

UNIT 1

Overview of Computer Security

(4 Hours) — Information Security (CACS459)

Almost everything we do today — banking, communicating, shopping, studying — depends on computer systems and the networks that connect them. This convenience comes with risk: the same systems can be attacked, misused, or made to fail, sometimes with serious financial, legal, or even safety consequences. Computer Security is the field of study that deals with protecting computer systems and the information they hold against such risks. This unit lays the conceptual foundation for the entire course: what we are protecting, who might attack it, what requirements a secure system must satisfy, and what general principles guide good security design.

1.1 Computer Security Concepts

The U.S. National Institute of Standards and Technology (NIST) defines Computer Security as the protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (including hardware, software, firmware, information/data, and telecommunications). In plain language: computer security means keeping a system's hardware, software, and data safe from anyone or anything that should not have access to it, while still letting authorised people use it normally.

1.1.1 The CIA Triad

Three core objectives sit at the heart of almost every security goal, together known as the CIA Triad:

- **Confidentiality:** ensuring that information is not disclosed to unauthorised individuals, entities, or processes. It has two related aspects — data confidentiality (private data stays private) and privacy (individuals control what information about themselves is collected and shared).
- **Integrity:** ensuring that information and systems are changed only in an authorised manner. This includes data integrity (information is not altered without authorisation) and system integrity (a system performs its intended function in an unimpaired manner, free from unauthorised manipulation).
- **Availability:** ensuring that systems work promptly and that authorised users are not denied access to information or resources they are entitled to use.

The CIA Triad — Core Security Objectives

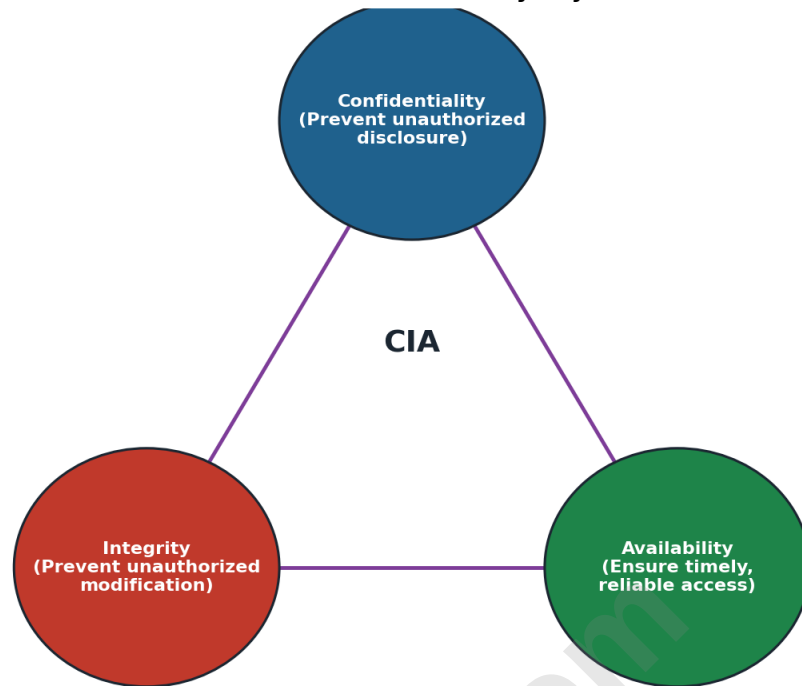


Fig. 1.1 — The CIA Triad: Confidentiality, Integrity, and Availability together form the foundation against which every security requirement is ultimately measured.

Many modern texts (including NIST) extend the classic CIA triad with two further objectives that are increasingly important in real systems:

- **Authenticity:** the property of being genuine and verifiable — a system must be able to verify that users, messages, and data really are who/what they claim to be.
- **Accountability:** the security goal that generates the requirement for actions of an entity to be traced uniquely back to that entity, supporting non-repudiation, deterrence, fault isolation, intrusion detection, and after-action recovery and legal action.

Note: A useful memory aid for examinations: CIA + AA — Confidentiality, Integrity, Availability, Authenticity, Accountability.

1.2 Computer Security, Information Security, and Network Security

These three terms are closely related and often used loosely, but they have slightly different scopes:

Term	Scope	Primary Focus
Computer Security	Protection of a single computer system (hardware, OS, applications, and the data stored on it)	Confidentiality, integrity, and availability of an individual system's resources

Term	Scope	Primary Focus
Information Security	Protection of information itself, in any form — electronic, printed, or spoken — wherever it exists or travels	Protecting the data/information asset, independent of the medium that carries it
Network Security	Protection of data as it moves across networks connecting multiple computer systems	Securing communication channels, protocols, and network infrastructure against interception, tampering, and disruption

In practice these three overlap heavily and are studied together, since a modern organisation's information rarely sits on just one machine — it is created, processed, stored, and transmitted across many systems and networks, all of which must be secured together for the information itself to be genuinely protected.

1.3 Threats, Attacks, and Assets

1.3.1 Basic Terminology

Term	Meaning
Asset	Anything of value to an organisation that needs protection: hardware, software, data, communication facilities/networks, and even an organisation's reputation
Vulnerability	A weakness in a system's design, implementation, or operation that could be exploited to violate the system's security policy
Threat	A potential for violation of security; a circumstance, capability, action, or event that could breach security and cause harm
Attack	An intelligent, deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system
Risk	The likelihood of a threat exploiting a vulnerability, combined with the impact if it succeeds

Relationship Between Threat Source, Threat, Attack, and Asset



Fig. 1.2 — A threat source exploits a vulnerability to carry out an attack, which then impacts an asset; risk grows wherever this whole chain is left unprotected.

1.3.2 Classification of Assets

- Hardware: computers, storage media, networking equipment — vulnerable to theft, physical damage, or unauthorised physical/remote access.
- Software: operating systems, utilities, and applications — vulnerable to deletion, modification, or theft (piracy), and to malicious alteration that introduces hidden flaws.
- Data: files and databases, as well as data in transit — vulnerable to unauthorised reading, unauthorised modification, or destruction.
- Communication facilities and networks: the wires, wireless links, switches, and protocols connecting systems — vulnerable to eavesdropping and disruption.

1.3.3 Classification of Attacks

Attacks are broadly classified into two categories, based on whether the attacker actively alters the system/data or merely observes it:

- Passive Attacks: the attacker only monitors or eavesdrops on data, without altering it. Examples include release of message contents (reading confidential data) and traffic analysis (observing communication patterns even when the content itself is encrypted). Passive attacks are hard to detect because they leave no visible trace, so the emphasis is on prevention (e.g., encryption) rather than detection.
- Active Attacks: the attacker attempts to alter system resources or affect their operation. Examples include masquerade (pretending to be a different entity), replay (capturing and later re-transmitting a valid data transmission), modification of messages (altering part of a legitimate message), and denial of service (DoS) (preventing or inhibiting the normal use of a resource). Active attacks are generally easier to detect, so the emphasis shifts to detection and recovery.

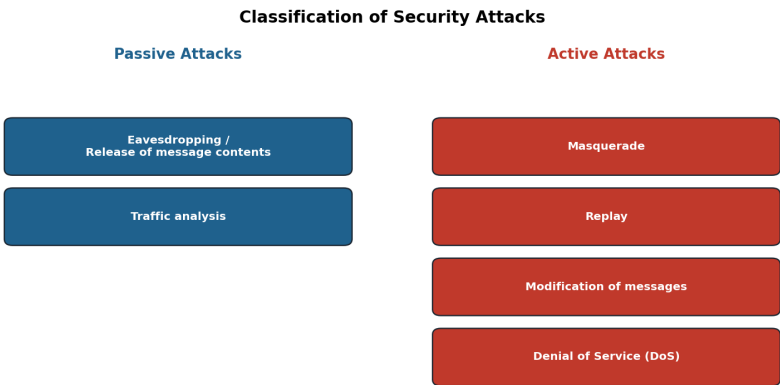


Fig. 1.3 — Security attacks fall into two broad families: passive attacks (focused on prevention) and active attacks (focused on detection and recovery).

Note: This classification, along with the security-services framework (authentication, access control, data confidentiality, data integrity, non-repudiation, availability), comes from the X.800 / RFC 4949 security architecture model widely referenced in this field.

1.4 Security Requirements

Building directly on the CIA Triad and its extensions, NIST's FIPS 200 standard identifies a broader set of security requirements that a secure information system should satisfy. These give a practical checklist that maps onto the five core concepts introduced earlier:

Requirement	Plain-English Description
Confidentiality	Preserving authorised restrictions on information access and disclosure, including protecting personal privacy and proprietary information
Integrity	Guarding against improper information modification or destruction, including ensuring non-repudiation and authenticity of information
Availability	Ensuring timely and reliable access to and use of information
Authenticity	Verifying that users are who they claim to be, and that each input arriving at the system came from a trusted source
Accountability	Tracing a security breach back to a responsible party uniquely, to support deterrence, fault isolation, and after-the-fact recovery

Note: Different applications weigh these requirements differently — a hospital records system may prioritise confidentiality and integrity above all else, while an emergency broadcast system may prioritise availability even at some cost to confidentiality.

1.5 Security Design Principles

Beyond knowing what to protect, security engineers also need general principles for how to design systems that are secure by construction, not as an afterthought. Many of these principles trace back to the classic 1975 paper by Saltzer and Schroeder, and remain the foundation of secure system design today.

Principle	Meaning
Economy of Mechanism	Keep the design of security mechanisms as simple and small as possible — simpler designs are easier to verify and less likely to contain flaws
Fail-Safe Defaults	Base access decisions on permission rather than exclusion; the default condition should be lack of access, and protection schemes should err on the side of denying access
Complete Mediation	Every single access to every object must be checked for proper authority — no caching of permission results that could become stale

Principle	Meaning
Open Design	The security of a mechanism should not depend on the secrecy of its design or implementation, only on the secrecy of keys or passwords (cf. Kerckhoffs's Principle)
Separation of Privilege	Where feasible, a protection mechanism that requires two independent keys/conditions to unlock is more robust than one that requires a single condition
Least Privilege	Every program and user should operate using the fewest privileges necessary to complete the job
Least Common Mechanism	Minimise the amount of mechanism shared between different users/components, since shared mechanisms are potential channels for unintended information flow
Psychological Acceptability	Security mechanisms should not make resources significantly harder to access than if the mechanism were absent — if it is too inconvenient, users will bypass it
Isolation	Keep public-access systems separate from critical resources; isolate security-critical code from the rest of the application
Encapsulation / Modularity	Group related security functionality into well-defined, independent modules with clear interfaces, simplifying testing and reasoning about security
Layering	Use multiple, overlapping protection approaches (defense in depth) so that if one layer fails, others still provide protection
Least Astonishment	A program or interface should always respond in the way that is least likely to astonish or surprise the user, reducing the chance of accidental security mistakes

Defense in Depth — Layering as a Security Design Principle

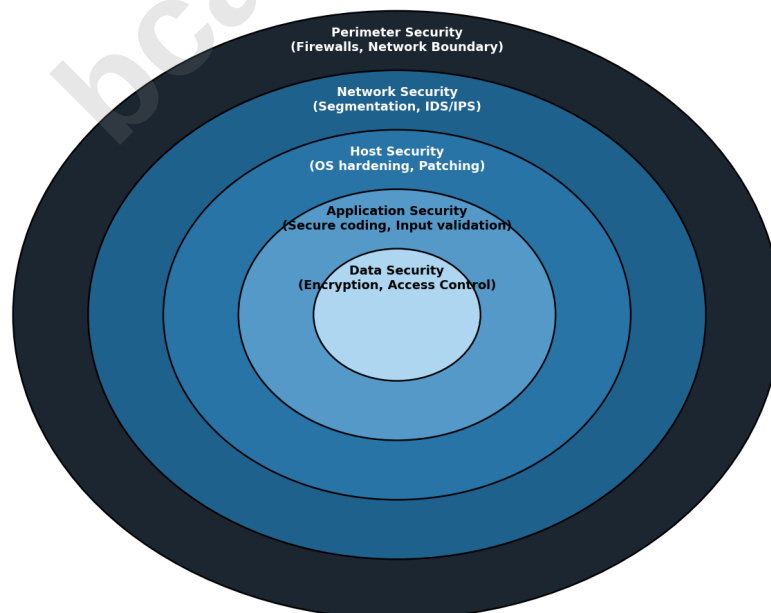


Fig. 1.4 — The Layering principle in practice: 'Defense in Depth' surrounds critical data with multiple independent rings of protection, so a single failure does not expose the whole system.

1.6 Attack Surfaces and Attack Trees

1.6.1 Attack Surface

The attack surface of a system is the sum of all the different points (the 'attack vectors') where an unauthorised user could try to enter data to or extract data from the system. A smaller, well-managed attack surface is generally easier to defend than a large, sprawling one. Attack surfaces are usually grouped into three categories:

- **Network Attack Surface:** vulnerabilities in network infrastructure, open ports, exposed services, and communication protocols, exploitable over a network connection.
- **Software Attack Surface:** vulnerabilities in application code, APIs, web interfaces, and any software that processes external input.
- **Human Attack Surface:** vulnerabilities arising from human behaviour — susceptibility to social engineering, weak passwords, insider threats, and simple human error.

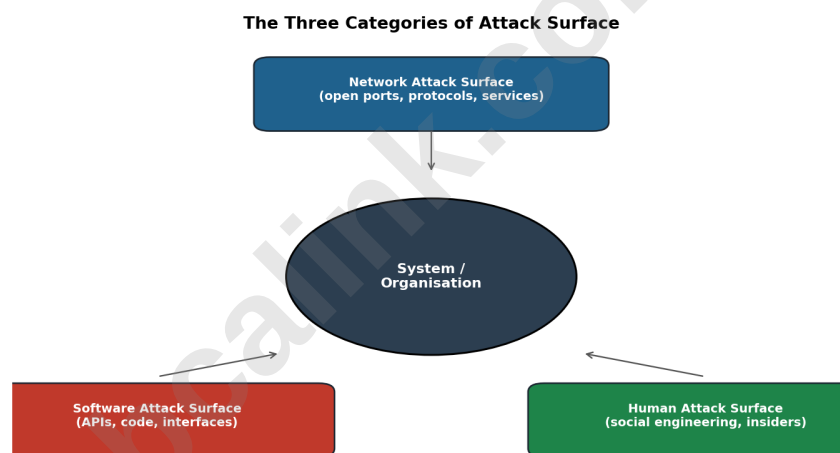


Fig. 1.5 — The three categories of attack surface that together determine how exposed a system or organisation is to attack.

1.6.2 Attack Trees

An attack tree is a structured, hierarchical diagram used to model the different ways an attacker could achieve a particular goal. The root node at the top represents the attacker's overall objective; each level below it breaks that objective down into smaller sub-goals (branches), and the leaf nodes at the bottom represent specific, concrete attack actions. Attack trees help security analysts systematically identify every realistic path to a compromise, and can be annotated with cost, likelihood, or required skill level for each leaf to help prioritise defensive effort.

Example 1.1 — Constructing an Attack Tree

Consider an attacker whose goal is to steal a victim's online banking credentials. The goal can be broken down into three broad approaches: tricking the user directly (phishing), compromising the user's device (malware), or attacking the bank's server infrastructure directly. Each of these, in turn, breaks down further into specific concrete techniques, as shown in the diagram below.

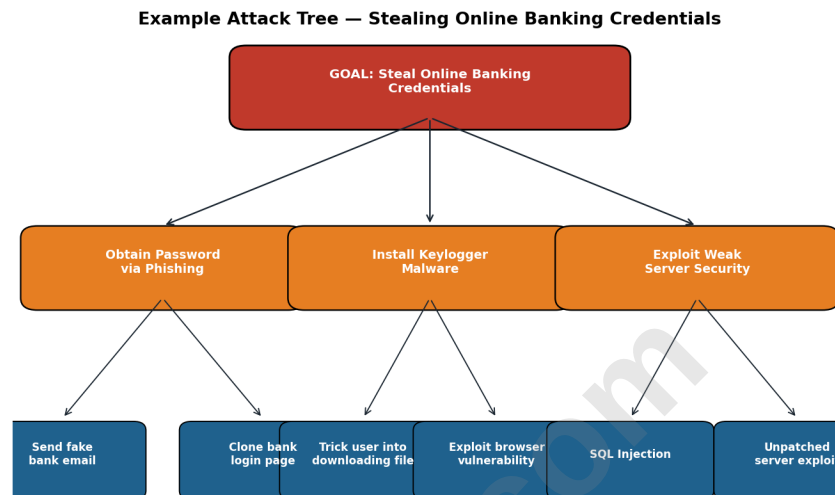


Fig. 1.6 — An example attack tree for the goal 'Steal Online Banking Credentials', showing how a high-level objective decomposes into specific, concrete attack techniques.

Note: Constructing an attack tree for a system early in its design forces defenders to think like attackers, often revealing weak points (such as the human attack surface) that purely technical security reviews can easily overlook.

1.7 Computer Security Strategy

A complete computer security strategy generally consists of three components working together: a security policy, a security implementation, and ongoing assurance and evaluation.

1.7.1 Security Policy

A security policy is a formal statement of the rules by which people who are given access to an organisation's technology and information assets must abide. It defines, in advance, what is considered an authorised use of resources and what is not, providing the baseline against which all later technical controls are designed and judged.

1.7.2 Security Implementation

Security implementation involves four complementary courses of action, each addressing the security problem from a different angle:

- **Prevention:** mechanisms (e.g., access controls, encryption) that stop an attack from succeeding in the first place.
- **Detection:** mechanisms (e.g., intrusion detection systems, audit logs) that identify an attack while it is happening or shortly after.
- **Response:** mechanisms and procedures used to stop an attack, prevent further damage, and repair any damage already done.
- **Recovery:** mechanisms (e.g., backups, redundant systems) that allow normal operations to resume after an attack or failure, such as restoring a database from a recent backup.

1.7.3 Assurance and Evaluation

Assurance refers to the degree of confidence one has that the security measures, both technical and operational, work as intended to protect the system and the information it processes. Evaluation is the process of examining a computer product or system with respect to certain criteria (such as the Common Criteria, an internationally recognised standard) to assess and certify the level of assurance it provides. Together, these activities give an organisation (and its customers, auditors, and regulators) confidence that the stated security policy is actually being upheld in practice, not just on paper.

Note: Security strategy is a continuous cycle, not a one-time project: policies must be revisited, implementations updated, and assurance re-evaluated as new threats, vulnerabilities, and business needs emerge over time.

Key Takeaways — Unit 1: Overview of Computer Security

- ✓ Computer security protects a system's hardware, software, and data; information security protects information in any form; network security protects data as it moves across networks — the three overlap heavily in practice.
- ✓ The CIA Triad (Confidentiality, Integrity, Availability) forms the foundation of security objectives, often extended with Authenticity and Accountability.
- ✓ Threats are potential security violations; attacks are deliberate attempts to realise a threat against an asset; attacks are classified as passive (eavesdropping/traffic analysis — emphasis on prevention) or active (masquerade/replay/modification/DoS — emphasis on detection and recovery).
- ✓ Security requirements (FIPS 200) translate the CIA+AA objectives into a practical checklist for evaluating any system.
- ✓ Security design principles (Saltzer & Schroeder and later additions) — economy of mechanism, fail-safe defaults, complete mediation, open design, least privilege, layering, and others — guide the construction of systems that are secure by design.
- ✓ The attack surface (network, software, human) describes every point an attacker could exploit; attack trees systematically map out the specific paths an attacker could follow to achieve a given goal.
- ✓ A complete security strategy combines a security policy (the rules), security implementation (prevention, detection, response, recovery), and assurance/evaluation (confidence that it all actually works).

— End of Unit 1 —