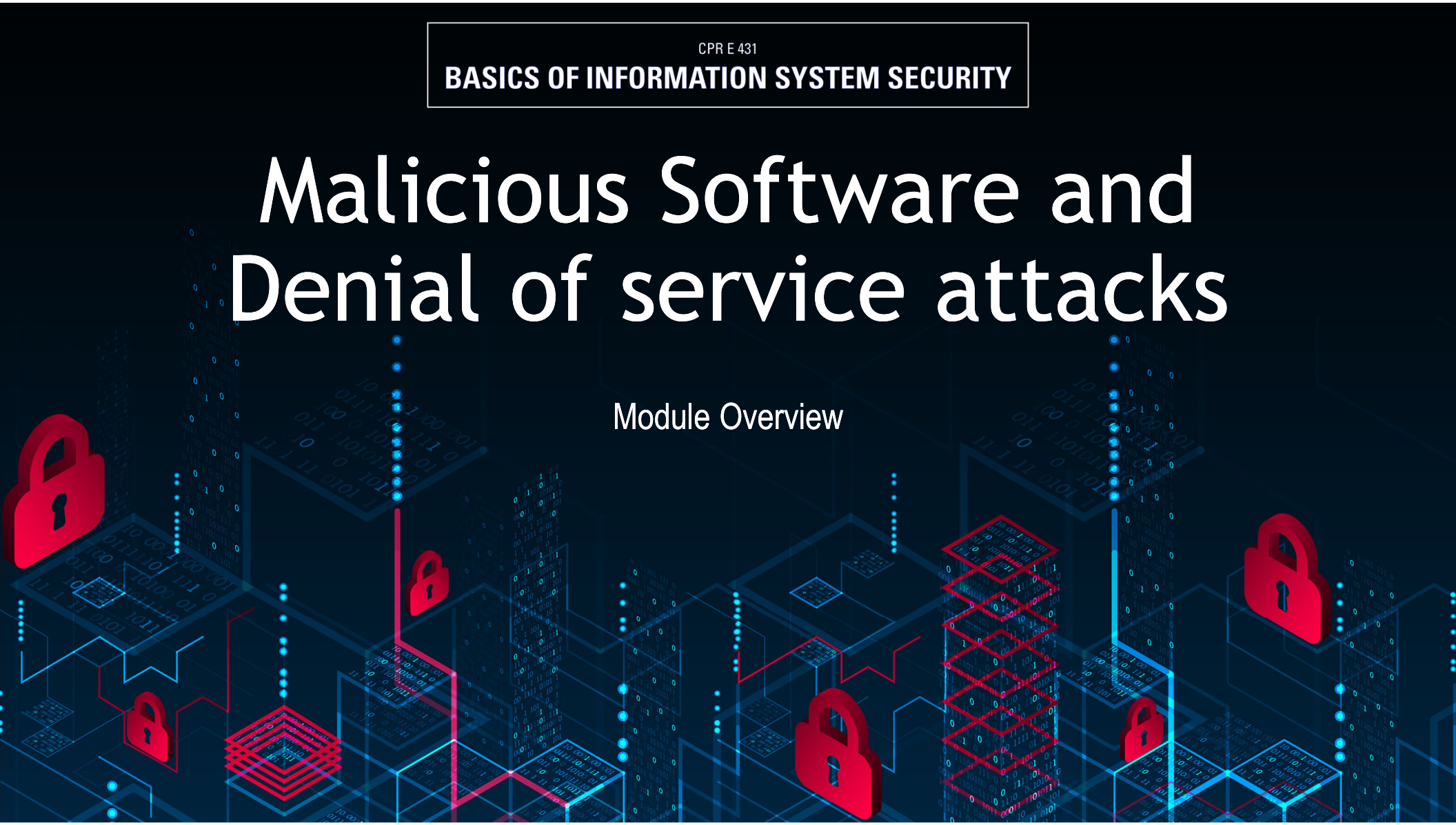
- OS Initial Setup and Patching
- OS Application Configuration
- Encryption Technology
- Security Maintenance
- Logging and Backup

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

Malicious Software and Denial of service attacks

Module Overview



Module Objectives:

- Introduction to Cryptography Tools
  - By the end of this module you will be able to:
    - ✓ Define Malware and its different types
    - ✓ Define denial of service attack
    - ✓ Understand the different concepts of generating denial of service attack
    - ✓ Implement ICMP/ping requests and learn how it is used in Distributed Denial of Service attack

Introduction to Information Security

What is a Malicious Software

Viruses

Worms and Social Engineering

Malware Countermeasures

What is a Denial of Service Attack

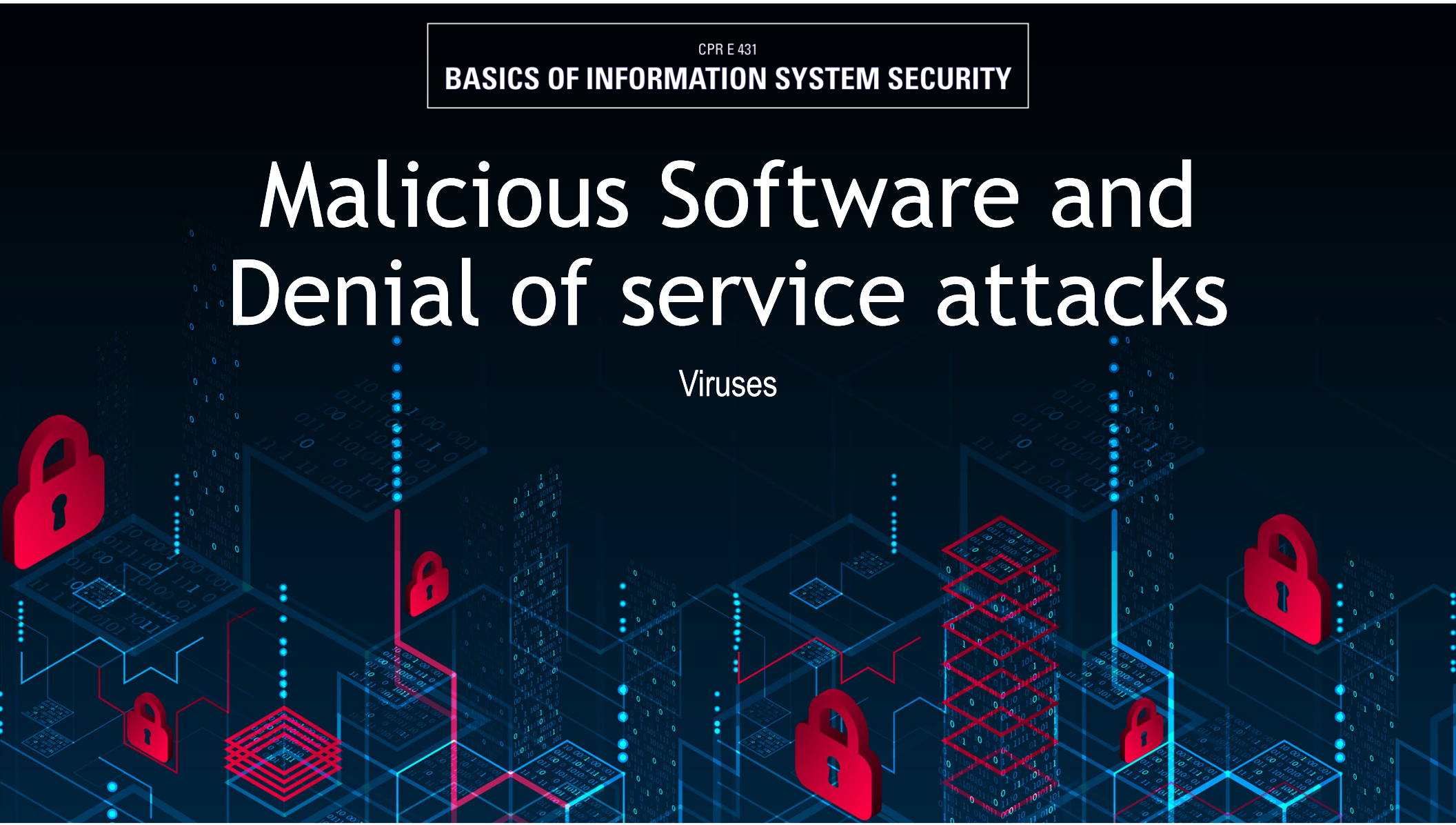
Distributed Denial of Service Attack

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

Malicious Software and Denial of service attacks

Viruses



Video Summary

- What is a Virus?
- Nature of Viruses
- Compression Virus
- Virus Classifications
- Macro-virus (Melissa)

Viruses

- Piece of software that infects programs
  - Modifies them to include a copy of the virus
  - Replicates and goes on to infect other content
  - Easily spread through network environments
- When attached to an executable program a virus can do anything that the program is permitted to do
  - Executes secretly when the host program is run
- Specific to operating system and hardware
  - Takes advantage of their details and weaknesses

Nature of Viruses

- A virus is piece of software that "infects" programs and copies itself to other programs
- The phases of a virus are:

Nature of Viruses

- A virus is piece of software that "infects" programs and copies itself to other programs
- The phases of a virus are:
  1. Dormant: virus is idle; will be activated by some event (like logic bomb)

Nature of Viruses

- A virus is piece of software that "infects" programs and copies itself to other programs
- The phases of a virus are:
  1. Dormant: virus is idle; will be activated by some event (like logic bomb)
  2. Propagation: virus copies itself into other programs or areas of operating system

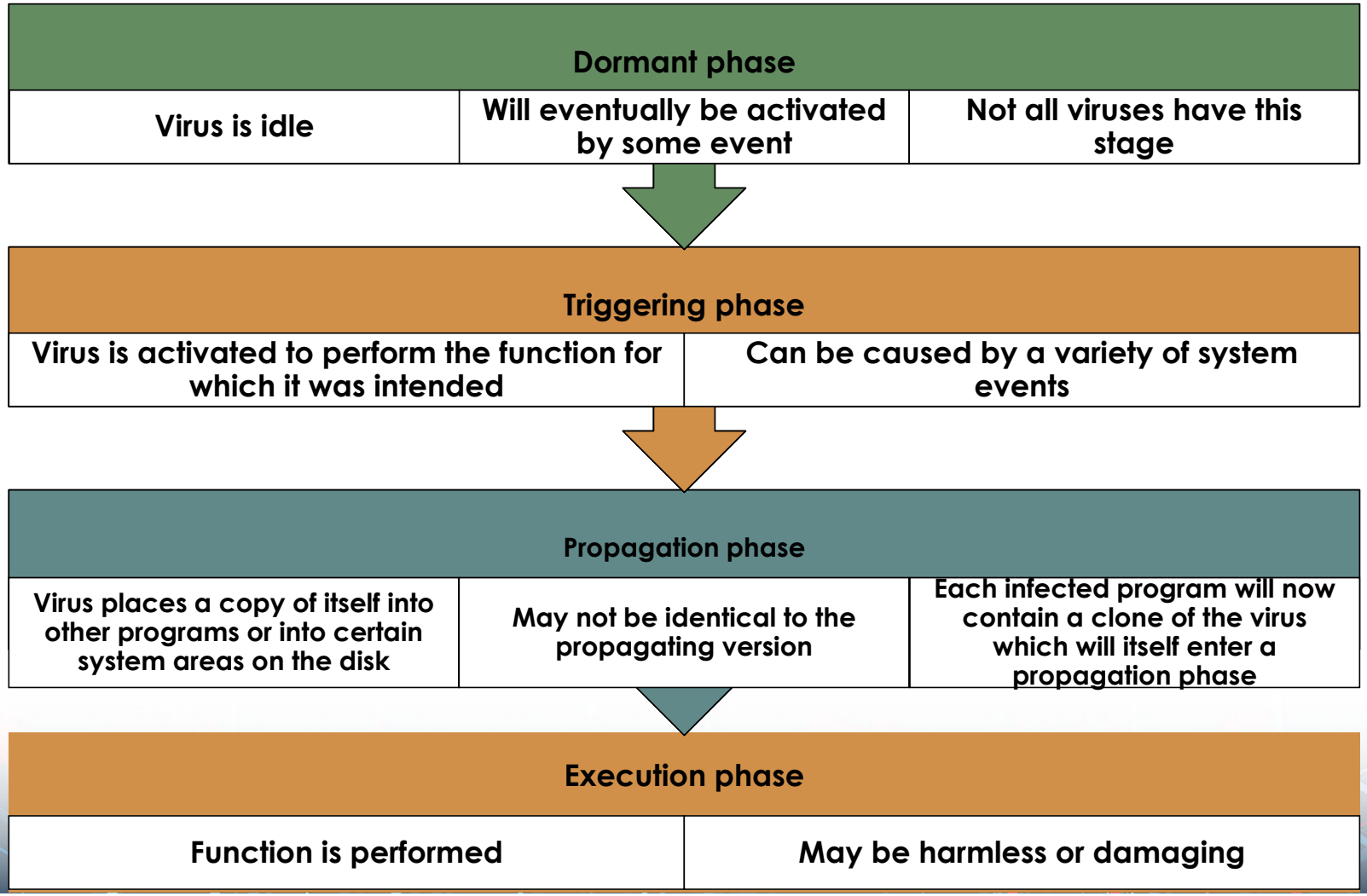
Nature of Viruses

- A virus is piece of software that "infects" programs and copies itself to other programs
- The phases of a virus are:
  1. Dormant: virus is idle; will be activated by some event (like logic bomb)
  2. Propagation: virus copies itself into other programs or areas of operating system
  3. Triggering: virus is activated to perform some function; similar triggers to logic bombs, but also number of times virus copied

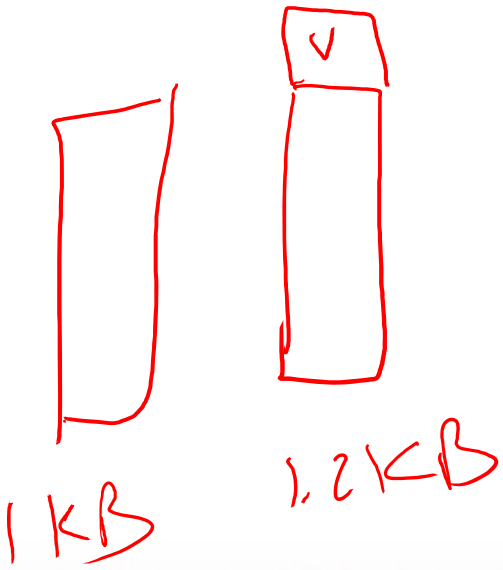
Nature of Viruses

- A virus is piece of software that "infects" programs and copies itself to other programs
- The phases of a virus are:
  1. Dormant: virus is idle; will be activated by some event (like logic bomb)
  2. Propagation: virus copies itself into other programs or areas of operating system
  3. Triggering: virus is activated to perform some function; similar triggers to logic bombs, but also number of times virus copied
  4. Execution: function is performed, either harmless (display a message) or malicious (delete or modify files)

Virus Phases



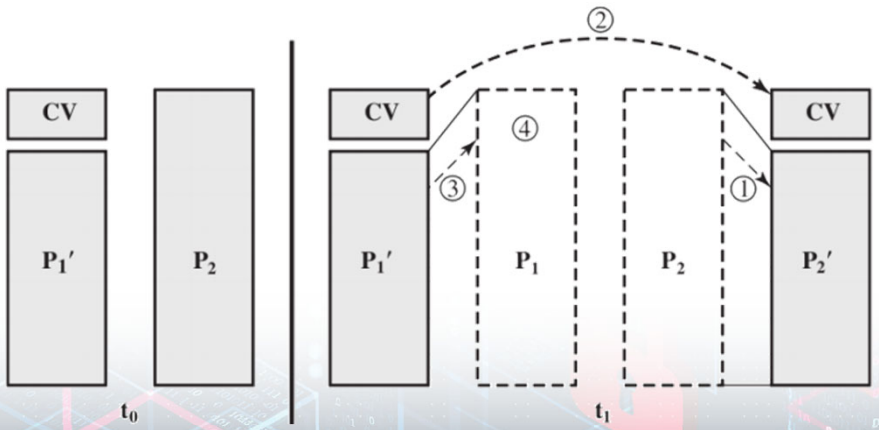
A Simple Virus

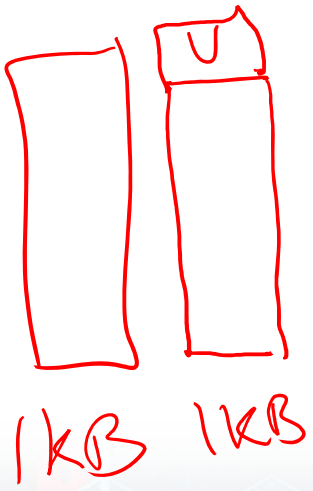



```
program V :=
{goto main;
   1234567;
   subroutine infect-executable :=
      {loop:
      file := get-random-executable-file;
      if (first-line-of-file = 1234567)
         then goto loop
      else
         prepend V to file; }
   subroutine do-damage :=
      {whatever damage is to be done}
   subroutine trigger-pulled :=
      {return true if some condition holds}
main: main-program :=
   {infect-executable;
   if trigger-pulled
      then do-damage;
   goto next;}
next:
}
```

Compression Virus

- The simple virus can be detected because file length is different from original program
- This detection can be avoided using compression
- Assume program P1 is infected with virus CV
  1. For each uninfected file P2, the virus compresses P2 to produce P2
  2. Virus CV is pre-pended to P2 (so resulting size is same as P2)
  3. P1 is uncompressed and (4) executed





Compression Virus

```
program CV :=
{  goto main;
   01234567;
   subroutine infect-executable :=
      {loop:
         file := get-random-executable-file;
         if (first-line-of-file = 01234567)
            then goto loop;
      (1) compress file;
      (2) prepend CV to file;
   }
main: main-program :=
{  if ask-permission
      then infect-executable;
   (3) uncompress rest-of-file;
   (4) run uncompressed file;}
}
```

Virus Classifications

- Classification by target
  - Boot sector infector
    - Infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus
  - File infector
    - Infects files that the operating system or shell considers to be executable
  - Macro virus
    - Infects files with macro or scripting code that is interpreted by an application
  - Multipartite virus
    - Infects files in multiple ways
- Classification by concealment strategy
  - Encrypted virus
    - A portion of the virus creates a random encryption key and encrypts the remainder of the virus
  - Stealth virus
    - A form of virus explicitly designed to hide itself from detection by anti-virus software
  - Polymorphic virus
    - A virus that mutates with every infection
  - Metamorphic virus
    - A virus that mutates and rewrites itself completely at each iteration and may change behavior as well as appearance

Hard disk Firmware Virus

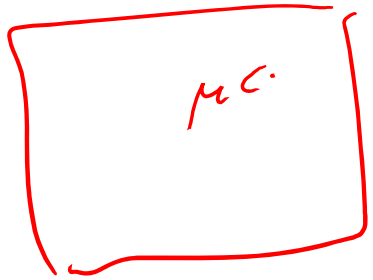
Indestructible malware by Equation cyberspies is out there – but don't panic (yet)

February 17, 2015 Serge Malenkovich Featured Post, News, Security No comments

Kaspersky's GReAT team just published research on the Equation cyber-espionage group's activity, and it revealed quite a few technical marvels. This old and powerful hacker group has produced a very complex series of malicious "implants", but the most interesting finding is the malware's ability to reprogram the victim's hard drives, making their "implants" invisible and almost indestructible.

> *Suite of Sophisticated Nation-State Attack Tools Found With Connection to Stuxnet – http://t.co/FsaH0Jzq5O*
> *— Kim Zetter (@KimZetter) February 16, 2015*

This is one of the long-anticipated scary stories in computer security – an incurable virus that persists in computer hardware forever was considered an urban legend for decades, but it seems people spend millions of dollars to make it happen. Some press reports on Equation's story go as far as saying this enables hackers "to eavesdrop on the majority of



Macro and Scripting Viruses

- NIST-IR 7298 defines a macro virus as:

  "a virus that attaches itself to documents and uses the macro programming capabilities of the document's application to execute and propagate"

- Are threatening for a number of reasons:
  - Is platform independent
  - Infect documents, not executable portions of code
  - Are easily spread
  - Are much easier to write or to modify than traditional executable viruses

Macro-Virus (Melissa)

- Virus released by David Smith in 26 March 1999
- Posted a message to a newsgroup containing an MS Word attachment (macro-virus)
- Estimated damage up to $1000 Million
- Designed to infect computers with Word 97/2000

Macro-Virus (Melissa)

- Virus sent as attachment to email:
  - Subject: “Important message from <user name>”
  - Body: “Here is that document you asked for… don’t show anyone else”

- When executed the macro automatically sent the email to 50 people in address book
  - Required MS Outlook to be running
  - Look like you receive an email from someone you know
  - It infects all other documents created on the computer

Macro-Virus (Melissa)

- Smith said: I had no idea that the virus would have this sort of impact and inflict this kind of damage. It was intended to be nothing more than a harmless joke."

- Smith arrested in 1 April 1999. Mr. Smith went to the prison for 20 months, fined $5,000 and ordered, on release, to "not be involved with computer networks or internet unless authorized by the court".

Macro-Virus (Melissa)

```
'WORD/Melissa written by Kwyjibo
'Works in both Word 2000 and Word 97
'Worm? Macro Virus? Word 97 Virus? Word 2000 Virus? You Decide!
'Word -> Email | Word 97 <--> Word 2000 ... it's a new age!

If Day(Now) = Minute(Now) Then Selection.TypeText " Twenty-two points,
plus triple-word-score, plus fifty points for using all my letters.
Game's over. I'm outta here."
```

Video Summary

- What is a Virus?
- Nature of Viruses
- Compression Virus
- Virus Classifications
- Macro-virus (Melissa)

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

Malicious Software and Denial of service attacks

Worms and Social Engineering

Video Summary

- What are worms?
- Examples for well-know worms
- Worm target discovery
- Social Engineering
- Social Engineering Case Study

Worms

- Program that actively seeks out more machines to infect and each infected machine serves as an automated launching pad for attacks on other machines
- Exploits software vulnerabilities in client or server programs
- Can use network connections to spread from system to system
- Spreads through shared media (USB drives, CD, DVD data disks)
- E-mail worms spread in macro or script code included in attachments
- Upon activation the worm may replicate and propagate again
- Usually carries some form of payload
- First known implementation was done in Xerox Palo Alto Labs in the early 1980s

Worm Target Discovery

Scanning (or fingerprinting)

- First function in the propagation phase for a network worm
- Searches for other systems to infect

Random

- Each compromised host probes random addresses in the IP address space using a different seed

Hit-list

- The attacker first compiles a long list of potential vulnerable machines
- Once the list is compiled the attacker begins infecting machines on the list
- Each infected machine is provided with a portion of the list to scan
- This results in a very short scanning period which may make it difficult to detect that infection is taking place

Topological

- This method uses information contained on an infected victim machine to find more hosts to scan

Local subnet

- If a host can be infected behind a firewall that host then looks for targets in its own local network
- The host uses the subnet address structure to find other hosts that would otherwise be protected by the firewall

Worm Attacks

Melissa	1998	E-mail worm First to include virus, worm and Trojan in one package
Code Red	July 2001	Exploited Microsoft Internet Information Services (IIS) bug Probes random IP addresses Consumes significant Internet capacity when active
Code Red II	August 2001	Also targeted Microsoft IIS Installs a backdoor for access
Nimda	September 2001	Had worm, virus and mobile code characteristics Spread using e-mail, Windows shares, Web servers, Web clients, backdoors
SQL Slammer	Early 2003	Exploited a buffer overflow vulnerability in SQL server compact and spread rapidly
Sobig.F	Late 2003	Exploited open proxy servers to turn infected machines into spam engines
Mydoom	2004	Mass-mailing e-mail worm Installed a backdoor in infected machines
Warezov	2006	Creates executables in system directories Sends itself as an e-mail attachment Can disable security related products
Conficker (Downadup)	November 2008	Exploits a Windows buffer overflow vulnerability Most widespread infection since SQL Slammer
Stuxnet	2010	Restricted rate of spread to reduce chance of detection Targeted industrial control systems

Code Red

- July 16 2001
- Worm aimed at Microsoft Internet Information Server (IIS) web servers (not users)
- Sent to web server as HTTP GET request
  - Bug in IIS allows the code to be stored by the server
  - Worm was stored in RAM; a reboot deleted the worm
- Worm had several states:
  - On first 19 days of month, send HTTP GET requests to random IP addresses, with the intention of infecting other web servers
  - On days 20 to 28, creates a DoS attack on www.whitehourse.gov
  - Dormant for the remaining of the month
- Infected 20,000 servers in 5 hours
- Consumed significant network resources (DoS attack)

Code Red II

- 4 August 2001
- Similar to CodeRed but also installed a Trojan horse on the web server
- Allowed anyone with web browser to send commands to web server:
  - Eg: delete or modify files on server

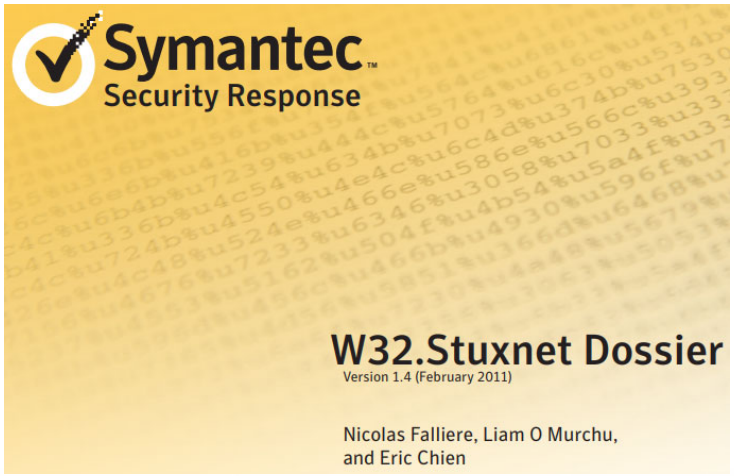
Stuxnet Worm

- **Stuxnet** is a malicious computer worm, first uncovered in 2010, thought to have been in development since at least 2005

- Stuxnet targets PLC and supervisory control and data acquisition (SCADA) systems and is believed to be responsible for causing substantial damage to the nuclear program of Iran

Stuxnet Worm

- Stuxnet functions by targeting machines using the Microsoft Windows operating system and networks, then seeking out Siemens Step7 software. Stuxnet reportedly compromised Iranian PLCs, collecting information on industrial systems and causing the fast-spinning centrifuges to tear themselves apart



Mobile Phone Worms

- First discovery was Cabir worm in 2004
- Then Lasco and CommWarrior in 2005
- Can completely disable the phone, delete data on the phone, or force the device to send costly messages
- CommWarrior replicates by means of Bluetooth to other phones, sends itself as an MMS file to contacts and as an auto reply to incoming text messages
- Recent attacks on Mobile phones: https://www.youtube.com/watch?time_continue=75&v=nZ_56MH_RV4&feature=emb_logo

Mobile Phone Worms

DroidKungFu

This piece of malware is unique in that it is able to avoid detection by antimalware software, according to the Wall Street Journal. It installs a backdoor in the Android OS that allows hackers to gain full control over a user's mobile device.

Mobile Phone Worms

DroidDream

This piece of malware, discovered in March 2011, has packaged itself inside legitimate applications in the official Android market that were released under developers "Kingmall2010," "we20090202," and "Myournet," according to MSNBC. The malware can then send user information to a remote server. A new variant of DroidDream – called DroidDreamLight – was discovered in May 2011.

Mobile Phone Worms

Android.Pjapps

Some Trojans can disguise themselves as legitimate applications. One example is Android.Pjapps, which hijacked the Steamy Windows app on Android, according to Symantec. The malicious app is similar to the legitimate one and even works – fogging up the screen – but it works in the background to send text messages to premium rate numbers, which in turn pays the creators of the Trojan.

What is Social Engineering

Social Engineer is someone who is a master of asking seemingly non-invasive or unimportant questions to gather information over time

- Gain trust
- Reduce defenses

Can be combined with a number of techniques to gather sensitive information

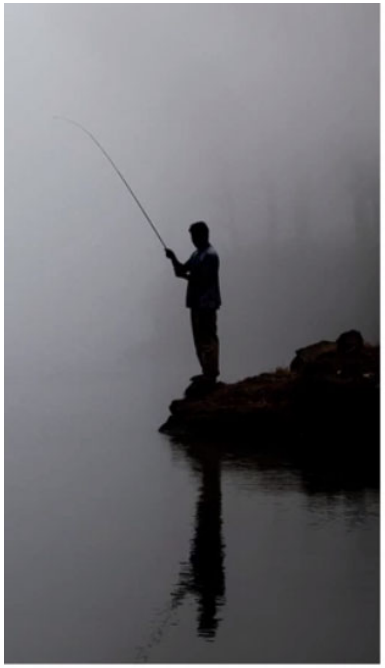
Social Engineering Attacks

Phishing
Attack via electronic communication (i.e. email) posing as someone trustworthy

Spear Phishing
Targeted attack appearing to come from a trusted source, often within the victim's own company, from someone in a position of authority

Whaling
Specific attack targeting high-profile business executives, upper management, etc.

Social Engineering Attacks (Fishing)



Very common

Impersonation (spoofing) of legitimate person or organization

Identity theft, credential theft

Delivery vehicles:

- E-mail
- Instant messaging
- Websites
- Phone calls

TrustedBank™

Dear valued customer of TrustedBank,

We have recieved notice that you have recently attempted to withdraw the following amount from your checking account while in another country: $135.25.

If this information is not correct, someone unknown may have access to your account. As a safety measure, please visit our website via the link below to verify your personal information:

http://www.trustedbank.com/general/custverifyinfo.asp

Once you have done this, our fraud department will work to resolve this discrepency. We are happy you have chosen us to do business with.

Thank you.

Social Engineering Attacks (Spear Fishing)

Highly targeted phishing attack

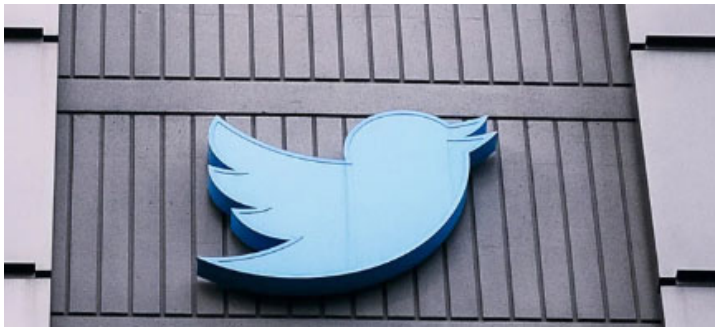
Limited distribution

More likely to use personalized information

May use impersonation of insiders

Users more likely to ascribe authenticity when they see personal details

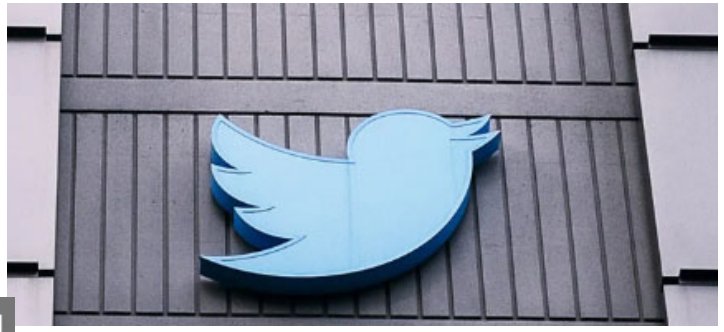
Social Engineering Case Study



- Summer 2020

- **AN** unprecedented Twitter hack saw the accounts of Elon Musk, Barack Obama, Joe Biden, Jeff Bezos, Bill Gates, Apple, Uber, and more fall into the hands of attackers who used that access to push a bitcoin scam?

Social Engineering Case Study



- Scammers received

around $100,000!!

2-3 hours

Video Summary

- What are worms?
- Examples for well-know worms
- Worm target discovery
- Social Engineering
- Social Engineering Case Study

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

Malicious Software and Denial of service attacks

Other Malwares and Countermeasures

Video Summary

- Other types of Malwares
- Zombies and Bots
- Information Theft
- System Corruptions
- Countermeasures

Ransomware

Restricts user access to data and/or programs

"crypto ransomware"

"locker ransomware"

May (or may not) decrypt files once ransom is paid, often in cryptocurrency

Early occurrence: PC CYBORG (1989)

WannaCry

SMB: Server Message Block

Rootkit

Malware with stealth features ("cloaking")

Hard to detect and remove

Can run before OS loads

Can have privileged ("root") access

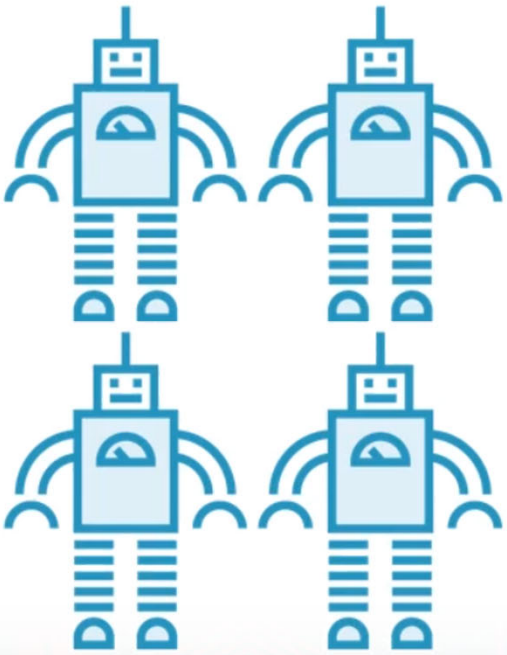
Early occurrences: NTRootkit (1999), Sony DRM (2005)

UEFI Secure Boot

Windows

- UEFI ensures boot loader is properly signed. If not, will not boot
- Helps prevent rootkits from replacing boot loader
- Can be turned off by anyone with admin access to UEFI settings

Botnet of Zombies



"Botnet" = Robot network

"Zombie" = computer under external control

Controlled by single entity

- Possibly for spam...
- ...or DDoS attacks

Typically includes stealth features

Early occurrence: Earthlink spammer (2000)

Zombies and Bots

- Uses:
  - ✓ Distributed DoS attacks
  - ✓ Spamming
  - ✓ Sniffing traffic
  - ✓ Keylogging
  - ✓ Spreading new malware
  - ✓ Installing advertisement add-ons and browser plugins

Information Theft

Keyloggers

- Captures keystrokes to allow attacker to monitor sensitive information
- Typically uses some form of filtering mechanism that only returns information close to keywords, e.g. "login", "password"

Spyware

- Subverts the compromised machine to allow monitoring of a wide range of activity on the system
- Monitoring history and content of browsing activity
- Redirecting certain Web page requests to fake sites
- Dynamically modifying data exchanged between the browser and certain Web sites of interest

May be legal (ex: ad-targeting)

System Corruption

- Action taken by malware on system: corrupt the system

- Data Destruction:
  - ✓ Delete data
  - ✓ Overwrite data
  - ✓ Encrypt data and then demand payment (ransomware)
- Real-World Damage:
  - ✓ Corrupt BIOS code so computer cannot boot
  - ✓ Control industrial systems to operate such that they fail (Stuxnet worm)
- Logic Bomb
  - ✓ Activate when certain conditions are met (date = time)

Countermeasures

Malware Countermeasure Approaches

- Prevention is an ideal solution, but almost impossible
  - ✓ Elements of prevention: policy, awareness, vulnerability mitigation, threat mitigation
  - ✓ Ensure systems are up-to-date (fix all bugs)
  - ✓ Apply Access Control (no permissions for malicious software to access files)
  - ✓ User awareness and training
- Detection, identification and removal (antivirus)
  - ✓ Generality
  - ✓ Timeliness
  - ✓ Resiliency (the antivirus can protect itself)
  - ✓ Global and local (your computer and the organization servers)
  - ✓ Transparent (antivirus shouldn't hide what it is doing)

User Education

- Know how to read e-mail headers
- Know how to check suspicious links
- Never provide credentials via email
- Never download unknown attachments

Generic Decryption

- A polymorphic virus must decrypt itself to activate
- Generic decryption runs executable code in virtual machine, monitors instructions
  - ✓ CPU emulator: virtual machine software
  - ✓ Virus signature scanner: scans for signatures
  - ✓ Emulation control module: monitors and controls execution of target code
- If decryption performed, malware is exposed and detected
- How long to run each anti-virus scan?
  - ✓ Too long: system performance degraded
  - ✓ Too short: do not see most of malware

Development of Anti-virus Software

- 1st generation: simple scanners
  - ✓ Requires a malware signature to identify the malware
  - ✓ Limited to the detection of known malware
- 2nd generation: heuristic scanners
  - ✓ Uses heuristic rules to search for probable malware instances
  - ✓ Another approach is integrity checking
- 3rd generation: activity traps
  - ✓ Memory-resident programs that identify malware by its actions rather than its structure in an infected program
- 4th generation: full featured protection
  - ✓ Packages consisting of a variety of anti-virus techniques used in conjunction
  - ✓ Include scanning and activity trap components and access control capability

Realtime Monitoring Softwares

This software wants to access “certain feature” do you want to allow it?

- Integrates with OS, monitors program behavior in real-time
- Block potentially malicious actions before they affect system
  - ✓ Attempts to open, view, delete, modify files
  - ✓ Attempts to format disks
  - ✓ Modifications to logic of executable files
  - ✓ Modification of critical system settings
  - ✓ Scripting of email to send executable files
  - ✓ Initiation of network connections

Antimalware Programs

Windows Defender comes with Windows 10

macOS does *not* include antimalware GUI

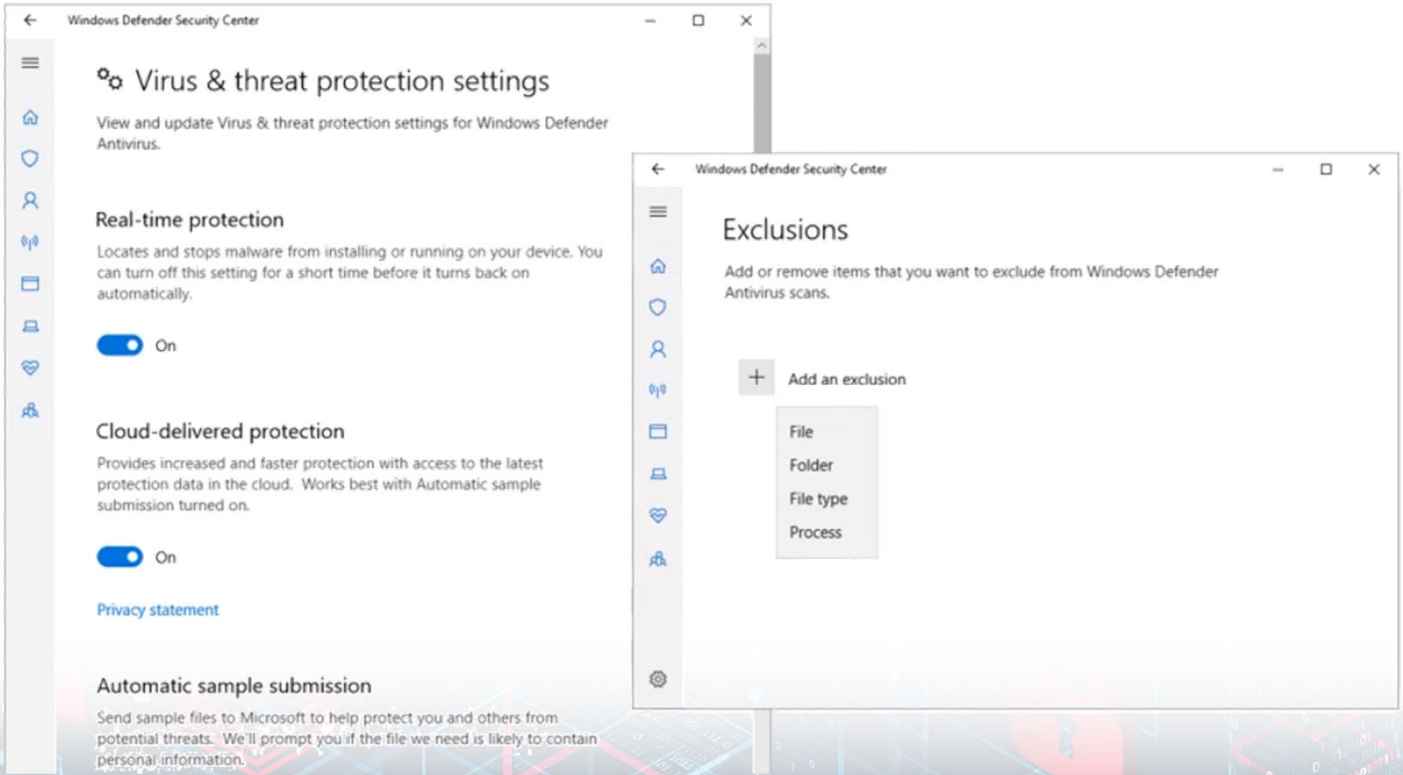
Features to look for:

- Real-time protection
- Ad-hoc scanning
- Periodic deep scans
- Offline scanning
- Light impact on OS
- Reputation databases

Windows Security

https://app.pluralsight.com/

Windows Defender Settings



https://app.pluralsight.com/

Video Summary

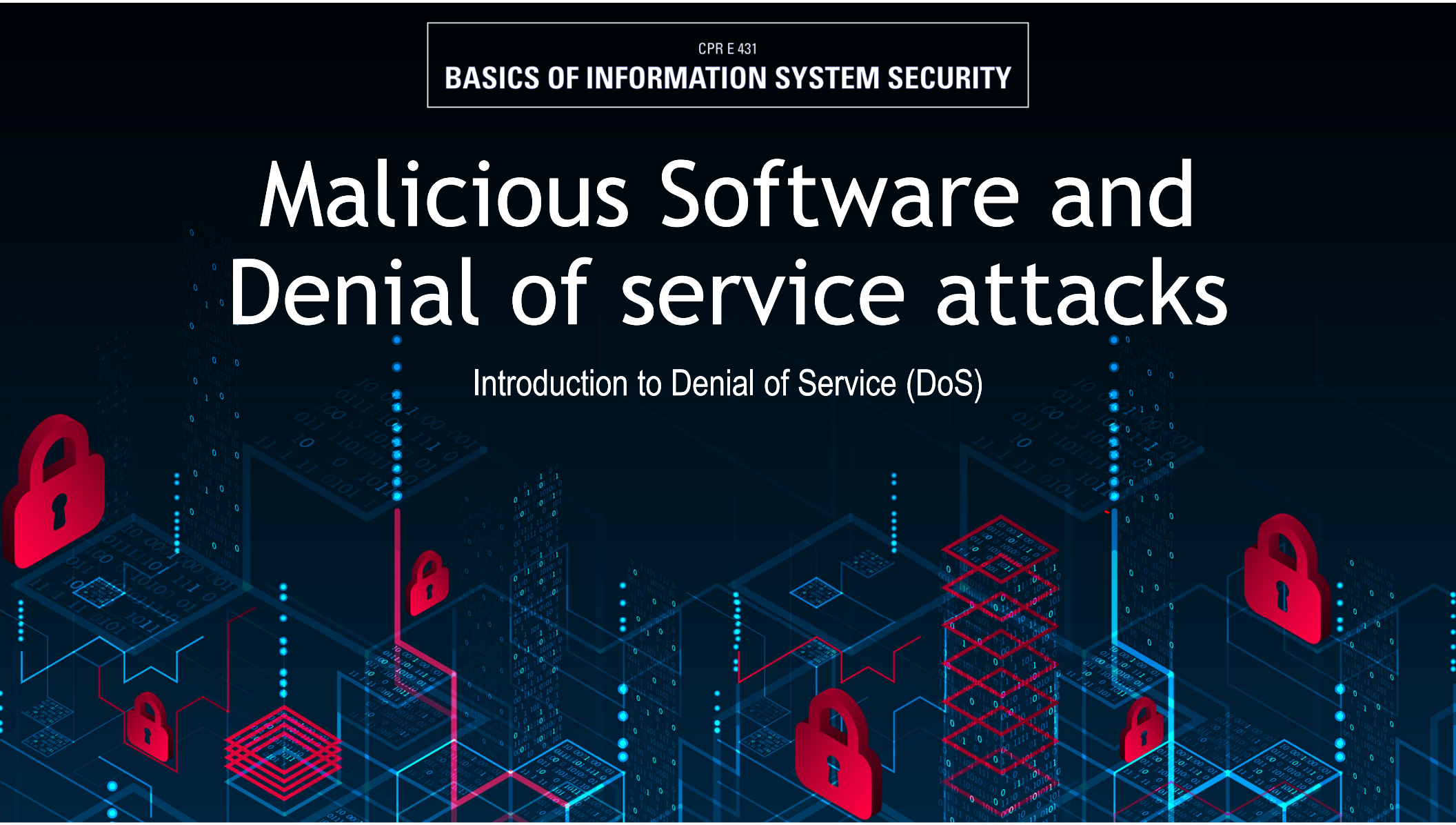
- Other types of Malwares
- Zombies and Bots
- Information Theft
- System Corruptions
- Countermeasures

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

Malicious Software and Denial of service attacks

Introduction to Denial of Service (DoS)



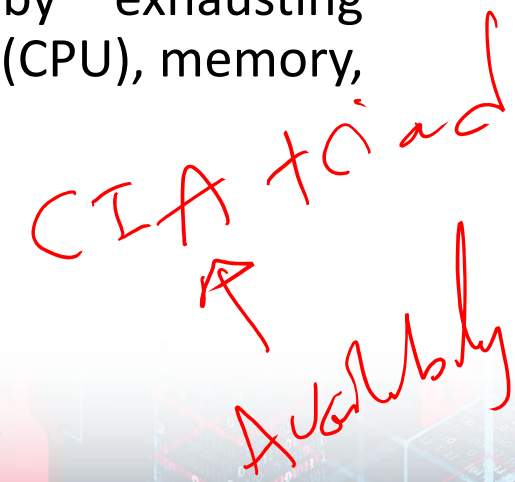
Video Summary

- What is DoS attack?
- Flooding Attacks
- Classic DoS (TCP SYN Flood)

Denial-of-Service (DoS) Attack

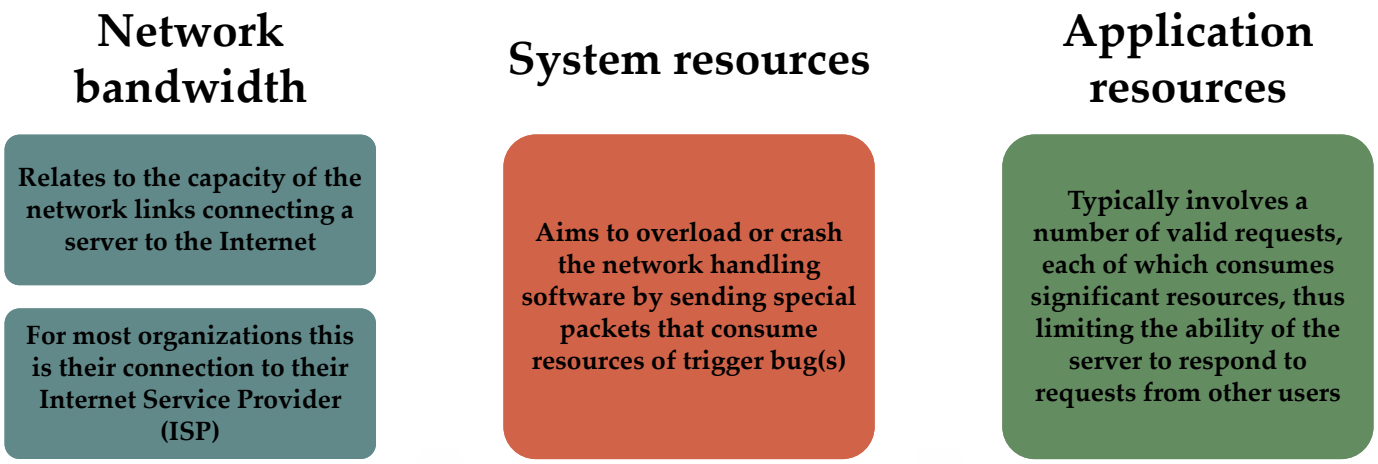
The NIST Computer Security Incident Handling Guide defines a DoS attack as:

"An action that prevents or impairs the authorized use of networks, systems, or applications by exhausting resources such as central processing units (CPU), memory, bandwidth, and disk space."



Denial-of-Service (DoS)

- A form of attack on the availability of some service
- Categories of resources that could be attacked are:



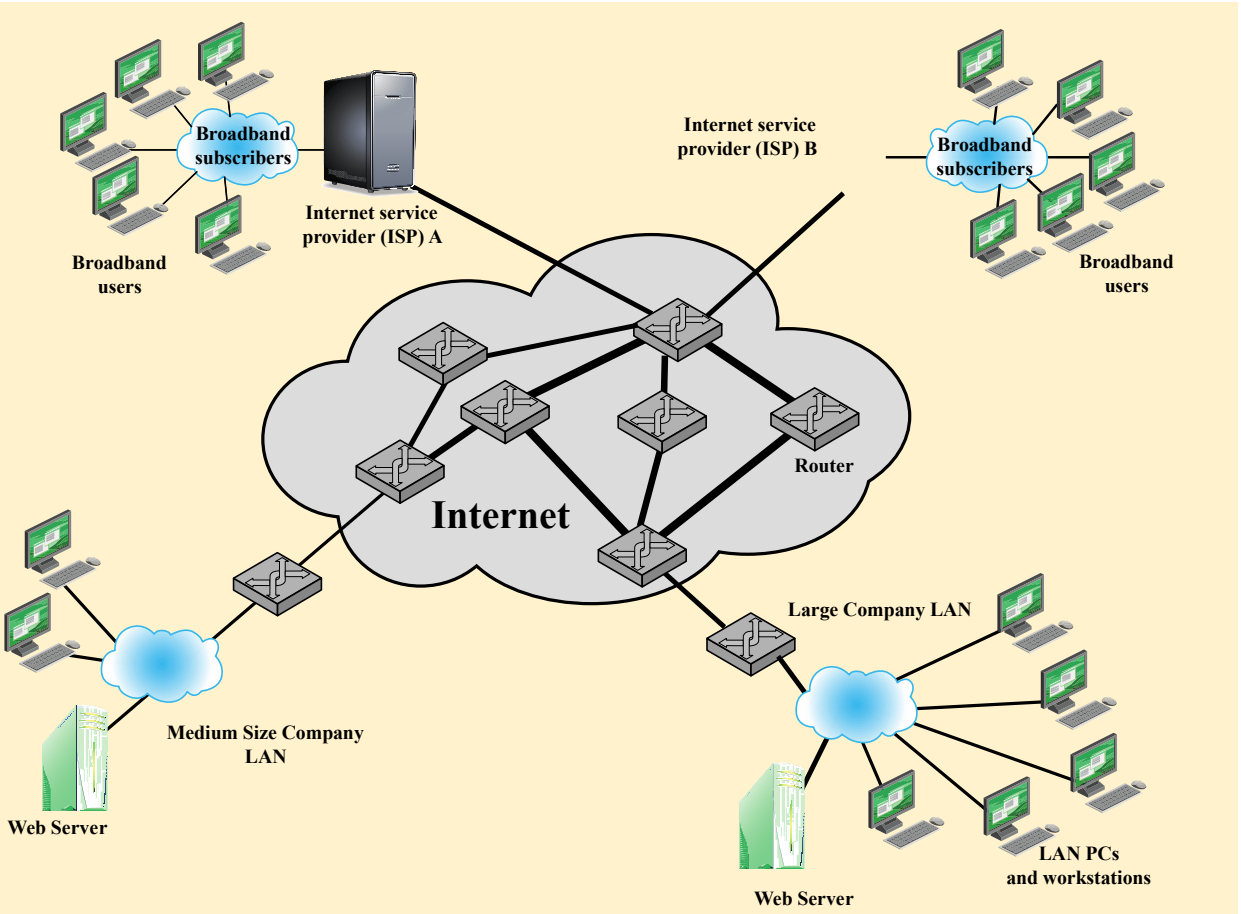


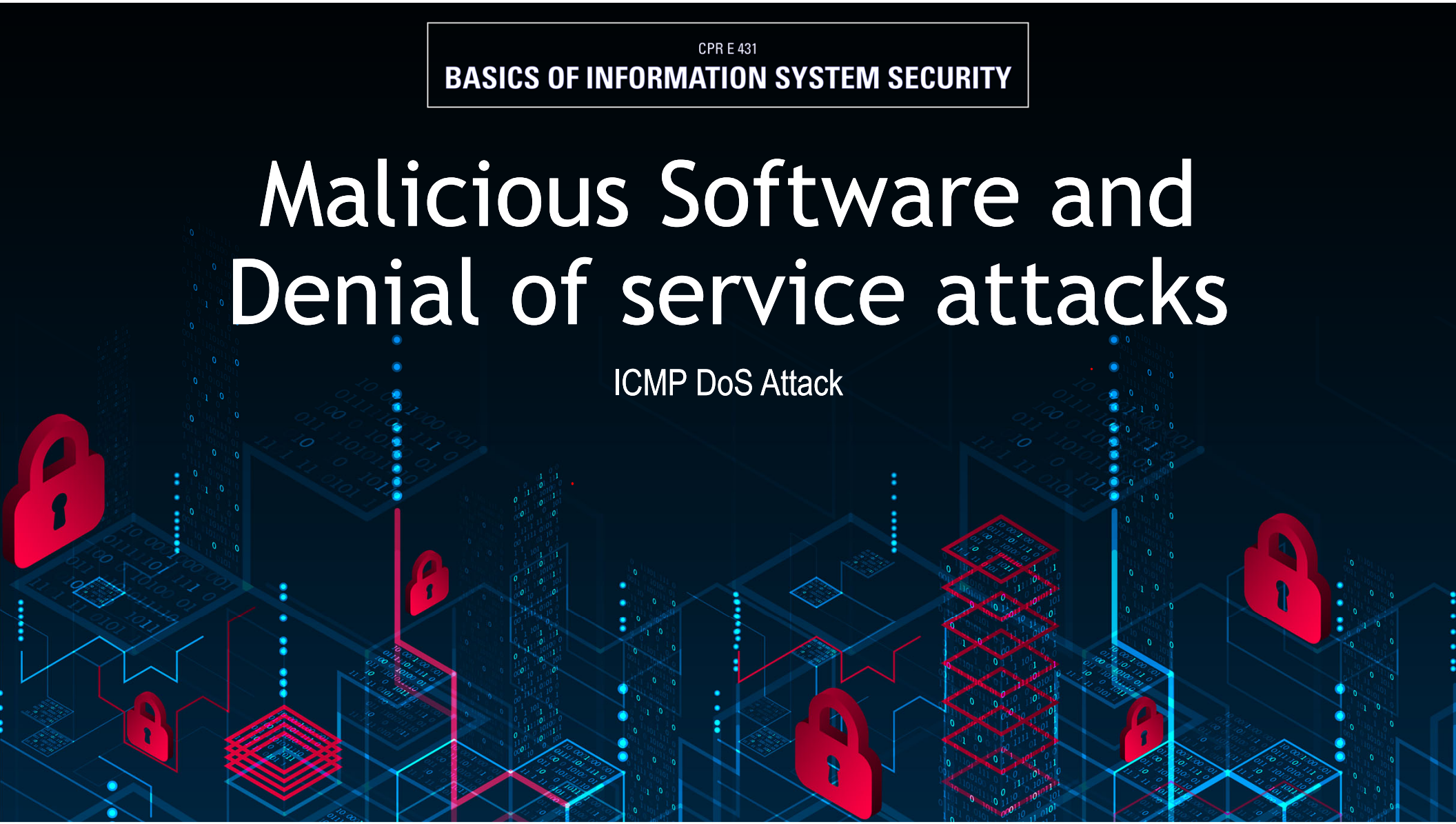
Figure 7.1 Example Network to Illustrate DoS Attacks

CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

Malicious Software and Denial of service attacks

ICMP DoS Attack



Video Summary

- Ping Test
- Simple Ping Flooding Attack
- Source Address Spoofing

Ping Test

```
$ ping -c 5 www.google.com
PING www.google.com (172.217.8.196): 56 data bytes
64 bytes from 172.217.8.196: icmp_seq=0 ttl=52 time=14.865 ms
64 bytes from 172.217.8.196: icmp_seq=1 ttl=52 time=14.943 ms
64 bytes from 172.217.8.196: icmp_seq=2 ttl=52 time=14.847 ms
64 bytes from 172.217.8.196: icmp_seq=3 ttl=52 time=14.970 ms
64 bytes from 172.217.8.196: icmp_seq=4 ttl=52 time=14.926 ms
--- www.google.com ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 14.847/14.910/14.970/0.047 ms
```

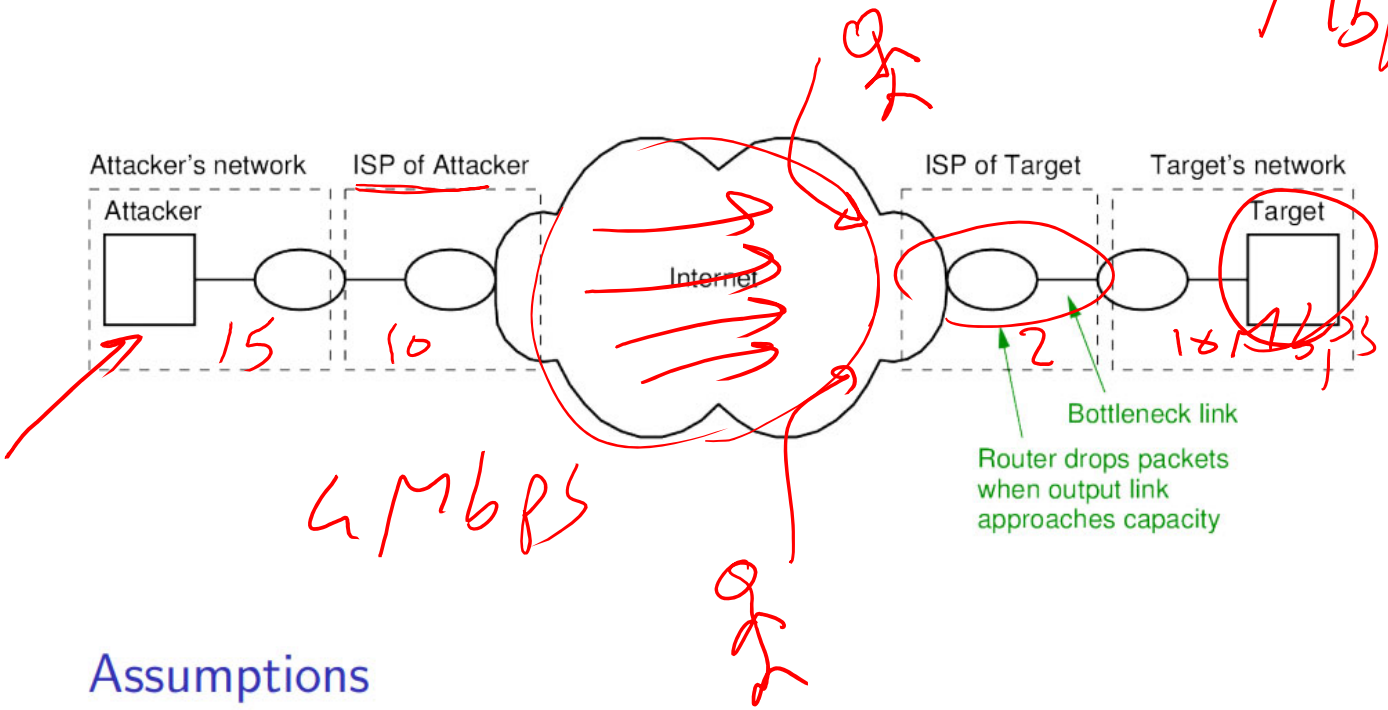
ICMP (Internet Control Message Protocol) is an error-reporting protocol network devices like routers use to generate error messages to the source IP address.

Ping uses the Internet Control Message Protocol (ICMP) to generate requests and handle responses.

ICMP is an error-reporting protocol network devices like routers use to generate error messages to the source IP address and also it is used to measure delay

Advantage: most computers will respond to the ping request

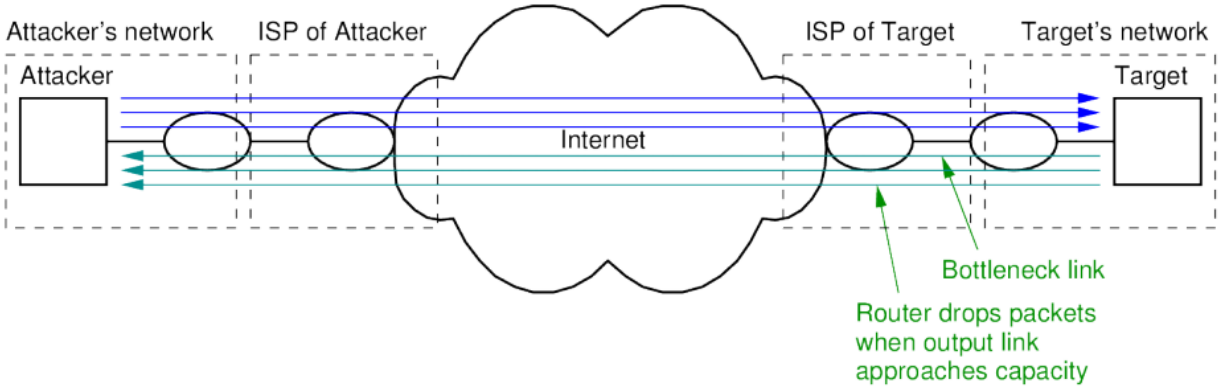
Simple Ping Flooding Attack



Assumptions

- Attacker has access to high capacity link
- Target's connection to Internet is lower capacity

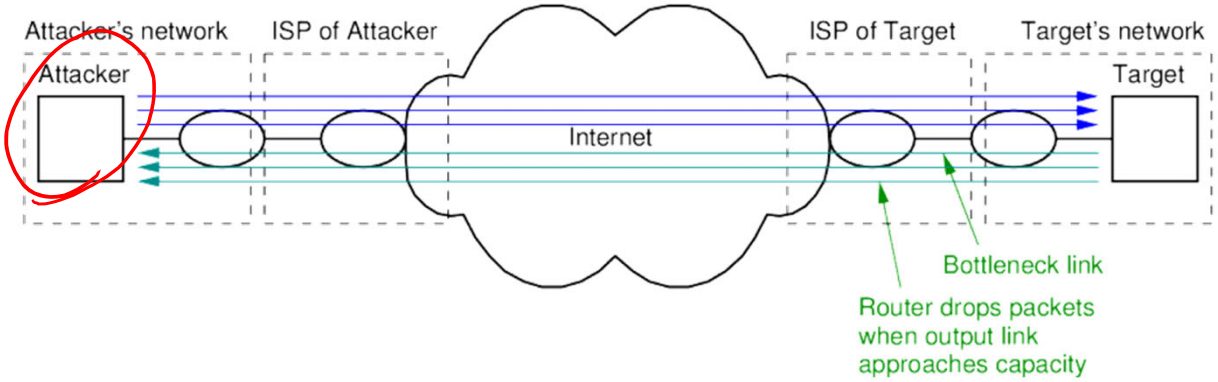
Simple Ping Flooding Attack



Attack

- Flood the server: Attacker uses `ping` to send many ICMP requests to target server
- Link from ISP to router is overloaded; router drops (valid) packets

Simple Ping Flooding Attack

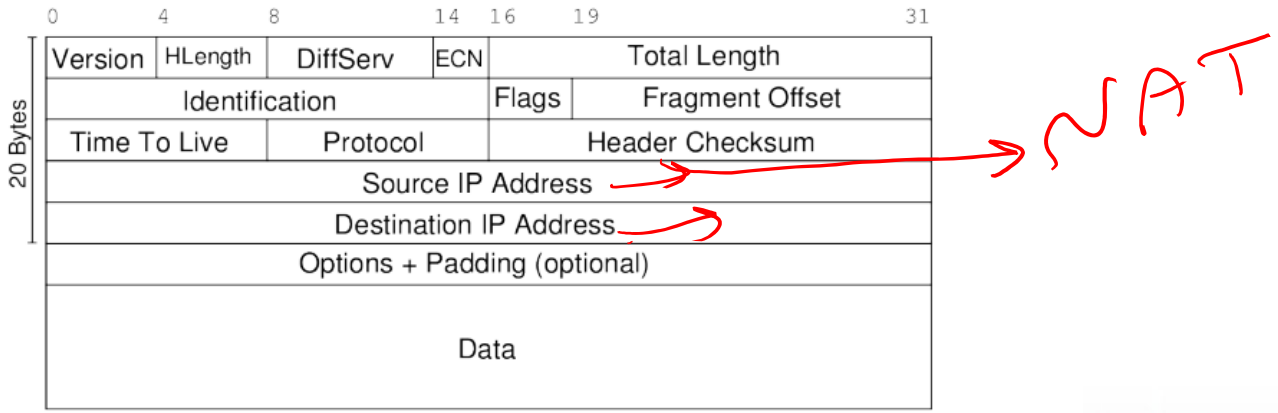


Countermeasures

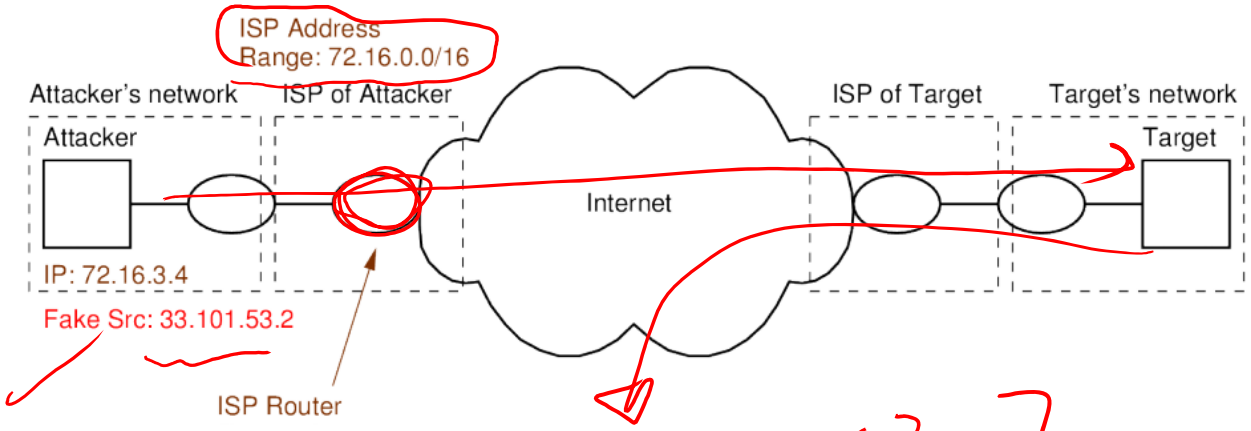
- ISPs block ping (ICMP) packets
- Target can identify the source: inform ISP, take legal action
- ICMP responses sent back to attacker, affecting their network performance

Source Address Spoofing

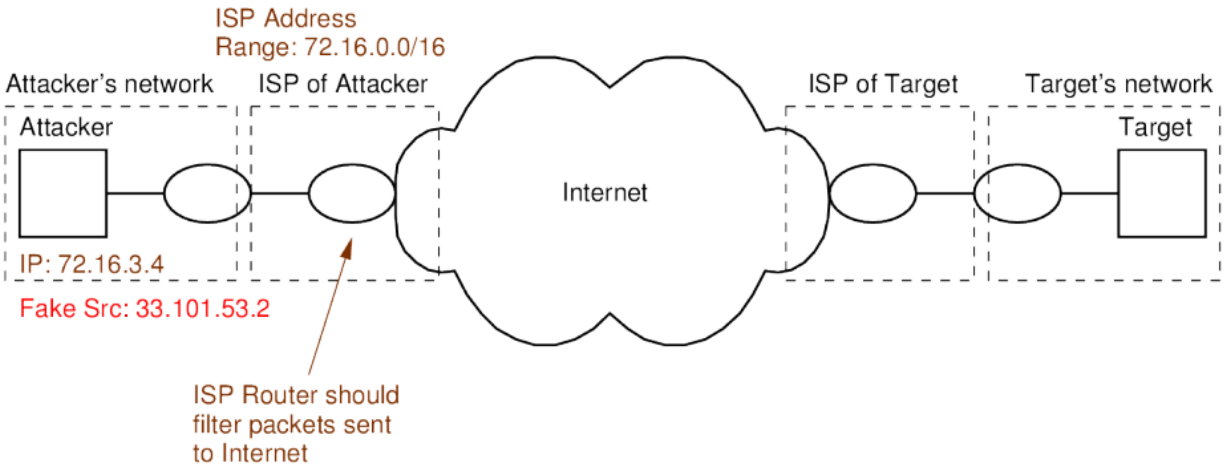
- Attacker sends packets with fake (or spoofed) source address
  - Target does not (immediately) know who performed attack
  - Responses are not sent to attacker
  - Source address may be of actual host or non-existent



Source Address Spoofing



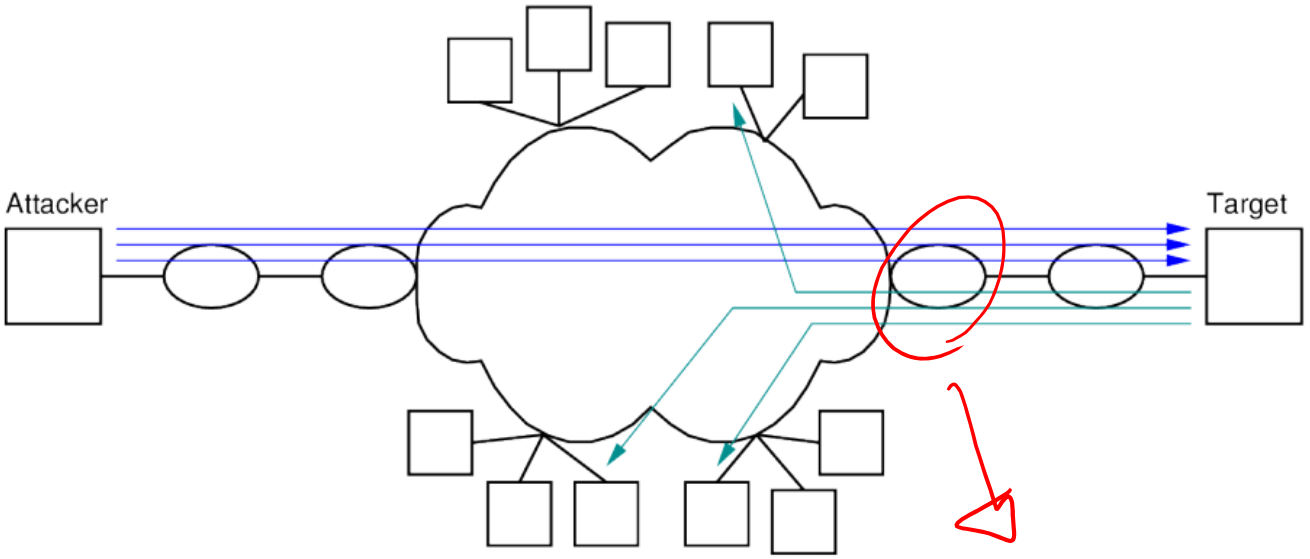
Source Address Spoofing



Countermeasure

- ISPs filter (drop) packets that come from invalid source address

Source Address Spoofing



Simple Ping Flooding Attack

- **How we can send pings at a rate that exceeds 1 Gbps to a certain webserver?**

Ping Request: by default it sends one packet per second and its size is 64 bytes (8 header and 56 payload)

1000 Bytes

$$\text{Pings/sec} = \frac{1 \times 10^9}{1000 \times 8} = 125000$$

bit/sec

$$-i \quad \frac{1}{125000}$$

Simple Ping Flooding Attack

- **How we can send pings at a rate that exceeds 1 Gbps to a certain webserver?**

Ping Request: by default it sends one packet per second and its size is 64 bytes (8 header and 56 payload)

Let's say we will change the size of the packet to be 1000 bytes

How many pings/sec to get 1 Gbps?

Pings/sec = 1000000000/(1000 x 8) = 125000

Note that one byte = 8 bits

Video Summary

- Ping Test
- Simple Ping Flooding Attack
- Source Address Spoofing