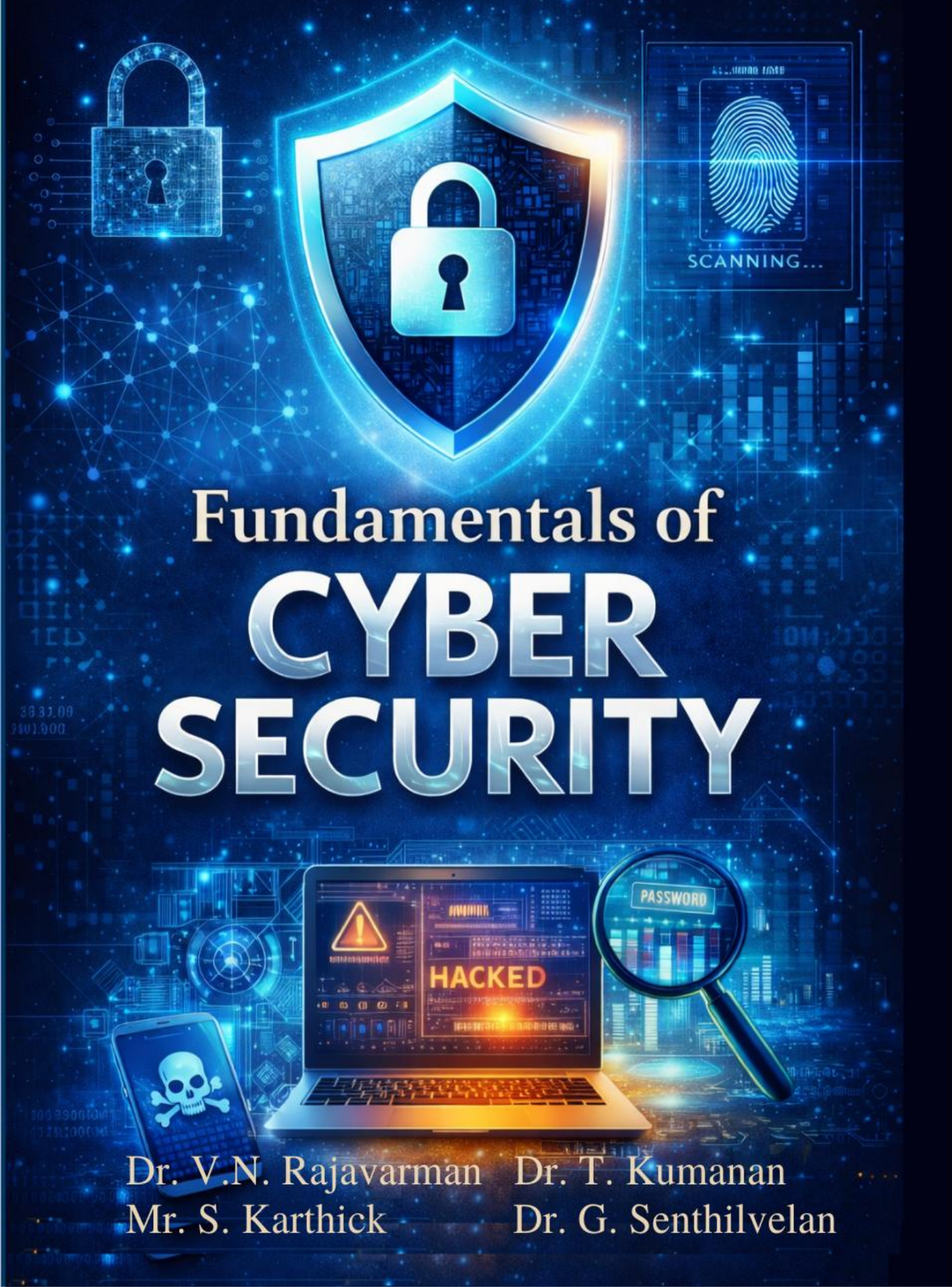


Fundamentals of CYBER SECURITY

Dr. V.N. Rajavarman
Dr. T. Kumanan
Mr. S. Karthick
Dr. G. Senthilvelan

Jupiter Publications Consortium

FUNDAMENTALS OF CYBER SECURITY

Dr. V. N. Rajavarman

Dr. T. Kumanan

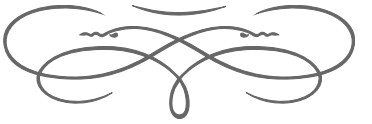
Mr. S. Karthick

Dr. G. Senthilvelan

Chennai, India

2026

Famous Quotes



“Security is a process, not a product.”

—— Bruce Schneier

“The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards—and even then I have my doubts.”

—— Gene Spafford

“An ounce of prevention is worth a pound of cure.”

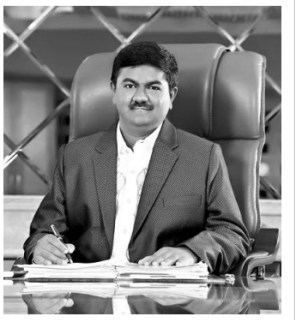
—— Benjamin Franklin

Copyright & Publishing Data

Title:	Fundamentals of Cyber Security
Authors:	Dr. V. N. Rajavarman; Dr. T. Kumanan; Mr. S. Karthick; Dr. G. Senthilvelan
Publisher:	Jupiter Publications Consortium
First Published:	February 2026
Edition:	First Edition
ISBN:	978-93-86388-76-6
DOI:	www.doi.org/10.47715/978-93-86388-76-6
MRP:	INR 450/-
Printing and Binding:	Printed in India
Cover and Layout:	Jupiter Publications Consortium
Typesetting by:	Jupiter Publications Consortium

Publisher Address
Jupiter Publications Consortium
22/102, Second Street, Venkatesa Nagar,
Virugambakkam, Chennai 600 092,
Tamil Nadu, India.

Dedication



Er. A.C.S. ARUNKUMAR
B.Tech (Hons)., LMISTE., MIET.,(UK).,
LMCSI.,

It is with profound pride and deep reverence that we dedicate this book to **Er. A. C. S. Arunkumar, B.Tech (Hons)., LMISTE., MIET., (UK)., LMCSI.**, the distinguished **President of Dr. M.G.R. Educational and Research Institute**, situated in the culturally rich city of **Chennai, Tamil Nadu, India**.

Our President's unwavering commitment to academic excellence and the advancement of knowledge stands as a testament to his global vision. His educational philosophy continues to inspire, serving as a guiding light that has illuminated the path to academic and personal growth for countless students, leaving an indelible mark on academic excellence.

Our gratitude for his visionary leadership is boundless, as his guidance consistently drives us to pursue excellence in every facet of our endeavours. It is not merely an honour but a privilege to dedicate this book to such a luminary—an enduring expression of our respect, admiration, and appreciation.

We extend our heartfelt thanks to you, Sir, for your remarkable contributions to education and for tirelessly inspiring us all with your leadership. Just as this book will serve future generations, so too will your legacy continue to inspire them.

Authors:
Dr. V. N. Rajavarman | Dr. T. Kumanan
Mr. S. Karthick | Dr. G. Senthilvelan

Abstract

Cybersecurity has become a foundational competence for modern computing professionals because digital systems now underpin education, commerce, governance, and everyday life. This book introduces cybersecurity as a structured discipline centered on protecting assets and managing risk, not merely as a set of tools or isolated "hacking" techniques. The text builds progressively from core terminology and security goals to threat landscapes, defensive controls, operating system and endpoint security, authentication and access control, malware fundamentals, cloud and container security basics, network security concepts, cryptographic foundations, web and application security, data security, security testing and reporting, and finally security operations and incident response. Throughout, emphasis is placed on responsible practice, conceptual clarity, and a defensible mapping between threats, vulnerabilities, controls, and verification.

Keywords: asset, threat, vulnerability, risk, CIA triad, attack surface, security controls, IAM, malware, incident response, logging, cryptography, web security, data protection, governance.

Preface

Cybersecurity is no longer an optional specialization; it is a core requirement for anyone who designs, deploys, operates, or manages digital systems. Universities rely on ERP/LMS platforms, cloud services, online assessments, and collaboration tools, while industries depend on networks, applications, identities, and data pipelines. As these systems grow in complexity and connectivity, security failures create compound consequences: disruption, financial loss, privacy violations, loss of trust, and—in some domains—real-world safety risks. This book is written to help learners develop disciplined security thinking that scales from academic environments to professional practice.

This text is designed for **B.Tech, M.Tech, and MCA** learners and assumes basic familiarity with computing fundamentals such as operating systems, networking basics, and programming concepts. However, no prior security specialization is required. The approach is deliberately **foundational and systems-oriented**: we begin with vocabulary and models that enable clear reasoning (asset, threat, vulnerability, risk; CIA triad), then progress to attacker behavior and lifecycle concepts, and finally move into controls and engineering practices across endpoints, identity, networks, applications, cloud, and data.

A recurring goal of this book is to connect concepts to decisions. In real environments, security work is not about memorizing attack names; it is about choosing controls that reduce risk, validating that they work, and responding effectively when prevention fails. For this reason, the book repeatedly uses the practical workflow: **assets** → **security goals** → **threats** → **vulnerabilities** → **controls** → **verification**. This workflow supports both technical and governance perspectives and prepares learners for entry-level roles as well as advanced study.

The book is organized into multiple parts that progress from foundations to systems and operations, reflecting how security is implemented in the real world—through layered controls, careful identity management, measurable monitoring, and ethical response capability. The learner is encouraged to treat security as a professional discipline where authorization, documentation, and responsible disclosure are integral to competence.

How to Use This Book

This book is designed to support multiple learning paths: structured semester courses, self-study, and skill-building for entry-level cybersecurity roles. Each chapter introduces concepts in a logical progression and reinforces them through summaries, key terms, and review questions. When used as a course textbook, learners should aim to complete each chapter's review set before moving to the next, because later topics assume familiarity with earlier definitions and models.

The recommended reading method is to first focus on **definitions and models**, then connect them to **examples**, and finally map each example to **controls**. For instance, when studying threats, do not stop at identifying the threat name; identify the asset at risk, the vulnerability that enables the threat, the likely impact, and the control categories that reduce likelihood or impact. This disciplined reasoning prevents security from becoming a set of disconnected facts.

Learners are encouraged to maintain a simple "security notebook" containing a running asset list, common threat patterns, and a control mapping table for familiar environments (campus labs, home networks, small apps, cloud accounts). Repeated practice of this mapping builds the core skill expected in both technical and governance roles: converting vague concerns into actionable security work.

Where mini-labs or workflows are suggested, they are intended for **safe, authorized environments** such as local virtual machines, classroom sandboxes, or purpose-built training platforms. The goal is to build competence without causing harm or violating policy.

Lab Safety & Ethics Notice

Cybersecurity knowledge must be applied responsibly. Many techniques used for defense—scanning, traffic analysis, exploitation concepts, credential testing, and log investigation—can cause disruption or violate law and policy if used without permission. Therefore, all hands-on activities suggested in this textbook must follow three non-negotiable conditions.

First, perform security testing only when you have **explicit authorization** from the system owner and your institution. Authorization must be written, specific, and time-bounded. "Curiosity" or "good intention" does not replace permission.

Second, use **safe environments**. Prefer isolated test systems, local virtual machines, or controlled lab networks. Avoid running tests on production systems, public networks, or systems that you do not own or manage.

Third, practice **professional documentation**. Maintain notes describing scope, tools used, timestamps, observed evidence, and remediation steps. Documentation is not optional; it is part of ethical and professional security practice.

This book supports learning for defense, risk management, and responsible engineering. Any misuse—including unauthorized access, data theft, service disruption, or bypassing controls—is unethical and may be illegal.

Contents

List of Figures

List of Tables

1

Cyber Security Foundations and Overview

1.1 Cyber Security: Scope and Importance

Section objectives

By the end of this section, a learner should be able to:

- **Define** cyber security and distinguish it from related terms like information security and IT security.
- **Identify** what is in scope (assets, systems, people, processes, data, and environments).
- **Explain** why cyber security is strategically important (business, safety, legal, and national-scale impacts).
- **Map** typical digital assets in a real context (university/enterprise) to likely threat categories.

What is Cyber Security

Cyber security is the discipline of protecting **digital assets**—such as data, software, hardware, networks, and services—from **unauthorized access, misuse, disruption, modification, or destruction**. It focuses on maintaining desired security properties (introduced in **Section 1.3: CIA triad and security goals**) across the full lifecycle of systems: **design → deployment → operation → retirement**.

Cyber security is not only "preventing hacking." It is also:

- **Reducing risk** to acceptable levels
- **Detecting** abnormal or malicious behavior quickly
- **Responding and recovering** effectively when incidents occur.

Common confusion:

- **Information security** emphasizes protection of information in *any form* (paper, verbal, digital).
- **Cyber security** emphasizes protection in *digital/connected* contexts (networks, systems, cloud, internet-facing services).

Scope: What Cyber Security Covers

Cyber security spans multiple layers. A practical way to understand scope is to ask: **"What are we protecting, from whom, and how?"**

1) Assets (What we protect)
Examples across common domains:

- **Data:** student records, payroll, source code, customer PII, transaction logs
- **Systems:** servers, laptops, mobile devices, endpoints in labs
- **Networks:** campus LAN/Wi-Fi, routers, DNS, VPN gateways
- **Applications:** ERP portals, learning management systems, payment systems, APIs
- **Identities:** user accounts, admin privileges, service accounts, keys/tokens
- **Infrastructure and services:** cloud workloads, containers, third-party SaaS

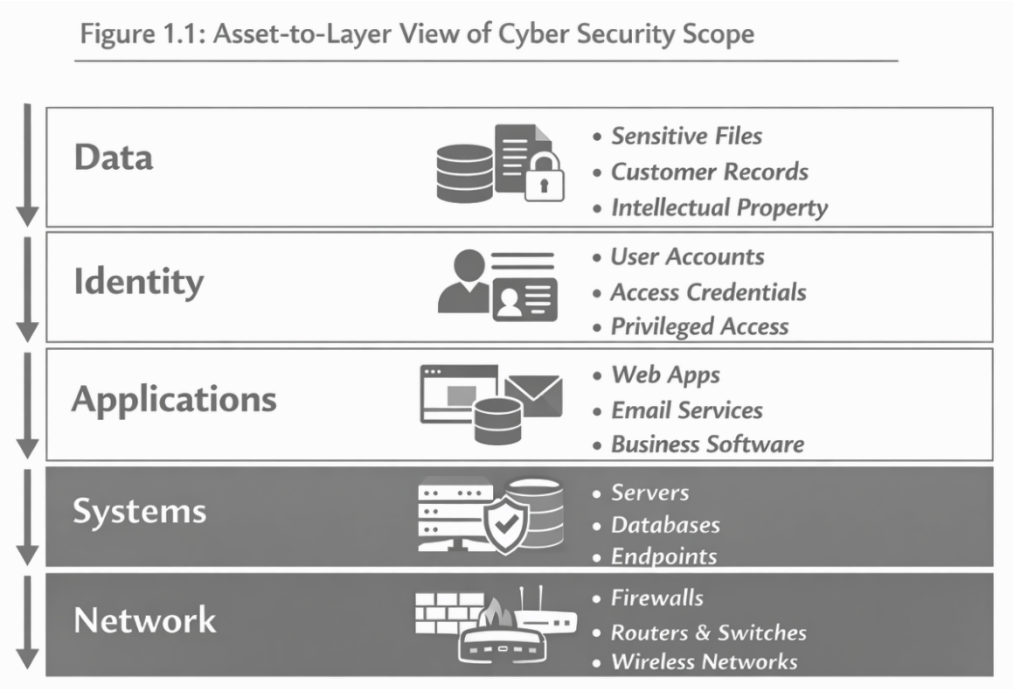


Figure 1.1. Asset-to-Layer View of Cyber Security Scope

2) Adversaries and uncertainty (Who/what threatens)
Threat sources include:

- **External attackers:** cyber criminals, opportunistic attackers, hacktivists
- **Insiders:** negligent users, malicious insiders, contractors
- **Third parties:** vendors, supply chain dependencies
- **Environmental/operational failures:** misconfigurations, outages, human error

3) Attack surfaces (Where attacks happen)
The **attack surface** is the set of entry points where an attacker might interact with a system. Typical surfaces include:

- Internet-facing services (web apps, APIs, SSH/RDP)
- Email and collaboration systems (phishing, malicious attachments)
- Endpoints (USBs, drive-by downloads, software installs)
- Cloud consoles and IAM misconfigurations
- Weak authentication flows (password reuse, poor MFA rollout)

Table 1.1. Examples of Attack Surfaces and Typical Risks

Asset/Surface	Example	Likely Risk	Basic Control Category
Internet-facing web application	University ERP / LMS portal reachable from public internet	Credential stuffing, SQL injection (concept), broken access control	Prevent (secure coding, WAF), Detect (logs/alerts)
Public API endpoints	REST API for mobile app / partner integration	Excessive data exposure, weak auth tokens, rate-limit bypass	Prevent (authz checks, rate limits), Detect (API monitoring)
Remote administration interface	SSH/RDP/WinRM exposed or poorly restricted	Brute force, account takeover, lateral movement	Prevent (VPN, allowlists, MFA), Detect (auth logs)
Email system	Staff email with attachments/links	Phishing, malicious attachments, credential theft	Prevent (MFA, filtering), Detect (user reports, SIEM)
Endpoint devices	Student lab PCs, faculty laptops	Malware infection, data theft, ransomware spread	Prevent (patching/EDR), Correct (backups)
Wi-Fi network	Open/weakly protected campus Wi-Fi	Eavesdropping, rogue AP, session hijack (concept)	Prevent (WPA2/3, segmentation), Detect (wireless monitoring)
DNS resolution path	Campus DNS resolver or misconfigured DNS	DNS spoofing/poisoning (concept), redirect to fake sites	Prevent (DNSSEC where feasible), Detect (DNS logs/anomaly)
Cloud storage buckets	Misconfigured object storage for backups/data	Public data exposure, unauthorized downloads	Prevent (least privilege, private-by-default), Detect (access logs)
Identity provider / SSO portal	SSO login page for multiple services	Account takeover, MFA fatigue (concept), token theft	Prevent (strong MFA, conditional access), Detect (impossible travel alerts)

Asset/Surface	Example	Likely Risk	Basic Control Category
Third-party SaaS integrations	Payment gateway, HR SaaS, code hosting	Supply-chain risk, token leakage, data sharing misconfig	Prevent (vendor controls, secrets mgmt), Detect (audit trails)
Software update mechanism	Internal package repo / auto-updates	Malicious update, dependency compromise	Prevent (signed updates, SBOM concept), Detect (integrity checks)
Physical ports & removable media	USB ports in labs/office systems	Malware via USB, data exfiltration	Prevent (device control), Detect (endpoint logs)

4) Controls (How we reduce risk)
Controls broadly fall into:

- **Preventive** (stop bad things): MFA, patching, access control, secure configs
- **Detective** (notice bad things): logs, IDS/IPS, SIEM alerts
- **Corrective/Responsive** (recover): backups, incident response, containment

(These are expanded in **Section 1.5: Security Controls: Prevent–Detect–Respond**.)
Why Cyber Security Matters
Cyber security is important because failures typically create **compound damage**:

1. **Financial impact**

- Fraud, extortion (e.g., ransomware), downtime, recovery costs, and lost revenue.

2. **Operational impact**

- Disruption of critical services: campus ERP unavailability, payment outages, manufacturing stoppage.

3. **Legal and compliance impact**

- Data protection obligations (privacy laws, contractual commitments) and audit requirements. *Note: "verify current text/version" for regulations.*

4. **Reputational impact**

- Trust loss can outlast technical recovery.

5. **Safety and societal impact**

- For cyber-physical systems (healthcare, transport, utilities), security incidents can affect human safety.

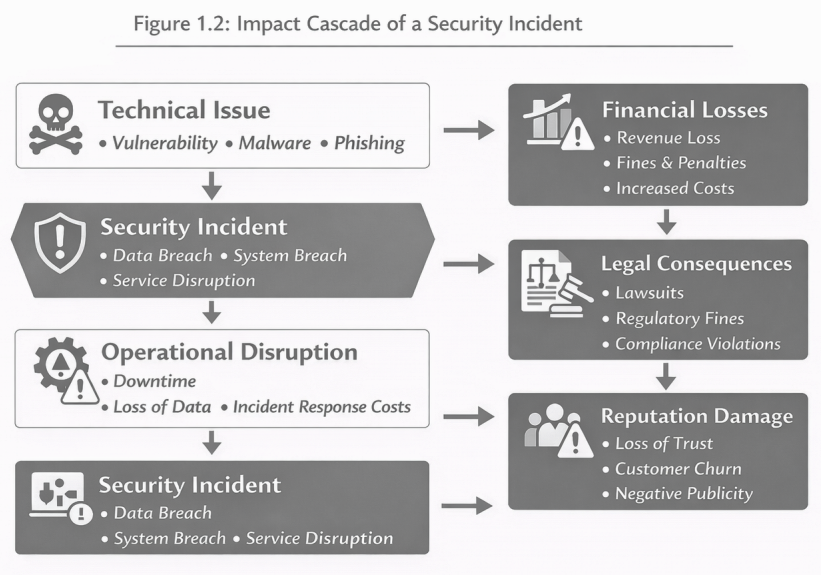


Figure 1.2. Impact Cascade of a Security Incident

Worked Example: Scoping Cyber Security for a University ERP

Scenario: A university runs an ERP portal for admissions, attendance, fees, and results.

Step 1 — List key assets

- Student PII (identity, address, marks)
- Payment records (fee receipts, transactions)
- Admin accounts (department staff, IT admins)
- ERP web app + database
- Backups + audit logs

Step 2 — Identify key security properties

- **Confidentiality:** student PII should not leak
- **Integrity:** marks and fee records must not be altered
- **Availability:** portal must be reachable during admission/results

Step 3 — Identify likely threat categories

- Credential stuffing/phishing against staff accounts
- Injection/misconfiguration risks in web app
- Ransomware on servers/endpoints affecting availability
- Misconfigured backups or exposed cloud storage

Step 4 — Map basic controls (high-level)

- MFA for staff/admin; strong password policy
- Secure configuration + patching of servers and dependencies
- Logging + monitoring of authentication and admin actions
- Tested backups + recovery drills

This is the core pattern you'll repeat throughout the book: **assets → security goals → threats → controls → verification**.
Mid-section Checkpoint (Self-test)

1. In one sentence, define "cyber security" in terms of assets and objectives.
2. Give two examples of **attack surface** in a campus network and one control for each.
3. Why is "availability" a security concern, not just an IT reliability concern?
4. What is one difference between information security and cyber security?
5. For a university ERP, which asset would you prioritize first and why?

Short takeaway
Cyber security is a **multi-layer, socio-technical** discipline: it protects assets across **technology + people + process**, manages evolving threats, and aims to sustain confidentiality, integrity, and availability while enabling the organization's goals.

1.2 Core Terms: Asset, Threat, Vulnerability, Risk

Section objectives
By the end of this section, you should be able to:

- **Define** asset, threat, vulnerability, and risk with precision (Remember/Understand).
- **Differentiate** threat vs vulnerability, and risk vs impact (Analyze).
- **Apply** a simple workflow to identify assets and risks in a real scenario (Apply).
- **Explain** why "risk" is the central management quantity in cyber security (Evaluate).

1.2.1 Why these four terms matter

Cyber security discussions become confusing when people use the same word to mean different things. The four terms in this section—**asset**, **threat**, **vulnerability**, **risk**—form the minimum vocabulary for structured security thinking.
You will reuse these terms in:

- **Chapter 2** (threat landscape, attack lifecycle),
- **Chapter 4** (risk management and governance),
- **Chapters 15–16** (monitoring and incident response).

1.2.2 Asset

An **asset** is anything that has value and therefore needs protection.
Assets can be:

- **Information assets:** personal data, exam papers, source code, research data
- **Technology assets:** servers, routers, laptops, cloud workloads
- **Service assets:** online admission portal, payment processing, email service
- **Identity assets:** admin accounts, API keys, certificates
- **Reputational/operational assets:** trust, uptime, institutional credibility (managed indirectly through technical controls)

Key point: In security, "value" is not only monetary. It can be academic integrity, continuity of service, legal exposure, and safety.

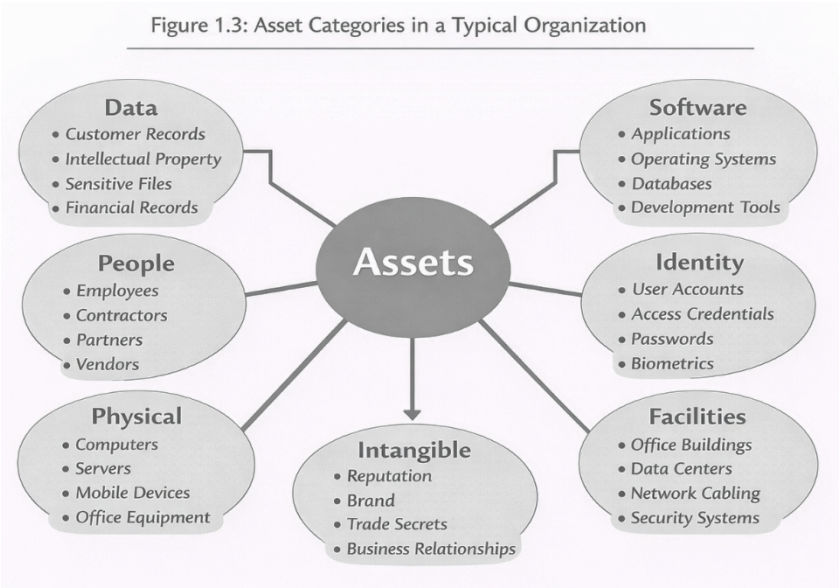


Figure 1.3. Asset Categories in a Typical Organization
Example (campus):
The *exam timetable PDF* is an asset, but so is the *account that can modify it* and the *system that publishes it.*

1.2.3 Threat

A **threat** is a potential cause of an unwanted incident that could harm an asset.
Threats can be described as:

- **Threat actors:** student pranksters, cyber criminals, insiders, competitors

- **Threat events:** phishing, ransomware, unauthorized access, data leakage, misconfiguration, power outage

Threats are not always malicious. Accidental threats (misclicks, wrong access grants) are common and often more frequent than targeted attacks.

Table 1.2: Threat Types and Representative Examples

Threat Type	Example Event	Primary Asset Affected	Typical Impact
Social engineering	Phishing email tricks user into entering credentials on fake login page	User identity (credentials), email account	Account takeover, data access, further phishing from trusted account
Malware (general)	User runs infected attachment; endpoint gets compromised	Endpoint device, local files	Data theft, persistence, lateral spread, service disruption
Ransomware	File shares encrypted after compromise of an admin account	Data stores, shared drives, backups	Loss of availability, recovery cost, operational downtime
Web application attacks	Exploiting weak authorization logic in a portal	Application data, user records	Unauthorized data access/changes, privacy breach
Injection (concept)	Unsanitized input causes unintended database query execution	Database, application integrity	Data leakage, data tampering, service disruption
Credential attacks	Password reuse enables credential stuffing against ERP	Identity systems, SSO accounts	Unauthorized access, privilege escalation, fraud
Misconfiguration / exposure	Cloud storage bucket accidentally set to public	Data repository, backups	Large-scale data leakage, compliance issues
Insider misuse	Staff member accesses records beyond job need	Sensitive data, audit trails	Confidentiality breach, legal/reputational damage
Third-party / supply chain	Compromised dependency/library introduced into application	Application codebase, build pipeline	Backdoor risk, data exfiltration, integrity compromise
Network-based attacks (concept)	Spoofing/redirecting traffic via DNS manipulation	Network services (DNS), user sessions	Traffic interception, credential theft, redirection to malicious sites
Availability attacks	DoS/DDoS overwhelms public-facing service	Online services, network bandwidth	Service outage, degraded performance, revenue/operational loss
Physical / environmental	Laptop theft or power failure in server room	Endpoint devices, on-prem servers	Data exposure, downtime, recovery cost

Example:
"Phishing" is a threat event. "External attacker using phishing" is a threat actor + threat event statement.

1.2.4 Vulnerability

A **vulnerability** is a weakness that can be exploited by a threat to harm an asset.
Vulnerabilities may exist in:

- **Software:** unpatched bugs, insecure coding patterns
- **Configuration:** default passwords, open storage buckets, weak firewall rules
- **Design/architecture:** poor trust boundaries, missing authorization checks
- **Process/people:** weak onboarding/offboarding, low security awareness

Important distinction:

- A **threat** is *something that could happen* (cause).
- A **vulnerability** is *a weakness that enables it* (condition).

Figure 1.4. Threat-Vulnerability-Impact Relationship
Example:

- Threat: credential theft via phishing
- Vulnerability: users reuse passwords and no MFA is enforced
- Outcome: attacker logs in and downloads sensitive data

1.2.5 Risk

Risk is the likelihood that a threat will exploit a vulnerability to harm an asset, combined with the severity of the consequences.
A common conceptual model is:
Proportional form

$$Risk \propto Likelihood \times Impact$$

Meaning (with details)

- **Risk:** the overall "expected" harm/concern from an event (e.g., a breach).
- **Likelihood:** how probable it is that the event occurs (or how often it might occur).
- **Impact:** how severe the consequences are if it occurs (financial loss, downtime, legal exposure, reputational damage, etc.).
- $\propto$ **("proportional to"):** risk increases as **likelihood** or **impact** increases, but it's not claiming a strict equality unless you define a constant/scale.

Practical scoring version (common in risk matrices)

$$Risk\ Score\ =\ Likelihood\ Score\ \times\ Impact\ Score$$

OR (linear input)

$$Risk\ Score\ =\ Likelihood\ Score\ \times Impact\ core$$

Example (1–5 scale): Likelihood = 4, Impact = 5 $\Rightarrow$ Risk = 20 (high).
This does **not** mean risk is always numerical; it can be qualitative (Low/Medium/High) or semi-quantitative (scores). The formal treatment is in **Chapter 4**, but you need the intuition now:

- **Likelihood** depends on exposure, attacker capability, ease of exploitation, and existing controls.
- **Impact** depends on what breaks: confidentiality, integrity, availability, legal obligations, and trust.

Risk is not the same as impact.

- Impact: "If it happens, how bad is it?"
- Risk: "How probable + how bad?"

Worked Example: Turning an "Issue" into a Risk Statement
Scenario: A department website runs on an old CMS version.
Step 1 — Asset: Department website + its content + admin account
Step 2 — Threat: External attacker compromises the site
Step 3 — Vulnerability: CMS is unpatched; known vulnerabilities may exist
Step 4 — Risk statement (good format):
"There is a risk that an external attacker exploits a known CMS vulnerability to deface the department website or upload malicious content, causing reputational damage and potential malware distribution."
Why this matters: This risk statement is actionable: it points to patching, hardening, monitoring, and incident response.
[Listing 1.1: "Risk Statement Template (Actionable Format)" | Language: text
Risk: There is a risk that <threat actor/event> exploits to impact , leading to , because <exposure/condition>.
Controls (current): <prevent/detect/respond controls if any>
Proposed actions: <top 3 actions>
Owner: <role/team> | Due date: | Residual risk: <L/M/H>
Mid-section Checkpoint (Self-test)

1. Give one example each of an **asset**, **threat**, **vulnerability**, and **risk** for a college Wi-Fi network.

2. Is “ransomware” a threat or a vulnerability? Explain.
3. Why is “default password” a vulnerability even if no attack has happened yet?
4. Convert this into a risk statement: “Our database is open to the internet.”
5. In your own words, what is the difference between risk and impact?

Common Mistakes & Misconceptions (Callout Box)

- Treating **threat** and **vulnerability** as the same thing (“SQL injection is a threat/vulnerability” — it’s better described as an attack technique enabled by vulnerabilities).
- Listing “hacker” as the risk instead of writing an actionable **risk statement**.
- Assuming risk is only about external attackers; **insider and accidental risks** are often dominant.
- Thinking “if we haven’t been attacked, we are safe” (risk exists even without observed incidents).
- Ignoring **identity assets** (accounts, keys, tokens) and focusing only on servers/data.

Security Mindset (Callout Box)

- Start with **assets and value**: protect what matters most, not what is easiest.
- Convert vague fears into **structured risk statements** that can be owned and tracked.
- Ask “What would an attacker need?”—then look for weaknesses that reduce their cost.
- Treat misconfiguration as a first-class security problem (not “just ops”).
- Controls should be layered: prevention reduces likelihood; detection limits dwell time; recovery limits impact.

Ethical Use Reminder
These terms are used to improve defense and risk management. Any testing or validation must be **authorized**, performed in **safe sandbox environments**, and aligned with institutional policies and law.
Short takeaway

- **Asset** = what has value
- **Threat** = what could cause harm
- **Vulnerability** = weakness that enables harm
- **Risk** = likelihood $\times$ impact (conceptually), used to prioritize security work

1.3 CIA Triad and Security Goals

The **CIA Triad** is a foundational model that defines the three core **security objectives** most controls and policies aim to achieve: **Confidentiality, Integrity, and Availability**. These objectives translate into practical security goals for protecting organizational assets (data, identities, applications, systems, and networks).

1.3.1 Overview of the CIA Triad

- A simple way to define **what "secure" means** for information and systems.
- Used to:
  - frame security requirements,
  - prioritize controls,
  - assess incident impact,
  - communicate risk to technical and non-technical stakeholders.
- Most incidents affect **more than one** CIA element (e.g., ransomware hits availability *and* integrity).

1.3.2 Confidentiality

Goal: Prevent **unauthorized disclosure** of information.

- **What it protects against:** data leaks, snooping, credential misuse, eavesdropping, insider abuse
- **Typical controls:**
  - Access control (RBAC/ABAC), least privilege
  - Encryption (at rest/in transit)
  - Data classification & handling rules
  - DLP, secrets management
- **Examples:**
  - Only HR can access payroll records
  - Customer PII encrypted in databases and APIs

1.3.3 Integrity

Goal: Prevent **unauthorized or improper modification** of data/systems; ensure accuracy and trustworthiness.

- **What it protects against:** tampering, fraud, silent config changes, corruption
- **Typical controls:**
  - Hashing/checksums, digital signatures
  - Change control, versioning, code signing
  - Database constraints, input validation
  - Audit logs, integrity monitoring (FIM)
- **Examples:**
  - Detecting altered invoices or payment details
  - Ensuring deployment artifacts aren't modified

1.3.4 Availability

Goal: Ensure systems and data are **accessible and usable when needed**.

- **What it protects against:** outages, DDoS, ransomware disruption, hardware/cloud failures
- **Typical controls:**
  - Redundancy, failover, load balancing
  - Backups, disaster recovery (DR), business continuity (BC)
  - Monitoring/alerting, capacity planning
  - Patch management and resilience engineering
- **Examples:**
  - Meeting uptime SLAs for customer-facing services
  - Restoring systems quickly after an incident

1.3.5 Security Goals Derived from CIA

Organizations usually convert CIA into **operational goals** such as:

- **Prevent:** stop unauthorized access/modification/disruption (controls, hardening)
- **Detect:** identify violations quickly (logging, monitoring, alerting)
- **Respond:** contain and eradicate threats (IR playbooks, segmentation)
- **Recover:** restore services and trust (backups, DR, rollback, communications)

This is why modern security programs emphasize **resilience**, not just prevention.

1.3.6 Additional Security Goals Beyond CIA

Many organizations extend CIA with supporting objectives:

- **Authentication / Authenticity:** verify identities and origins (MFA, certificates, signed updates)
- **Authorization:** enforce what an identity can do (least privilege, policy enforcement)
- **Accountability / Auditability:** actions are traceable (logs, SIEM, tamper-resistant records)
- **Non-repudiation:** users cannot deny actions (digital signatures, secure audit trails)
- **Privacy:** proper use and protection of personal data (lawful processing, minimization)

1.4 Security Domains (Network, App, Data, Cloud)

Security programs are commonly structured into **domains**—major areas of focus that group threats, controls, and ownership. Domains overlap in practice, but this breakdown helps define scope and implement **defense-in-depth**.

In this section you will learn to:

1. Distinguish four core security domains.

Dr. V. N. Rajavarman, a distinguished professor at Dr. M.G.R. Educational and Research University, Chennai, earned his doctorate from the same institution. Since beginning as a Lecturer in 1992, he has accumulated 28 years in academia and four in industry, advancing to Professor in Computer Science. He has authored over 75 internationally indexed journal articles and published 6 books and one patent. As a doctoral guide, he has supervised 17 PhDs and is mentoring 8 more. Honored with numerous awards, he currently serves as Professor, Dean of Part-time Studies, and Additional Dean of Computer Studies at Dr. M.G.R. University.

Prof. Dr. T. Kumanan, M.E., Ph.D., is a distinguished academician with over 29 years of teaching and research experience in Computer Science and Engineering and serves as Professor at Dr. MGR Educational and Research Institute, Chennai. He earned his Ph.D. from Anna University and holds a Master's degree in Computer Science and Engineering. A recognized Ph.D. supervisor, he has guided 17 scholars and supervises six more. He has published over 63 research papers, serves on editorial and review boards of international journals, and received the Global Faculty Award (2022) and Research Excellence Award (2023).

Mr. S. Karthick is an academic professional with a strong foundation in Information Technology. He completed his B.Tech. and M.Tech. degrees at Manonmaniam Sundaranar University, Tirunelveli. He began his career as a Software Engineer, gaining valuable industry experience before transitioning to academia. He currently serves as an Assistant Professor at Sethu Institute of Technology, where he is engaged in teaching, mentoring students, and contributing to institutional development initiatives. He is pursuing his Ph.D. at Dr. M.G.R. Educational and Research Institute. Actively associated with the Youth Hostels Association of India, he remains committed to research, teaching excellence, and student development.

Dr. Senthilvelan G, Assistant Professor in Computer Science and Engineering at Dr. M.G.R. Educational and Research Institute, Chennai, has 15 years of academic experience. He earned his Master's in Computer Science and Engineering from the same institution and his Ph.D. from St. Peter's University, Chennai. His research work includes publications in Scopus-indexed journals and one published book. Actively contributing to academic development, he has organized numerous seminars, workshops, and conferences. His innovative approach is reflected in the acquisition of an Indian patent in Computer Science and Engineering, underscoring his commitment to advancing research, technology, and education in his field.

DOI: 10.47715/978-93-86388-76-6
ISBN: 978-93-86388-76-6
Publisher: Jupiter Publications Consortium
Published URL: www.jpc.in.net