UPractice.org

CPA ISC Study Guide

Information Systems & Controls — chapter-by-chapter review

2026 CPA Exam Blueprint · SOC, COSO, NIST CSF 2.0

Free study guide from UPractice.org — original material written for independent exam preparation. Not affiliated with, endorsed by or sponsored by any exam owner or government agency; exam names belong to their owners. Laws, amounts and exam rules change: always confirm with the official source before the exam. © UPractice.org. Free to download and print for personal study.

Contents

1. How to Use This Guide and the ISC Exam Format
    - The ISC section at a glance
    - Which frameworks and standards this guide uses
    - How each chapter is organized
    - The calculations you must be able to do
    - A study plan that works
    - Test-day strategy
    - Key takeaways
2. 1. IT Infrastructure and Cloud Computing
    - Infrastructure building blocks
    - Cloud computing — the NIST definition
    - The shared responsibility model
    - Availability and service level agreements
    - Other infrastructure concepts that appear on the exam
    - Common traps
    - Key takeaways
3. 2. Enterprise and Accounting Information Systems
    - Enterprise systems
    - Transaction cycles
    - The three-way match
    - Processing methods
    - Files and records
    - Data interchange and integration
    - System conversion approaches
    - Emerging technologies and their controls
    - Common traps
    - Key takeaways
4. 3. IT General Controls and Application Controls
    - COSO 2013 and technology
    - IT general controls versus application controls
    - Control classifications
    - Input controls (edit checks)

Processing controls
Output controls
IT segregation of duties
Common traps
Key takeaways

5. 4. Change Management and the SDLC
The SDLC phases
Testing levels
The change management process
Post-implementation review
Data conversion controls
Common traps
Key takeaways

6. 5. Business Resiliency, Backup and Recovery
Business continuity versus disaster recovery
Business impact analysis (BIA)
Recovery site strategies
Backups
RAID and fault tolerance
Reliability formulas
Testing the plan
Common traps
Key takeaways

7. 6. Data Management, Integrity and Analytics
The data life cycle
Data governance roles
Data quality dimensions
Data storage architectures
Retention and destruction
Data analytics
Common traps
Key takeaways

8. 7. Databases and SQL
Relational database basics
Normalization

Database transactions and ACID
SQL command categories
The sample tables
Logical order of a SELECT query
Filtering rows with WHERE
Aggregates, GROUP BY and HAVING
Joins
COUNT and null rules

9. 8. Security Frameworks and Governance
Why frameworks?
NIST Cybersecurity Framework 2.0
COBIT 2019
CIS Critical Security Controls v8.1
ISO/IEC 27001 and 27002
Policies, standards, procedures and guidelines
Security governance roles
Risk management
Common traps
Key takeaways

10. 9. Threats, Attacks and Cyber Risk
Threat actors
Social engineering
Malware
Application attacks
Network and other technical attacks
The CIA triad and attack mapping
Quantitative risk analysis: SLE and ALE
Defense in depth
Common traps
Key takeaways

11. 10. Logical and Physical Access Controls
The access control process
Authentication factors
Passwords
Biometrics

Authorization models
The user access life cycle
Privileged access management
Physical and environmental controls
Common traps
Key takeaways

12. 11. Encryption, PKI and Network Security
Cryptography vocabulary
Symmetric encryption
Asymmetric (public key) encryption
Hashing
Digital signatures
Public key infrastructure (PKI)
Protecting payment and sensitive data
Network security
Common traps
Key takeaways

13. 12. Vulnerability Management and Incident Response
The vulnerability management cycle
Prioritizing with CVSS and context
Patch and configuration management
Incident response
Digital forensics
SEC cybersecurity incident disclosure
Measuring the program
Common traps
Key takeaways

14. 13. Privacy and Data Protection Rules
Privacy foundations
GDPR (European Union)
HIPAA (U.S. health information)
GLBA (U.S. financial institutions)
CCPA / CPRA (California)
PCI DSS (payment cards)
Choosing the right rule

Common traps
Key takeaways
15. 14. SOC Engagements: Types and Planning
The parties
The SOC report family
Type 1 versus Type 2
Trust Services Criteria (SOC 2)
Subservice organizations: carve-out or inclusive
Complementary user entity controls (CUECs)
Planning a SOC engagement
Report periods and bridge letters
Common traps
Key takeaways
16. 15. SOC Testing, Reporting and Opinions
Evidence for operating effectiveness
Structure of a SOC report
Opinion types
The user auditor: AU-C 402
Common traps
Key takeaways
17. Final Review Checklist
Numbers and lists sheet
Formula sheet
Area I — Information systems and data management (Chapters 1–7)
Area II — Security, confidentiality and privacy (Chapters 8–13)
Area III — SOC engagements (Chapters 14–15)
The night before

How to Use This Guide and the ISC Exam Format

This guide is a chapter-by-chapter review of the **Information Systems and Controls (ISC)** discipline section of the Uniform CPA Examination. It follows the same fifteen chapters as the ISC practice bank on UPractice.org, so you can read a chapter and then immediately drill the matching multiple-choice questions and simulations.

ISC is the technology discipline. It builds on the internal control and IT material in AUD and asks you to think like an IT auditor, a controls consultant or a service auditor. You will need to know how systems are built and run, how data is stored and queried, how security programs are organized, which privacy rules apply, and how SOC engagements are planned, performed and reported.

The ISC section at a glance

The CPA Exam has three core sections (AUD, FAR, REG) plus one discipline section of your choice (BAR, ISC or TCP). ISC is the systems-and-controls discipline.

Feature	ISC (2026 Blueprints)
Testing time	4 hours
Multiple-choice questions (MCQs)	82, in two testlets of 41
Task-based simulations (TBSs)	6, in three testlets (1, 3 and 2)
Total testlets	5
Standardized break	15 minutes, offered after testlet 3 (does not count against your time)
Score weighting	MCQs 60%, TBSs 40%
Score scale / passing score	0–99 / 75
Skill levels	Remembering and understanding 55–65%; Application 20–30%; Analysis 10–20%

Two features make ISC different from the other sections. First, it has the **most MCQs** of any section and is the only section where MCQs carry **60%** of the score. Second, the skill mix leans toward **remembering and understanding**. You still need to calculate (availability, RPO, ALE, deviation rates, SQL results), but a large share of the points come from knowing definitions, frameworks and report types precisely.

Content areas and weights

Blueprint area	Weight	Chapters in this guide
I. Information Systems and Data Management	35–45%	1–7
II. Security, Confidentiality and Privacy	35–45%	8–13
III. Considerations for System and Organization Controls (SOC) Engagements	15–25%	14–15

Area I covers IT infrastructure and cloud, enterprise and accounting systems, IT general and application controls, change management, resiliency and recovery, data management and analytics, and databases and SQL. Area II covers security frameworks, threats, access controls, encryption and network security, vulnerability management and incident response, and privacy rules. Area III covers planning, testing and reporting on SOC 1, SOC 2 and SOC 3 engagements.

> **Exam tip:** Area III is the smallest area but the one most candidates find least familiar, and it is very rule-based. Chapters 14 and 15 are short, high-yield and worth reading twice.

Which frameworks and standards this guide uses

ISC tests frameworks by name. This guide uses the versions in effect for 2026 testing:

Topic	Framework or rule used in this guide
Internal control	COSO *Internal Control — Integrated Framework* (2013): 5 components, 17 principles
IT governance	COBIT 2019
Cybersecurity	NIST Cybersecurity Framework (CSF) 2.0, released February 2024 (six Functions, including Govern)
Security controls	NIST SP 800-53 Rev. 5; CIS Critical Security Controls v8.1; ISO/IEC 27001:2022
Cloud and zero trust	NIST SP 800-145 (cloud definition); NIST SP 800-207 (zero trust)
Digital identity	NIST SP 800-63B (Revision 4 issued in 2025)
Incident response	NIST SP 800-61 Rev. 3 (April 2025), aligned with CSF 2.0
Vulnerability scoring	CVSS qualitative severity bands
Privacy and data protection	GDPR, HIPAA, GLBA Safeguards Rule (as amended), CCPA/CPRA, PCI DSS v4.0.1
Public company cyber disclosure	SEC Form 8-K Item 1.05 and Regulation S-K Item 106
SOC engagements	SSAE No. 18 and No. 21 (AT-C 105, 205, 206, 320); AU-C 402 for user auditors; 2017 Trust Services Criteria (with revised points of focus, 2022); DC 200 description criteria

> **Watch out:** Many older study materials still describe NIST CSF with **five** Functions. CSF 2.0 has **six** — Govern was added. If an answer choice relies on the old structure (for example, putting supply chain risk management under Identify), it is out of date.

How each chapter is organized

Every chapter follows the same pattern so you always know where to look:

1. **Plain-language explanation** of each rule or concept the bank tests.
2. **Key-fact tables** — framework parts, thresholds, deadlines and formulas.
3. **Worked examples** with every step shown. Every number was computed and re-checked before it was printed, and every SQL query was run against the sample tables to confirm its result.
4. **Exam tips, traps and memory tricks** in colored boxes.
5. **Key takeaways** — a short summary to reread the night before the exam.

Callout boxes use four colors: green for exam tips, blue for worked examples, red for common traps and purple for memory tricks.

The calculations you must be able to do

ISC is not a math-heavy section, but the same handful of computations appear again and again. Each is worked in this guide:

Computation	Formula	Chapter
Allowed downtime under an SLA	Period in minutes × (1 − availability)	1
Composite availability	Serial: multiply; parallel: 1 − product of failure rates	1
Three-way match payment	Lowest of ordered, received and billed quantity × PO price	2
Batch control totals	Record count, financial total, hash total	2
Check digit (Luhn)	Double every second digit from the right, sum, round up to 10	3
RPO / RTO / MTD tests	Worst-case loss = backup interval; RTO + WRT ≤ MTD	5
Availability from reliability data	MTBF ÷ (MTBF + MTTR)	5
RAID usable capacity	RAID 5 = n − 1 disks; RAID 6 = n − 2; RAID 10 = n ÷ 2	5
Benford expected frequency	log10(1 + 1/d)	6
SQL result sets	WHERE, GROUP BY, HAVING, joins, COUNT rules	7
Annualized loss expectancy	SLE = AV × EF; ALE = SLE × ARO	9
Biometric error rates	FAR = false accepts ÷ impostor attempts; FRR = false rejects ÷ genuine attempts	10
Key counts	Symmetric n(n − 1) ÷ 2; asymmetric 2n	11
Notification deadlines	72 hours, 4 business days, 30 days, 60 days	12–13
Deviation rate	Deviations ÷ items tested	15

A study plan that works

1. **Read the chapters in order.** Area I (Chapters 1–7) gives you the vocabulary of systems and data. Area II (8–13) layers security and privacy on top. Area III (14–15) uses everything before it.

2. **Read, then drill immediately.** After each chapter, do a timed set of 25–40 MCQs from the matching bank chapter, then its simulations.
3. **Memorize the high-yield lists.** COSO's 5 components, CSF 2.0's 6 Functions, the SOC report types, Type 1 versus Type 2, the CVSS bands, and the privacy deadlines (72 hours, 4 business days, 30 days, 60 days).
4. **Practice every computation** until you can do it without notes. They are quick points.
5. **Write SQL by hand.** Predict the result of a query on a small table before checking it. The exam shows you tables and asks what a query returns.
6. **Take full mock exams** under 4-hour timing in the last two weeks and review every miss by chapter.

Test-day strategy

- **Budget your time.** With 82 MCQs, aim for about 1.2–1.4 minutes per question (roughly 100–115 minutes for both MCQ testlets). That leaves about 2 hours for six simulations, or roughly 18–20 minutes each.
- **Watch scope words.** ISC answer choices are often all partly true. Words such as **MOST**, **BEST**, **FIRST**, **PRIMARY** and **EXCEPT** decide the item.
- **Classify before you answer.** Many items reduce to a classification: preventive or detective, ITGC or application control, SOC 1 or SOC 2, Type 1 or Type 2, owner or custodian. Decide the category first.
- **In simulations, read the requirement first**, then the exhibits. Answer the easy rows before the computations, and answer every cell — a blank earns nothing.

> **Memory trick:** "Who, what, when." For any control question ask **who** performs it (user, IT, service organization), **what** it does (prevents, detects, corrects), and **when** it operates (before, during or after processing).

Key takeaways

- ISC: 4 hours, 82 MCQs (two testlets of 41) and 6 TBSs (testlets of 1, 3 and 2); MCQs 60%, TBSs 40%; passing score 75.
- Areas: Information Systems and Data Management 35–45%, Security, Confidentiality and Privacy 35–45%, SOC Engagements 15–25%.
- The skill mix is weighted toward remembering and understanding (55–65%), so precise definitions and framework structures matter.
- This guide uses COSO 2013, COBIT 2019, NIST CSF 2.0, SP 800-61 Rev. 3, CIS v8.1, ISO/IEC 27001:2022, PCI DSS v4.0.1 and the SSAE 18/21 SOC standards.

1. IT Infrastructure and Cloud Computing

Every control you will study in ISC sits on top of infrastructure: the hardware, networks, operating systems and platforms that applications run on. This chapter covers the building blocks of infrastructure, the NIST definition of cloud computing, the service and deployment models, the shared responsibility model, and the availability arithmetic behind service level agreements (SLAs).

Infrastructure building blocks

Layer	What it includes	Control relevance
Physical	Data centers, servers, storage arrays, power, cooling	Physical access, environmental controls
Network	Routers, switches, firewalls, Wi-Fi, WAN links	Segmentation, perimeter security, monitoring
Compute	Physical servers, virtual machines, containers	Hardening, patching, capacity
Operating system	Windows, Linux, mainframe OS	Privileged accounts, configuration baselines
Middleware and database	Web servers, application servers, DBMS	Database administrator access, logging
Application	ERP, payroll, billing, SaaS tools	Application controls, user access

A useful way to think about infrastructure on the exam: the **lower** a layer sits, the more applications depend on it. A weakness at the operating system or database level (for example, too many people with administrator rights) can undermine every application control that runs above it.

Common infrastructure terms

- **Client-server:** clients (user devices) request services from servers that hold shared data and processing.
- **Mainframe:** a large centralized computer for very high-volume transaction processing; still common in banking and insurance.
- **Edge computing:** processing near where data is created (a store, a factory floor, a sensor) to reduce delay.

- **Internet of Things (IoT):** networked sensors and devices; they expand the attack surface and are often hard to patch.
- **Network types:** LAN (one site), WAN (connects sites), VPN (encrypted tunnel across a public network), SD-WAN (software-managed WAN).
- **Storage types:** direct-attached storage, network-attached storage (NAS, file level) and storage area networks (SAN, block level).

Virtualization and containers

Virtualization lets one physical machine run several isolated virtual machines (VMs). The software that does this is a **hypervisor**.

Item	Runs on	Typical use
Type 1 (bare-metal) hypervisor	Directly on the hardware	Data centers and cloud providers
Type 2 (hosted) hypervisor	On top of a host operating system	Desktop testing and labs
Container	Shares the host operating system kernel	Lightweight, portable application packages

Containers start faster and use fewer resources than VMs because each container does not carry a full operating system. The trade-off is weaker isolation: all containers on a host share its kernel, so a kernel flaw can affect them all.

> **Exam tip:** "Bare metal" = Type 1. If the question says the hypervisor is installed as an application on a laptop's operating system, it is Type 2.

Cloud computing — the NIST definition

NIST SP 800-145 defines cloud computing through **five essential characteristics**, **three service models** and **four deployment models**.

Essential characteristic	Meaning in plain language
On-demand self-service	The customer provisions resources without calling the provider
Broad network access	Services are reachable over the network from many device types
Resource pooling	The provider serves many customers from shared resources (multi-tenancy)
Rapid elasticity	Capacity scales up and down quickly, often automatically
Measured service	Usage is metered, so customers pay for what they use

Memory trick: "**O**ur **B**ig **R**ed **R**ocket **M**easures" — On-demand, Broad access, Resource pooling, Rapid elasticity, Measured service.

Service models

Model	Provider manages	Customer manages	Example
Infrastructure as a Service (IaaS)	Facilities, hardware, network, hypervisor	Guest operating system, middleware, applications, data	Renting virtual servers and storage
Platform as a Service (PaaS)	Everything above plus the operating system and runtime	Its own code, configuration and data	Managed application platforms, managed databases
Software as a Service (SaaS)	Almost everything, including the application	Configuration, users, access settings and its data	Hosted CRM, payroll or e-mail

The key question is usually **who patches the operating system**. In IaaS, the customer does. In PaaS and SaaS, the provider does.

Memory trick: "I-P-S" — Infrastructure, Platform, Software. Each step up, the provider manages more and you manage less.

Deployment models

Model	Who uses it
Public cloud	Open to the general public; owned by a provider
Private cloud	Used by a single organization (on or off premises)
Community cloud	Shared by organizations with common concerns (for example, regulators or a group of hospitals)
Hybrid cloud	A combination of two or more distinct models bound together so data and applications can move between them

Watch out: A combination of a private cloud and a community cloud is **hybrid**, not community. Any mix of two or more models is hybrid.

The shared responsibility model

In every cloud model, responsibility is split between the provider and the customer. The split moves with the service model, but two points never change:

- The **provider always** secures the physical facilities, hardware and the underlying virtualization layer.
- The **customer always** owns its data, its user identities and the access decisions it makes (who can see what, whether storage is public, whether MFA is enabled).

Responsibility	IaaS	PaaS	SaaS
Physical data center	Provider	Provider	Provider
Hypervisor / host	Provider	Provider	Provider
Guest operating system patching	**Customer**	Provider	Provider
Application code	Customer	Customer	Provider
Application configuration and user access	Customer	Customer	**Customer**
Data classification and protection	**Customer**	**Customer**	**Customer**

Example: A company stores customer files in a cloud storage bucket and leaves the bucket readable by anyone on the internet. A researcher finds the data. Who failed?

1. The provider operated the storage service correctly; physical and platform security were not breached.
2. The setting that made the bucket public is an access decision about the customer's own data.
3. Therefore the **customer** is responsible. Moving to the cloud did not transfer responsibility for configuration.

Cloud benefits and risks

Benefits	Risks
Shift from capital expenditure (buying servers) to operating expenditure (paying per use)	Vendor lock-in and difficult exit
Scalability and elasticity	Data residency and cross-border transfer rules
Faster deployment	Misconfiguration by the customer
Provider-scale resilience and security tools	Reduced visibility; reliance on the provider's controls
	Multi-tenancy (another tenant's problem could affect you)

Because the customer relies on the provider's controls, the usual assurance is a **SOC 2 Type 2 report** on the provider (see Chapters 14–15), plus the customer's own controls over configuration and access, and a contract that covers data ownership, breach notification, audit rights, data return and deletion at exit.

> **Exam tip:** When a question asks how a cloud customer gains comfort over the provider's controls, look for "obtain and review the provider's SOC 2 Type 2 report, including complementary user entity controls."

Availability and service level agreements

An SLA is the provider's promise about service quality, most often **availability** (uptime). Availability is a percentage of total time:

Availability = (Total time − Downtime) ÷ Total time

To convert an availability promise into allowed downtime, multiply the period in minutes by (1 − availability).

Availability	Allowed downtime per 30-day month	Allowed downtime per 365-day year
99% ("two nines")	432 minutes (7.2 hours)	5,256 minutes (87.6 hours)
99.9% ("three nines")	43.2 minutes	525.6 minutes (8.76 hours)
99.95%	21.6 minutes	262.8 minutes
99.99% ("four nines")	4.32 minutes	52.56 minutes
99.999% ("five nines")	0.432 minutes (about 26 seconds)	about 5.26 minutes

> **Memory trick:** Each extra "nine" divides allowed downtime by ten.

> **Example:** A 30-day month has 30 × 24 × 60 = **43,200 minutes**. The SLA promises 99.9% availability.
>
> 1. Allowed downtime = 43,200 × (1 − 0.999) = 43,200 × 0.001 = **43.2 minutes**.
> 2. If availability were 99.95%: 43,200 × 0.0005 = **21.6 minutes**.

Example: A SaaS contract costs $18,000 per month. It promises 99.9% monthly availability and grants a service credit of 10% of the monthly fee if availability falls below that level. Announced maintenance is excluded. During a 30-day month the service was down for 95 minutes, of which 35 minutes were announced maintenance.

1. Total minutes = 43,200.
2. Countable downtime = 95 − 35 = **60 minutes**.
3. Measured availability = (43,200 − 60) ÷ 43,200 = **99.861%** (rounded).
4. 99.861% is below 99.9%, so the credit applies: 10% × $18,000 = **$1,800**.

If the customer had forgotten to exclude maintenance, it would have computed (43,200 − 95) ÷ 43,200 = 99.780% — the same conclusion here, but in other cases the exclusion changes the answer. Always read what the contract excludes.

Composite availability

Systems usually depend on several components.

- **In series** (all must work): multiply the availabilities. The whole is **less** available than its weakest part.
- **In parallel** (redundant, any one is enough): availability = 1 − (product of the failure probabilities). Redundancy makes the whole **more** available than any single part.

Example: An order system needs a web server (99.9%), an application server (99.95%) and a database (99.8%), all in series.

1. Composite = 0.999 × 0.9995 × 0.998 = **0.99650**, or about **99.65%**.
2. Allowed downtime at that level in a 30-day month = 43,200 × (1 − 0.996503) = about **151 minutes** — far more than any one component alone.

Now replace the single web server with two redundant web servers, each 99% available, in front of a 99.9% database:

1. Web tier (parallel) = 1 − (0.01 × 0.01) = 1 − 0.0001 = **99.99%**.
2. Whole system (web tier in series with the database) = 0.9999 × 0.999 = **99.89%** (rounded).

Watch out: Slipping a decimal is the most common SLA error. 99.9% means a failure fraction of **0.001**, not 0.01.

Other infrastructure concepts that appear on the exam

Term	Meaning
Load balancer	Spreads traffic across servers; improves capacity and availability
Content delivery network (CDN)	Caches content near users; also absorbs some denial-of-service traffic
Infrastructure as code (IaC)	Infrastructure defined in version-controlled scripts — so changes can go through change management
Serverless / functions as a service	Provider runs code on demand; customer manages only the function and its permissions
Capacity management	Monitoring usage trends so resources are added before performance degrades
End-user computing (EUC)	Spreadsheets and small tools built by users; need version control, access and review controls when used for financial reporting

Exam tip: Infrastructure as code is a control **opportunity**: once servers are defined in scripts, configuration changes can be peer-reviewed, tested and logged like any other code change.

Common traps

- Thinking the cloud provider is responsible for a customer's misconfiguration — it is not.
- Saying the customer patches the operating system in PaaS or SaaS — the provider does.
- Calling a private-plus-community arrangement "community" — any combination is hybrid.
- Forgetting maintenance exclusions in an SLA computation.
- Assuming redundancy always helps in a series chain — it helps only the tier where it is added.

Key takeaways

- NIST SP 800-145: 5 characteristics (on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service), 3 service models, 4 deployment models.
- IaaS: customer manages the guest OS and up. PaaS: customer manages code and data. SaaS: customer manages configuration, users and data.
- The customer always owns its data, identities and access decisions.
- Allowed downtime = minutes in period × (1 − availability); 99.9% of a 30-day month = 43.2 minutes.

- Series: multiply availabilities. Parallel: 1 − product of failure rates.
- Assurance over a cloud provider usually comes from a SOC 2 Type 2 report plus the customer's complementary controls.

2. Enterprise and Accounting Information Systems

An accounting information system (AIS) captures business events, processes them into records and produces reports. In most organizations today that system is an **enterprise resource planning (ERP)** system that integrates many functions on one database. This chapter covers ERP and other enterprise systems, the transaction cycles, processing methods, files and documents, data interchange, and newer technologies such as robotic process automation (RPA), artificial intelligence and blockchain — with the controls each needs.

Enterprise systems

System	Purpose
ERP	Integrates finance, purchasing, sales, inventory, manufacturing and HR on one shared database
Customer relationship management (CRM)	Manages customer contacts, sales pipeline and service history
Supply chain management (SCM)	Plans and tracks the flow of goods from suppliers to customers
Human capital management (HCM)	Employee records, payroll, benefits and time
Business intelligence (BI)	Reporting and dashboards drawn from a data warehouse
Governance, risk and compliance (GRC)	Tracks risks, controls, policies and testing

Why integration matters for controls

In an integrated ERP, one entry updates every module at once. A sales order reduces available inventory, creates a receivable when shipped and posts revenue to the general ledger — without re-keying. That brings real benefits and real risks:

Benefit	Risk
One version of the data; fewer re-entry errors	One error spreads everywhere instantly
Real-time information for decisions	Access rights are complex; it is easy to grant incompatible permissions
Built-in workflow and approvals	Configuration settings (tolerances, approval limits) become key controls
Easier consolidation and reporting	Implementation is expensive and risky

> **Watch out:** Integration does **not** remove the need for reconciliations and reviews. Interfaces to systems outside the ERP, manual journal entries and configuration changes still need controls.

Configuration as a control

ERP systems let the organization set parameters — for example, a price tolerance for invoice matching, required fields, credit limits or approval thresholds. These **configurable controls** are automated application controls. Once set, they operate consistently, so the key risk shifts to **who can change the configuration** (an IT general control, Chapter 3).

Transaction cycles

Organizing controls by cycle helps you identify the risk in a question. Ask: who is the other party?

Cycle	Other party	Main activities	Key documents and records
Revenue	Customer	Order entry, credit approval, shipping, billing, cash receipts	Sales order, shipping document, invoice, remittance advice
Expenditure	Vendor	Requisition, purchasing, receiving, invoice approval, payment	Purchase requisition, purchase order, receiving report, vendor invoice, check or payment file
Payroll / HR	Employee	Hiring, time recording, payroll processing, payment	Personnel file, time records, payroll register
Production	(internal)	Planning, material issues, labor, overhead, completion	Bill of materials, production order, move ticket
Financing	Lenders and investors	Borrowing, issuing stock, paying interest and dividends	Loan agreements, board minutes
General ledger and reporting	(internal)	Posting, adjusting, closing, reporting	Journal entries, trial balance, financial statements

Segregation of duties in the cycles

The classic rule separates four functions so that no one person can both commit and conceal an error or fraud:

1. **Authorization** of transactions
2. **Recording** of transactions
3. **Custody** of assets
4. **Reconciliation** (comparing records to assets)

> **Example:** In the expenditure cycle, a purchasing clerk who can also add new vendors to the vendor master file and approve invoices could create a fictitious vendor and pay it. The incompatible pair is **vendor master maintenance** and **invoice approval or payment**.

> **Memory trick:** "**ARC-R**" — Authorize, Record, Custody, Reconcile. Any two in one person's hands is a warning sign.

The three-way match

Before paying a vendor, the system compares three documents:

1. **Purchase order** (what was ordered, at what price)
2. **Receiving report** (what actually arrived)
3. **Vendor invoice** (what the vendor billed)

The amount to pay is the **lowest** of the quantities ordered, received and billed, at the **purchase order price** (unless an approved price change exists). Differences outside the configured tolerance are held for review.

Example: A purchase order is for 500 units at $12.00. The receiving report shows 480 units. The vendor invoices 500 units at $12.50.

1. Quantity to pay = lowest of 500 ordered, 480 received, 500 billed = **480 units**.
2. Price to pay = PO price = **$12.00** (the $12.50 was never approved).
3. Amount payable = 480 × $12.00 = **$5,760**.
4. Invoice total = 500 × $12.50 = $6,250, so the system holds **$490** of the invoice as an exception.

Memory trick: "PO-RR-INV" — order, receive, invoice: the three legs of the match.

Watch out: Do not pay the invoice total. The invoice is the vendor's claim; the purchase order and receiving report are the company's evidence.

Processing methods

Method	How it works	Strengths	Weaknesses
Batch processing	Transactions collected and processed together at intervals (for example, nightly payroll)	Efficient; batch totals give strong control	Master files are out of date between runs
Online real-time processing	Each transaction updates the master file immediately	Current information	Harder to control individually; needs strong input edits and logs
Online batch	Entered online, edited immediately, but posted in batches	Combines immediate editing with batch efficiency	Some delay before posting

Batch control totals

Batch totals prove that every record in a batch was processed completely and accurately. They are computed when the batch is prepared and compared after processing.

Total	What it adds	Detects
Record count	The number of documents or records	Lost or duplicated records
Financial total	A field with real meaning, such as dollar amounts	Lost records and changed amounts
Hash total	A field with no arithmetic meaning, such as vendor or account numbers	Lost records and changed identifiers

Example: A batch of four vendor invoices is prepared:

Vendor no.	Date	Amount
40117	11/3/2025	$1,250.00
40233	11/3/2025	$980.50
40391	11/4/2025	$4,310.25
40402	11/4/2025	$659.25

1. Record count = **4**.
2. Financial total = 1,250.00 + 980.50 + 4,310.25 + 659.25 = **$7,200.00**.
3. Hash total of vendor numbers = 40117 + 40233 + 40391 + 40402 = **161,143**.

If a clerk keyed vendor 40391 as 40319, the record count and financial total would still agree, but the hash total would be off by 72. Only the **hash total** catches that error.

Memory trick: "Hash total = nonsense total." It adds numbers that mean nothing when added.

Files and records

Term	Meaning
Master file	Standing data that changes rarely: customers, vendors, employees, prices, chart of accounts
Transaction file	Individual events for a period: invoices, receipts, time records
Reference (table) file	Lookup values such as tax rates or currency codes
Audit trail	The chain of records that lets a transaction be traced from source to report and back

Master files deserve special attention because a single unauthorized change affects every later transaction. The highest-risk example is a change to a vendor's **bank account number**, which can redirect every future payment. Strong controls include restricted access to master data, independent review of a change report, and a call-back to the vendor at a number already on file before activating new bank details.

Data interchange and integration

- **Electronic data interchange (EDI):** structured business documents (purchase orders, invoices, advance ship notices) exchanged computer-to-computer in standard formats. Controls: trading partner agreements, authentication of partners, acknowledgments, control totals per transmission, and encryption in transit.
- **Application programming interfaces (APIs):** programs exchanging data directly. Controls: authentication tokens, rate limits, input validation, logging and change control over the interface.
- **Interfaces and middleware:** each interface between systems is a place where data can be lost or changed, so reconcile record counts and totals between the sending and receiving systems.

System conversion approaches

When a new system replaces an old one, the organization chooses how to cut over:

Approach	Description	Risk
Parallel	Old and new systems run together; results compared	**Lowest** risk; highest cost
Pilot	New system used at one location first	Low–moderate
Phased	New system introduced module by module	Moderate
Direct (plunge, "big bang")	Old system switched off and new one on at once	**Highest** risk

Conversion controls include reconciling record counts and control totals of migrated data, reviewing exception reports, and having business owners sign off on converted balances.

Emerging technologies and their controls

Technology	What it is	Key controls
Robotic process automation (RPA)	Software "bots" that mimic user steps in applications	Each bot gets a unique, least-privilege service account; bot changes go through change management; output reviewed; owner assigned
Artificial intelligence and machine learning	Models that classify, predict or generate content from data	Model governance, training-data quality, validation and monitoring for drift, human review of high-impact decisions, explainability
Blockchain / distributed ledger	Shared append-only ledger in which each block is cryptographically linked to the previous one	Controls over who can submit transactions, private key protection, smart contract code review, and off-chain validation
Internet of Things	Sensors feeding data into systems (for example, inventory weights)	Device authentication, patching, network segmentation, data validation

Watch out: A blockchain proves that a record has **not been altered** since it was added. It does **not** prove the transaction was valid, authorized or correctly valued. A fraudulent transaction recorded on a blockchain is permanently and faithfully recorded fraud.

Exam tip: RPA bots do exactly what they are told, very fast. A bot with a shared or over-privileged account destroys accountability and can make the same error thousands of times. Look for answers about unique bot credentials and change control.

Common traps

- Paying the invoice amount instead of the three-way-matched amount.
- Confusing a hash total (meaningless field) with a financial total (meaningful amount).
- Assuming ERP integration removes the need for reconciliations.
- Calling direct cutover low-risk because it is fast and cheap.
- Treating a blockchain entry as proof of validity.

Key takeaways

- ERP integrates functions on one database: errors and unauthorized changes spread instantly, so configuration and access controls are critical.
- Identify the cycle by the outside party: customer (revenue), vendor (expenditure), employee (payroll), lender or investor (financing).

- Segregate authorization, recording, custody and reconciliation.
- Three-way match: pay the lowest of ordered, received and billed quantities at the PO price.
- Batch totals: record count, financial total, hash total — only the hash total catches a changed identifier.
- Parallel conversion is safest; direct cutover is riskiest.
- RPA, AI and blockchain each need their own governance, access and change controls.

3. IT General Controls and Application Controls

This chapter is the backbone of Area I. It explains how COSO treats technology controls, the difference between IT general controls (ITGCs) and application controls, the main input, processing and output controls, how controls are classified, and why ITGC weaknesses matter so much to auditors.

COSO 2013 and technology

The COSO *Internal Control — Integrated Framework* (2013) has **five components** and **17 principles**.

Component	Principles	Focus
Control environment	1–5	Integrity and ethics, board oversight, structure, competence, accountability
Risk assessment	6–9	Objectives, risk identification, fraud risk, significant change
Control activities	10–12	Select and develop controls; **general controls over technology (Principle 11)**; policies and procedures
Information and communication	13–15	Quality information, internal and external communication
Monitoring activities	16–17	Ongoing and separate evaluations; communicating deficiencies

> **Memory trick:** "**C**ats **R**arely **C**atch **I**nvisible **M**ice" — Control environment, Risk assessment, Control activities, Information and communication, Monitoring.

Principle 11 says the organization selects and develops general control activities over technology to support the achievement of objectives. It names technology infrastructure, security management, and technology acquisition, development and maintenance.

IT general controls versus application controls

	IT general controls (ITGCs)	Application controls
Scope	Apply across many systems — the environment	Apply to specific transactions in a specific application
Examples	Logical access, change management, program development, computer operations (backup, job scheduling, incident handling)	Input edit checks, three-way match, automated calculations, interface reconciliations, approval workflows
Why they matter	They make sure application controls keep working as designed and cannot be bypassed	They prevent or detect errors in transactions

The four ITGC domains

1. **Access to programs and data** — user provisioning, termination, periodic access reviews, privileged access, password and authentication settings (Chapter 10).
2. **Program changes** — changes are authorized, tested, approved and migrated by someone independent of the developer (Chapter 4).
3. **Program development** — new systems are acquired or built with proper design, testing, data conversion and sign-off (Chapter 4).
4. **Computer operations** — job scheduling and monitoring, backup and recovery, problem and incident management (Chapter 5).

Why ITGCs come first

An automated application control does the same thing every time — as long as nobody changes the program and nobody can bypass it. ITGCs over **change management** and **access** are what make that "as long as" true. This leads to a key audit concept:

> **Exam tip:** If ITGCs over change management and access are effective, an auditor can test an automated application control with a **sample of one** (a single test of each relevant scenario) and, in later periods, may rely on a **benchmarking** strategy — confirming the program has not changed instead of retesting fully. If ITGCs are **not** effective, reliance on automated controls is reduced and more substantive testing is needed.

Control classifications

By timing / purpose

Type	What it does	Examples
Preventive	Stops an error or irregularity before it happens	Segregation of duties, approval before payment, input edit checks, access restrictions, encryption
Detective	Finds an error after it happened	Reconciliations, exception reports, reviews of logs, variance analysis, CCTV review
Corrective	Fixes the problem and reduces its impact	Restoring from backup, error correction and resubmission, patching after an incident
Directive	Encourages desired behavior	Policies, procedures, training, signage
Deterrent	Discourages violations	Warning banners, visible cameras
Compensating	Reduces risk when a primary control is missing	Supervisory review where duties cannot be segregated in a small office

Watch out: A review of a report **after** transactions are posted is **detective**, even if it is done by a senior person. "Preventive" requires that the control act before the error is recorded.

By nature

Nature	Description	Testing consideration
Automated	Performed entirely by the system	Test once per scenario if ITGCs are effective
IT-dependent manual	A person acts on system output (for example, reviewing an exception report)	Test the manual review **and** the completeness and accuracy of the report (information produced by the entity, IPE)
Manual	Performed without system support	Sample size grows with frequency

Example: A controller reviews a system-generated report of all journal entries over $50,000 and signs off monthly. The auditor tests a sample of months for evidence of review. Is that enough?

1. The control is **IT-dependent manual** — the review relies on the report.
2. If the report silently omitted some entries, the review would miss them.
3. The auditor must also test the report (its logic and parameters, or reperform it from the ledger) for **completeness and accuracy**. Only then is the review effective.

Input controls (edit checks)

Input controls make sure data entered into the system is authorized, complete and accurate.

Edit check	Tests whether...	Example
Field (format) check	Characters are the right type	Letters rejected in a numeric amount field
Missing data / completeness check	Required fields are filled	Invoice cannot be saved without a vendor number
Limit check	A value is below a set maximum	Hours worked per week ≤ 80
Range check	A value lies between a minimum and maximum	Discount between 0% and 15%
Validity check	A value matches an authorized list or master file	Vendor number must exist in the vendor master
Reasonableness check	A value makes sense in context	Order quantity far above the customer's usual pattern flagged
Sign check	A value has the correct sign	Quantity received cannot be negative
Sequence check	Documents are in order with no gaps or duplicates	Check numbers are consecutive
Check digit	An identifier was keyed correctly	Account number's last digit verifies the others
Closed-loop verification	Entered ID is confirmed by showing related data	After typing a customer number, the screen shows the customer's name for confirmation
Duplicate check	The same item is not entered twice	Same vendor + invoice number rejected
Zero-balance check	Debits equal credits	Journal entry cannot post unless it balances

Check digits — the Luhn (mod 10) method

A check digit is an extra digit calculated from the other digits of an identifier. When the number is entered, the system recalculates it. The Luhn method (used for payment cards and many account numbers) catches all single-digit errors and most transpositions of adjacent digits.

To compute the check digit: starting from the rightmost digit of the base number and moving left, double every second digit (the rightmost digit is doubled first). If a doubled value exceeds 9, subtract 9. Add all the digits. The check digit is the amount needed to round the sum up to the next multiple of 10.

Example: Compute the Luhn check digit for account base number **5419382**.

Digit (left to right)	**5**	**4**	**1**	**9**	**3**	**8**	**2**
Doubled?	Yes	No	Yes	No	Yes	No	Yes
Value used	10 − 9 = 1	4	2	9	6	8	4

1. Sum = 1 + 4 + 2 + 9 + 6 + 8 + 4 = **34**.
2. Next multiple of 10 is 40, so the check digit = 40 − 34 = **6**.
3. Full account number = **54193826**.

Now a clerk transposes two digits and keys 54**91**3826. The system recomputes the Luhn sum for the full number and gets **39**, which is not a multiple of 10, so the entry is rejected.

Watch out: A **validity check** confirms a number exists in the master file — but a transposition might produce another valid number. Catching transposed digits is the job of a **check digit**.

Processing controls

Processing controls make sure transactions are processed completely and accurately and that files are updated correctly.

- **Run-to-run totals:** control totals carried from one processing step to the next must agree.
- **Batch totals** (Chapter 2): record count, financial total, hash total.
- **Automated calculations:** depreciation, interest, payroll taxes computed by the system.
- **Interface reconciliations:** record counts and totals sent equal those received.
- **Error logs and suspense handling:** rejected items are logged, investigated, corrected and resubmitted.
- **Concurrency controls:** database locking stops two users from updating the same record at once.
- **Job scheduling controls:** jobs run in the right order, and failures are flagged and resolved.

Output controls

Output controls make sure results are complete, accurate and reach only authorized recipients.

- Reconcile output totals to input and processing totals.
- Review exception and error reports.
- Distribute reports according to a distribution list; secure printing; mark confidential output.
- Retain output per policy; shred or securely delete sensitive output.
- User review of output for reasonableness.

IT segregation of duties

Within IT, the key separations are:

Function	Should not also...
Developers / programmers	Migrate their own code to production, or have update access to production data
Database administrators	Approve their own changes, or initiate business transactions
Security administrators	Perform business transactions with the accounts they create
Computer operators	Change programs
Users	Change programs or system configuration

> **Exam tip:** The single most tested IT segregation rule: **developers must not move their own changes into production.** A separate person or an automated pipeline with an independent approval gate performs the migration.

Common traps

- Calling an after-the-fact review "preventive" — it is detective.
- Choosing a validity check to catch transposed digits — use a check digit.
- Testing an IT-dependent manual control without testing the report it relies on.
- Forgetting that ITGC weaknesses reduce reliance on every automated control that depends on them.
- Treating a policy (directive) as if it operated by itself; policies need supporting control activities.

Key takeaways

- COSO 2013: 5 components, 17 principles; Principle 11 covers general controls over technology.
- ITGCs (access, change, development, operations) support the reliability of application controls.
- Automated controls can be tested once per scenario if change and access ITGCs are effective.
- IT-dependent manual controls require testing the completeness and accuracy of the report (IPE).
- Know the edit checks: limit, range, validity, reasonableness, sign, sequence, check digit, closed-loop, completeness, duplicate.
- Luhn: double every second digit from the right, subtract 9 from results over 9, sum, round up to the next multiple of 10.

4. Change Management and the SDLC

Systems change constantly: patches, new features, configuration tweaks, new interfaces and whole new systems. Each change is a chance for an error — or a deliberate manipulation — to slip into production. Change management makes sure only **authorized, tested and approved** changes reach production. The system development life cycle (SDLC) applies the same discipline to building or buying new systems.

The SDLC phases

Phase	Key question	Main outputs and controls
1. Planning and feasibility	Should we do this?	Business case; technical, economic, operational and schedule feasibility; steering committee approval
2. Analysis (requirements)	**What** must the system do?	User requirements, including control and security requirements; current-process analysis
3. Design	**How** will it do it?	Logical and physical design, data structures, interfaces, built-in controls
4. Development (or acquisition)	Build or buy it	Coding standards, peer review, vendor selection (RFP), contract review
5. Testing	Does it work?	Unit, integration, system, regression and user acceptance testing
6. Implementation	Put it into use	Data conversion, training, cutover approach, go-live approval
7. Operation and maintenance	Keep it working	Change management, monitoring, post-implementation review

Memory trick: Analysis = **what**; Design = **how**. If the question describes deciding what the users need, it is analysis. If it describes table layouts, screens and interfaces, it is design.

Exam tip: The best time to build controls into a system is during **analysis and design**, not after go-live. Retrofitting controls is more expensive and less effective. An auditor or control specialist adds the most value by participating early, while keeping independence (advising, not approving or building).

Build versus buy

When buying a packaged system, the organization issues a **request for proposal (RFP)**, evaluates vendors against requirements, checks references and financial stability, reviews the vendor's SOC reports, and negotiates the contract (support, upgrades, source code escrow, data ownership, exit terms). **Source code escrow** places the vendor's code with a third party so the customer can maintain the software if the vendor fails.

Development methodologies

Method	Characteristics	Control considerations
Waterfall	Sequential phases with sign-off at each gate	Clear documentation; inflexible to change
Agile (e.g., Scrum)	Short iterations (sprints), working software each cycle, product owner prioritizes a backlog	Documentation lighter; controls built into the "definition of done," peer review and backlog approval
DevOps / CI/CD	Development and operations combined; continuous integration and automated deployment pipelines	Pipeline gates (automated tests, code review approval) replace manual hand-offs; pipeline configuration itself must be change-controlled
Rapid application development / prototyping	Quick working models refined with users	Risk that prototypes go live without full testing and controls

Watch out: Agile and DevOps do not remove segregation of duties. In a well-controlled pipeline, a developer's code cannot be deployed until **another person** approves the pull request and automated tests pass. The developer cannot override those gates.

Testing levels

Test	Who	What it checks
Unit test	Developer	One program or component works by itself
Integration test	Developers / testers	Components work together, including interfaces
System test	Testers (IT)	The whole system meets technical and functional requirements, including performance and security
Regression test	Testers / automated suite	Previously working functions still work after a change
User acceptance test (UAT)	Business users	The system meets business needs; formal **sign-off** by the business

Other testing terms:

- **Stress (load) testing:** behavior under heavy volume.
- **Security testing:** vulnerability scans and penetration tests before go-live.
- **Black-box testing:** tests inputs and outputs without looking at the code; **white-box testing:** tests internal logic and paths.

> **Watch out:** Unit testing is done by the **developer** on one component. UAT is done by **business users** on the whole system. Do not confuse them.

Test environments and test data

Development, test and production should be **separate environments**. Production data copied into test environments must be **masked** (or synthetic data used), because test environments usually have weaker access controls. Developers should have no update access to production.

The change management process

A normal change follows these steps:

1. **Request** — logged in a ticketing system with a business reason.
2. **Authorization** — approved by the business owner (and change advisory board for significant changes) **before** development begins.
3. **Development** — in a development environment, under version control.
4. **Testing** — by someone other than the developer; regression testing as needed; documented UAT sign-off.
5. **Approval to deploy** — final approval based on test results.
6. **Migration to production** — by someone **independent of the developer** (or an automated pipeline with approval gates).
7. **Post-implementation review** — confirm the change works and no unexpected effects occurred.

Change type	Description	Control rule
Standard	Pre-approved, low-risk, repeatable (for example, a routine password reset or a template-based patch)	Follows a pre-authorized procedure
Normal	Typical change	Full process: approval **before** development and deployment
Emergency	Urgent fix to restore service	May be deployed first, but must be **logged**, **documented** and **approved retroactively** promptly; emergency ("firefighter") access is temporary and its use reviewed

Watch out: After-the-fact approval is acceptable **only** for genuine emergency changes, and even then it must be timely and documented. A normal change approved after deployment is a control deviation.

Testing change management as an auditor

The direction of the sample matters.

- Sampling **from change tickets** and tracing to production tests whether ticketed changes followed the process. It cannot find changes that **never had a ticket**.
- Sampling **from the production environment** — the migration log, system-generated change log or a comparison of program versions — and tracing back to tickets tests **completeness**: did any change bypass the process?

Example: During the year, the production migration log shows 212 deployments to the general ledger application. The change ticketing system contains 205 closed tickets for that application.

1. 212 − 205 = **7 deployments** have no matching ticket at the count level.
2. The auditor should identify those deployments and investigate each: were they emergency changes approved later, were several deployments made under one ticket, or did someone deploy unauthorized code?
3. Sampling only from the 205 tickets would never have revealed the 7 unmatched deployments. The **production log is the right population** for a completeness test.

Configuration and database changes

Changes do not have to be code. Changing an ERP tolerance, an approval limit, a tax table or a user role, or running a direct database update ("data fix"), can affect financial data just as much. These should follow change management too, and direct data changes in production should be rare, approved, logged and reviewed.

Post-implementation review

After a new system or major change has run for a while, a post-implementation review compares actual results with the business case: were the benefits achieved, are controls working, what lessons apply to future projects? It is an important **monitoring** activity.

> **Exam tip:** If the question asks which SDLC step determines whether a project delivered its expected benefits, the answer is the **post-implementation review**, not UAT. UAT happens before go-live; the review happens after.

Data conversion controls

When moving to a new system:

- Cleanse data before migration (remove duplicates, fix invalid codes).
- Reconcile **record counts** and **control totals** (for example, total receivables, number of vendors) between old and new systems.
- Review exception reports of records that failed to convert.
- Obtain business-owner sign-off on converted balances.
- Keep the old system's data accessible for a period in case questions arise.

> **Example:** The legacy receivables subledger shows 3,418 open invoices totaling $2,746,310. After conversion, the new ERP shows 3,416 open invoices totaling $2,741,880.
>
> 1. Record difference = 3,418 − 3,416 = **2 invoices** missing.
> 2. Amount difference = $2,746,310 − $2,741,880 = **$4,430** missing.
> 3. The two invoices on the conversion exception report should total $4,430. If they do, correct and load them; if not, more items are wrong and the conversion should not be signed off.

Common traps

- Accepting after-the-fact approval for a non-emergency change.
- Sampling from tickets when the objective is to find unauthorized changes.
- Confusing unit testing (developer) with UAT (business users).
- Treating configuration changes or maintenance as exempt from change control.

- Using unmasked production data in test environments.
- Letting the developer migrate code because "the team is small" without any compensating review.

Key takeaways

- SDLC: planning, analysis (what), design (how), development, testing, implementation, maintenance; controls are cheapest to build in during analysis and design.
- Normal changes: approve before development and deployment, independent testing, UAT sign-off, migration by someone other than the developer.
- Emergency changes: deploy first, then log and approve retroactively; firefighter access is temporary and reviewed.
- To find changes that bypassed the process, sample from the production change log, not from tickets.
- Mask production data used in testing; separate development, test and production environments.
- Reconcile record counts and control totals on every data conversion.

5. Business Resiliency, Backup and Recovery

Resiliency is the ability to keep critical business processes running through a disruption and to restore systems and data afterward. ISC tests the planning vocabulary (BIA, RTO, RPO, MTD), recovery site choices, backup types and restore sets, RAID, and the reliability formulas. Expect computations here.

Business continuity versus disaster recovery

Term	Focus
Business continuity plan (BCP)	Keeping **critical business processes** running during and after a disruption: people, facilities, suppliers, manual workarounds, communications
Disaster recovery plan (DRP)	Restoring **IT systems and data** after a disruption; a subset of business continuity
Crisis management / communications plan	Leadership decisions and communications to staff, customers, regulators and the media
Incident response plan	Handling security incidents (Chapter 12); may trigger the DRP

The plan should be owned by the business, approved by senior management, kept current, stored where it is reachable during a disaster (not only on the systems that failed), and **tested**.

Business impact analysis (BIA)

The BIA comes first. It identifies critical processes, the systems and resources they depend on, and the impact of losing them over time (financial, legal, reputational, safety). From the BIA the organization sets recovery objectives.

Metric	Question it answers	Drives
Recovery point objective (RPO)	How much **data** can we afford to lose, measured in time? (backward-looking from the failure)	**Backup or replication frequency**
Recovery time objective (RTO)	How quickly must the **system** be restored? (forward-looking from the failure)	**Recovery strategy** (hot, warm, cold site; replication)
Work recovery time (WRT)	After the system is back, how long to verify data and catch up on lost work?	Staffing and procedures
Maximum tolerable downtime (MTD)	The outer limit before the business suffers unacceptable harm (also called maximum tolerable period of disruption)	The ceiling: **RTO + WRT ≤ MTD**

Memory trick: Point = **P**ast (how far back you lose data). **T**ime = **T**omorrow (how long until you are running again).

Example: Backups of an order database are taken every 4 hours.

1. If failure occurs just before the next backup, all work since the last backup is lost: the **worst-case data loss is 4 hours**.
2. To meet an RPO of 4 hours, a 4-hour backup interval is just sufficient. To meet an RPO of 1 hour, backups (or replication) must run at least hourly.
3. The **average** loss would be about 2 hours, but RPO is about the worst case.

Example: A payroll process has an MTD of 8 hours. IT's recovery plan has an RTO of 6 hours, and the payroll team needs 3 hours afterward to validate data and re-enter lost transactions.

1. RTO + WRT = 6 + 3 = **9 hours**.
2. 9 hours > 8-hour MTD, so the plan **fails** the business requirement.
3. The maximum acceptable RTO = MTD − WRT = 8 − 3 = **5 hours**. IT needs a faster recovery strategy or the business must shorten its work recovery time.

Watch out: Using the **average** loss (half the backup interval) instead of the worst case is a common RPO error. And remember that RTO is not the whole story — the business also needs time to recover its work.

Recovery site strategies

Strategy	Description	Recovery speed	Cost
Cold site	Space with power and connectivity; no equipment ready	Days to weeks	Lowest
Warm site	Space with some equipment; data must be restored from backup	Hours to days	Moderate
Hot site	Fully equipped, data replicated or near current	Minutes to hours	High
Mirrored / active-active	Two live sites processing simultaneously	Near zero	Highest
Cloud-based DR (DR as a service)	Replicated images in the cloud, started on demand	Varies; often hours	Pay-as-used
Reciprocal agreement	Two organizations agree to host each other	Uncertain	Low, but often unreliable

The shorter the RTO and RPO, the more expensive the strategy. The right choice balances the cost of the strategy against the cost of downtime — **subject to meeting the MTD**.

Example: A distributor loses $25,000 of contribution margin for every hour its order system is down. It expects a site-level disaster about once every five years (annual rate 0.2). Two strategies are proposed:

	Hot site	Warm site
Annual cost	$180,000	$60,000
Expected RTO	1 hour	24 hours
Expected annual outage loss = 0.2 × RTO × $25,000	$5,000	$120,000
Total expected annual cost	**$185,000**	**$180,000**

On expected cost alone, the warm site is $5,000 cheaper. But if the BIA sets the MTD for order processing at 8 hours, a 24-hour RTO **fails** the requirement, and the hot site is the only acceptable choice. Requirements first, then cost.

Watch out: A cold site is never the answer when the RTO is short.

Backups

Backup types

Type	What it copies	Backup speed / size	Restore needs
Full	Everything	Slowest, largest	The full backup only
Incremental	Changes since the **last backup of any type**	Fastest, smallest	Last full **+ every** incremental since, in order
Differential	Changes since the **last full** backup	Grows each day	Last full **+ the latest** differential only
Snapshot	Point-in-time image, often on the same storage	Very fast	Quick rollback, but not a true backup if stored on the same system

Example: A full backup of 800 GB runs Sunday night. About 50 GB of data changes each weekday. The server fails Saturday morning, after Friday night's backup.

	Incremental strategy (Mon–Fri)	Differential strategy (Mon–Fri)
Friday night's backup size	50 GB	5 × 50 = 250 GB
Total stored for the week	800 + 5 × 50 = **1,050 GB**	800 + 50 + 100 + 150 + 200 + 250 = **1,550 GB**
Backup sets needed to restore	Sunday full + Mon, Tue, Wed, Thu, Fri incrementals = **6 sets**	Sunday full + Friday differential = **2 sets**

Incremental backups are faster and smaller to create; differential backups are faster and simpler to restore.

Memory trick: Incremental = **I** need them **all**. **D**ifferential = **D**on't need the earlier ones.

Backup principles

- **3-2-1 rule:** keep at least **3** copies of data, on **2** different types of media, with **1** copy offsite.
- **Immutable or offline (air-gapped) copies:** ransomware attacks often try to encrypt or delete backups; a copy that cannot be changed or is disconnected defeats that.
- **Geographic separation:** the offsite copy should be far enough away not to be affected by the same regional event.
- **Encrypt backups** and protect the keys separately.
- **Test restores regularly.** Only a successful restore proves a backup works.

- **Retention:** keep backups according to legal and business requirements; legal holds override normal deletion.

> **Exam tip:** "Backups were completed successfully every night" is not evidence that data can be recovered. The best evidence is a **periodic test restore** of a sample of backups.

RAID and fault tolerance

RAID (redundant array of independent disks) spreads data across several disks so that a disk failure does not stop the system.

Level	Method	Survives	Usable capacity (n disks of equal size)
RAID 0	Striping only	No disk failure	n disks (no redundancy)
RAID 1	Mirroring	1 disk failure (of a mirrored pair)	1 disk per pair (50%)
RAID 5	Striping with single parity	1 disk failure	**n − 1** disks
RAID 6	Striping with double parity	2 disk failures	**n − 2** disks
RAID 10	Mirrored pairs, then striped	1 per mirrored pair	**n ÷ 2** disks

> **Example:** A server has six 4 TB disks.
>
> | Level | Usable capacity |
> |---|---|
> | RAID 0 | 6 × 4 = 24 TB (no protection) |
> | RAID 5 | (6 − 1) × 4 = **20 TB** |
> | RAID 6 | (6 − 2) × 4 = **16 TB** |
> | RAID 10 | (6 ÷ 2) × 4 = **12 TB** |

> **Watch out: RAID is not a backup.** If a user deletes a file, or ransomware encrypts it, RAID faithfully copies the deletion or encryption to every disk. RAID protects against hardware failure, not data loss.

Other fault-tolerance tools

- **Uninterruptible power supply (UPS):** batteries bridge short outages and allow orderly shutdown; **generators** handle longer outages.
- **Redundant network links and power feeds.**
- **Clustering and load balancing:** if one server fails, another takes over (failover).
- **Replication:** synchronous (no data loss, needs short distance) or asynchronous (small lag, works over long distances).

Reliability formulas

Term	Meaning
MTBF — mean time between failures	Average running time between failures (higher is better)
MTTR — mean time to repair (restore)	Average time to fix a failure (lower is better)
Availability	MTBF ÷ (MTBF + MTTR)

Example: A storage controller runs on average 2,000 hours between failures, and each failure takes 4 hours to repair.

1. Availability = 2,000 ÷ (2,000 + 4) = 2,000 ÷ 2,004 = **0.99800**, or about **99.80%**.
2. Over an 8,760-hour year, expected downtime ≈ 8,760 × (1 − 0.998004) ≈ **17.5 hours**.
3. If a hot-swap spare cut MTTR to 0.5 hours with MTBF of 1,500 hours, availability = 1,500 ÷ 1,500.5 = **99.97%** — reducing repair time can matter more than improving reliability.

Testing the plan

Test type	Description	Disruption
Checklist (desk check) review	Participants review the plan for completeness	None
Tabletop exercise (structured walk-through)	Team talks through a scenario step by step	None
Simulation	Practice a scenario using some real resources	Low
Parallel test	Recovery systems brought up and run alongside production	Low–moderate
Full interruption test	Production actually shut down and moved to the recovery site	Highest; rarely done

After every test or real event, document lessons learned and update the plan.

Common traps

- Using the average instead of the worst case for RPO.
- Forgetting work recovery time: RTO alone must be compared with MTD **after** adding WRT.
- Treating RAID or same-storage snapshots as backups.
- Choosing a cold site for a short RTO.
- Counting only the last incremental when restoring — you need all of them since the full backup.
- Accepting "backup job succeeded" logs as proof of recoverability.

Key takeaways

- BIA first: it sets RTO, RPO and MTD for each critical process.
- RPO = maximum data loss → backup frequency; worst-case loss = backup interval.
- RTO = maximum time to restore → recovery strategy; RTO + WRT ≤ MTD.
- Incremental restore = last full + all incrementals since; differential restore = last full + latest differential.
- RAID 5 = n − 1, RAID 6 = n − 2, RAID 10 = n ÷ 2 usable disks; RAID is not a backup.
- 3-2-1 rule plus immutable or offline copies; only tested restores prove recovery.
- Availability = MTBF ÷ (MTBF + MTTR).

6. Data Management, Integrity and Analytics

Accountants increasingly work with data rather than documents. This chapter covers the data life cycle, governance roles, data quality dimensions, data storage architectures, retention and destruction, and the analytics techniques the exam expects you to recognize and apply — including Benford's law and z-scores.

The data life cycle

Stage	Key controls
Create / collect	Collect only what is needed (data minimization); input edits; consent where required
Store	Classification, access control, encryption, backups
Use / process	Least-privilege access, logging, approved purposes only
Share / transfer	Agreements with recipients, encryption in transit, data loss prevention (DLP) tools
Archive	Retention schedule, protected long-term storage, retrievability
Destroy	Secure, documented destruction when retention ends (unless a legal hold applies)

Data governance roles

Role	Responsibility
Data owner	A business manager **accountable** for the data: decides classification, who gets access, and retention
Data steward	Defines data meanings and quality rules; monitors and improves quality
Data custodian	IT staff who **operate** the controls: storage, backups, access provisioning as approved by the owner
Data user	Uses data for approved purposes
Chief data officer / data governance council	Sets enterprise data policy and resolves cross-functional issues

> **Memory trick:** The **owner** decides, the **steward** defines, the **custodian** does.

Watch out: IT is usually the **custodian**, not the owner. If the question asks who should approve a user's access to payroll data, the answer is the payroll data owner (a business manager), not the database administrator.

Data classification

A typical scheme has levels such as **public**, **internal**, **confidential** and **restricted** (or highly confidential). Classification drives handling rules: who may access, whether encryption is required, how it may be shared and how it is destroyed. Personal information, payment card data, health information and trade secrets usually sit at the top levels.

Data quality dimensions

Dimension	Question	Example of failure
Accuracy	Does the value reflect reality?	A customer's address is valid but wrong
Completeness	Are required values present?	Tax ID missing on vendor records
Validity	Does the value follow the defined format or rules?	Date entered as "31/02/2025"
Consistency	Do values agree across systems and records?	Customer marked active in CRM, inactive in ERP
Timeliness	Is the data current enough for its use?	Inventory counts updated weekly for a daily decision
Uniqueness	Is each real-world item recorded once?	The same vendor set up twice
Integrity	Are relationships between data maintained?	An invoice points to a customer that does not exist

Watch out: Validity is about format and rules; **accuracy** is about truth. A correctly formatted but wrong bank account number is valid and inaccurate.

Example: A vendor master file has 12,500 records. 375 records have no tax ID and 50 records are duplicates of other vendors.

1. Completeness of the tax ID field = (12,500 − 375) ÷ 12,500 = **97.0%**.
2. Uniqueness = (12,500 − 50) ÷ 12,500 = **99.6%**.
3. Duplicate vendors are a fraud and error risk (duplicate payments), so they should be merged or blocked, and a duplicate check added at entry.

Data storage architectures

Architecture	Characteristics	Typical use
Operational (transactional) database	Normalized, optimized for many small reads and writes (OLTP)	ERP, order entry
Data warehouse	Integrated, cleaned, historical data; structure defined before loading (**schema-on-write**); optimized for analysis (OLAP)	Enterprise reporting and BI
Data mart	Subject- or department-specific subset of a warehouse	Sales analytics for one division
Data lake	Raw data of any type stored in native format; structure applied when read (**schema-on-read**)	Data science, large varied data sets
Data lakehouse	Combines lake storage with warehouse-style management	Mixed analytics

Data moves into a warehouse through **ETL** (extract, transform, load) or, in many cloud designs, **ELT** (load first, transform inside the target). Each step should reconcile record counts and control totals so nothing is lost or duplicated.

> **Exam tip:** "Schema-on-write" = warehouse (cleaned before loading). "Schema-on-read" = lake (raw until used). A data mart is a slice of a warehouse.

Big data

Big data is often described by the "Vs": **volume**, **velocity** (speed of arrival), **variety** (structured, semi-structured, unstructured), plus **veracity** (trustworthiness) and **value**. Variety and veracity raise the most control questions: unstructured data is harder to validate, classify and protect.

Retention and destruction

- A **records retention schedule** sets how long each record type is kept, based on legal, tax, regulatory and business needs.
- A **legal hold** suspends normal destruction for records relevant to actual or expected litigation or investigation. It **overrides** the retention schedule.
- Destruction methods must fit the media:

Method	Works for	Notes
Overwriting (clearing)	Magnetic hard drives	Multiple tools available; less reliable for SSDs
Degaussing	**Magnetic** media (tapes, hard disk drives)	Does **not** work on solid-state drives (SSDs) or optical media
Cryptographic erasure (crypto-shredding)	Encrypted data, including cloud and SSDs	Destroy the encryption keys so data cannot be decrypted
Physical destruction	Any media	Shredding, disintegration, incineration; keep certificates of destruction

Watch out: Degaussing an SSD does nothing useful. For SSDs and cloud storage, look for cryptographic erasure or physical destruction.

Data analytics

Types of analytics

Type	Question	Example
Descriptive	What happened?	Monthly sales by region dashboard
Diagnostic	Why did it happen?	Drill-down showing a sales drop came from one product line
Predictive	What will happen?	Model forecasting which customers will pay late
Prescriptive	What should we do?	Optimization recommending inventory reorder quantities

Memory trick: "**D**id, **D**id-why, **P**redict, **P**rescribe" — the order rises in difficulty and value.

The analytics process

1. Define the question or objective.
2. Identify and obtain the data (with authorization).
3. **Validate the extract** — reconcile record counts and totals to the source (for example, total invoices in the extract equals the general ledger balance). Without this, every result is suspect.
4. Clean and prepare the data (fix formats, handle missing values, remove true duplicates).
5. Analyze.
6. Interpret and communicate results, including limitations.

Common audit and fraud analytics

Test	Looks for
Duplicate test	Same vendor + amount + date or invoice number (duplicate payments)
Gap test	Missing check or invoice numbers
Stratification	Distribution of transactions across value bands
Join / match	Employees whose bank accounts or addresses match vendors
Round-number test	Unusual number of round amounts (possible estimates or fabrication)
Just-below-threshold test	Many amounts just under an approval limit
Weekend / after-hours test	Entries posted at unusual times
Benford's law	First-digit distribution that departs from the expected pattern

Benford's law

In many naturally occurring data sets that span several orders of magnitude (invoice amounts, expense claims, account balances), the first digit is **not** evenly distributed. The expected proportion of numbers beginning with digit d is:

P(d) = log10(1 + 1/d)

First digit	1	2	3	4	5	6	7	8	9
Expected %	30.1	17.6	12.5	9.7	7.9	6.7	5.8	5.1	4.6

Example: Expected proportion of amounts beginning with 1 = log10(1 + 1/1) = log10(2) = **0.301 (30.1%)**. For 9 = log10(1 + 1/9) = log10(1.111) = **0.046 (4.6%)**. If 2,000 expense claims were tested, about 2,000 × 0.301 ≈ 602 would be expected to begin with 1.

Watch out: Benford's law does **not** apply to **assigned** or **constrained** numbers: check numbers, invoice numbers, ZIP codes, telephone numbers, prices set at $9.99, or amounts with a fixed minimum or maximum. A departure is a signal to investigate, not proof of fraud.

Outliers and z-scores

A z-score measures how many standard deviations a value lies from the mean:

z = (x − mean) ÷ standard deviation

Values with |z| above about 2 or 3 are commonly flagged for investigation.

> **Example:** Average weekly overtime cost for a department is $6,200 with a standard deviation of $1,500. One week shows $9,800.
>
> 1. z = (9,800 − 6,200) ÷ 1,500 = 3,600 ÷ 1,500 = **2.4**.
> 2. The week is 2.4 standard deviations above the mean, so it should be investigated (not deleted).

The **median** resists outliers better than the mean. In the set 1,200; 1,260; 1,280; 1,310; 1,350; 5,400, the mean is about 1,967 because of the single 5,400 value, while the median is (1,280 + 1,310) ÷ 2 = **1,295**.

> **Exam tip:** The correct response to an outlier is to **investigate** it. Deleting outliers to make data "cleaner" can hide exactly the error or fraud the analysis was meant to find.

Visualization basics

Choose the chart for the question: line charts for trends over time, bar charts for comparing categories, scatter plots for relationships between two variables, histograms for distributions. Misleading visuals (truncated axes, inconsistent scales, 3-D effects) are a data integrity risk too.

Common traps

- Confusing validity (format) with accuracy (truth).
- Treating IT as the data owner.
- Applying Benford's law to assigned numbers.
- Deleting outliers instead of investigating them.
- Degaussing SSDs.
- Running analytics before reconciling the extract to the source.
- Letting a retention schedule override a legal hold.

Key takeaways

- Owner decides, steward defines quality rules, custodian operates controls.
- Quality dimensions: accuracy, completeness, validity, consistency, timeliness, uniqueness (and integrity).
- Warehouse = schema-on-write; lake = schema-on-read; mart = subset of a warehouse.
- Legal holds override retention schedules; degaussing works only on magnetic media; crypto-shredding destroys keys.

- Analytics: descriptive, diagnostic, predictive, prescriptive; always validate the extract first.
- Benford: P(d) = log10(1 + 1/d); 30.1% begin with 1, 4.6% with 9.
- z = (x − mean) ÷ standard deviation; investigate outliers.

7. Databases and SQL

Almost all accounting data lives in relational databases. ISC expects you to understand how tables are structured and linked, why normalization matters, how database transactions protect integrity, and — most of all — to **read a SQL query and predict its result**. Every query in this chapter was run against the sample tables shown, and the results printed here are the actual outputs.

Relational database basics

Term	Meaning
Table (relation)	Data about one kind of thing, in rows and columns
Row (record, tuple)	One instance — one customer, one invoice
Column (field, attribute)	One property — name, amount, date
Primary key	Column(s) whose value **uniquely identifies** each row; cannot be null
Composite key	A primary key made of two or more columns (for example, order number + line number)
Foreign key	Column in one table that refers to the primary key of another table
Referential integrity	Every foreign key value must match an existing primary key (or be null if allowed)
Database management system (DBMS)	Software that stores data, enforces rules and processes queries
Data dictionary	Metadata: table and field names, types, definitions, owners
Schema	The overall structure of tables and relationships

Example: An Invoices table has a CustID column that refers to the Customers table. With referential integrity enforced, an attempt to insert an invoice for CustID 999 — a customer that does not exist — is **rejected** by the DBMS. (In the test run for this guide, the database returned "FOREIGN KEY constraint failed.") Referential integrity is a preventive control against orphan records.

Relationships

- **One-to-one:** each employee has one badge record.
- **One-to-many:** one customer has many invoices (the most common).

- **Many-to-many:** students and courses; implemented with a **junction (linking) table** holding both keys.

Normalization

Normalization organizes tables to reduce redundancy and prevent **anomalies**:

- **Insert anomaly:** you cannot record a fact without an unrelated fact (cannot add a new product until someone orders it).
- **Update anomaly:** the same fact is stored in many places, so an update can miss some copies (a customer address stored on every invoice).
- **Delete anomaly:** deleting one fact deletes another (deleting the only order for a customer also deletes the customer's details).

Normal form	Rule	Fix
First normal form (1NF)	Every field holds one atomic value; no repeating groups	Split repeating items into separate rows or a separate table
Second normal form (2NF)	1NF, and no non-key field depends on only **part** of a composite key	Move partially dependent fields to their own table
Third normal form (3NF)	2NF, and no non-key field depends on **another non-key field** (no transitive dependency)	Move the dependent fields to their own table

Example: An OrderLines table has the composite key (OrderNo, ProductID) and the columns Quantity, ProductName and CustomerName.

1. **ProductName** depends only on ProductID — part of the key. That is a **partial dependency**, so the table is not in 2NF. Move ProductName to a Products table.
2. **CustomerName** depends on OrderNo (part of the key) — also partial. It belongs in an Orders table keyed by OrderNo, with a CustID foreign key.
3. If the Orders table then held CustID **and** CustomerCity, CustomerCity would depend on CustID (a non-key field), a **transitive dependency** — move it to Customers to reach 3NF.

Memory trick: Each non-key field must depend on "**the key** (1NF), **the whole key** (2NF), and **nothing but the key** (3NF)."

Data warehouses often **denormalize** on purpose (for example, star schemas with a central fact table and dimension tables) because analysis reads much more than it writes.

Database transactions and ACID

A database transaction is a group of operations that must succeed or fail together. ACID properties protect integrity:

Property	Meaning
Atomicity	All steps happen or none do (a transfer debits and credits, or neither)
Consistency	A transaction moves the database from one valid state to another, respecting all rules
Isolation	Concurrent transactions do not interfere with each other
Durability	Once committed, changes survive failures

COMMIT makes a transaction permanent; ROLLBACK undoes it. Transaction logs record changes so the DBMS can recover after a failure.

SQL command categories

Category	Commands	Purpose
DDL — data definition language	CREATE, ALTER, DROP, TRUNCATE	Define structures
DML — data manipulation language	SELECT, INSERT, UPDATE, DELETE	Work with rows
DCL — data control language	GRANT, REVOKE	Permissions
TCL — transaction control language	COMMIT, ROLLBACK, SAVEPOINT	Transaction boundaries

> **Exam tip:** GRANT and REVOKE are **access controls** inside the database. If a question asks which statement gives a user read access to a table, the category is DCL.

The sample tables

All queries below use these two tables.

Customers

CustID	Name	Region	CreditLimit
101	Alder Supply	East	20,000
102	Birch Foods	West	15,000
103	Cedar Labs	East	30,000
104	Dogwood Inc	North	10,000
105	Elm Partners	West	(null)

Invoices

InvNo	CustID	InvDate	Amount	PaidDate
5001	101	2025-10-02	4,200	2025-10-30
5002	102	2025-10-05	8,600	(null)
5003	101	2025-10-11	3,900	(null)
5004	103	2025-10-12	12,500	2025-11-03
5005	102	2025-10-19	7,100	(null)
5006	103	2025-10-25	950	(null)
5007	101	2025-10-28	4,200	(null)

A null PaidDate means the invoice is still open.

Logical order of a SELECT query

You write a query in this order: SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY. But the database **evaluates** it in this order:

1. **FROM** (and JOIN) — which tables
2. **WHERE** — filter individual rows
3. **GROUP BY** — form groups
4. **HAVING** — filter groups (using aggregates)
5. **SELECT** — choose and compute output columns
6. **ORDER BY** — sort the result

> **Memory trick:** "**F**rank **W**ants **G**reen **H**ats, **S**ays **O**liver" — From, Where, Group by, Having, Select, Order by.

Filtering rows with WHERE

> **Example:** Which invoices are $5,000 or more, largest first?
>
> ```
> SELECT InvNo, CustID, Amount
> FROM Invoices
> WHERE Amount >= 5000
> ORDER BY Amount DESC;
> ```
>
> Result: **5004 / 103 / 12,500; 5002 / 102 / 8,600; 5005 / 102 / 7,100** — three rows. Invoice 5001 ($4,200) fails the condition.

Useful WHERE operators: `=`, `<>`, `>`, `>=`, `BETWEEN a AND b` (inclusive of both ends), `IN (list)`, `LIKE 'A%'` (pattern; % = any characters), `IS NULL` / `IS NOT NULL`, combined with AND, OR and NOT.

Watch out: You cannot test for null with `= NULL`. Null means "unknown," and any comparison with it is unknown, not true. Use `IS NULL`.

Aggregates, GROUP BY and HAVING

Aggregate functions: COUNT, SUM, AVG, MIN, MAX. GROUP BY produces one output row per group.

Example: Invoice count and total by customer.

```
SELECT CustID, COUNT(*) AS NumInv,
       SUM(Amount) AS Total
FROM Invoices
GROUP BY CustID;
```

CustID	NumInv	Total
101	3	12,300
102	2	15,700
103	2	13,450

Customers 104 and 105 do not appear — they have no rows in Invoices.

Example: Add a HAVING clause to keep only customers whose total exceeds $13,000.

```
SELECT CustID, SUM(Amount) AS Total
FROM Invoices
GROUP BY CustID
HAVING SUM(Amount) > 13000;
```

Result: **102 (15,700) and 103 (13,450)**. Customer 101's total of 12,300 fails the group condition.

WHERE versus HAVING — the classic exam item

Example: Which customers have **open** (unpaid) invoices totaling more than $13,000?

```
SELECT CustID, SUM(Amount) AS OpenTotal
FROM Invoices
WHERE PaidDate IS NULL
GROUP BY CustID
HAVING SUM(Amount) > 13000;
```

1. WHERE first removes the paid invoices 5001 and 5004.
2. GROUP BY then totals the remaining open invoices: 101 = 3,900 + 4,200 = **8,100**; 102 = 8,600 + 7,100 = **15,700**; 103 = **950**.
3. HAVING keeps groups over 13,000: only **customer 102 (15,700)**.

Compare with the previous query: without the WHERE clause, customer 103 qualified because its paid $12,500 invoice was counted. The row filter (WHERE) and the group filter (HAVING) answer different questions.

Watch out: WHERE filters **rows before grouping** and cannot contain an aggregate such as SUM. HAVING filters **groups after aggregation**.

Joins

A join combines rows from two tables using a matching condition, usually primary key = foreign key.

Join	Returns
INNER JOIN	Only rows with a match in **both** tables
LEFT (OUTER) JOIN	**Every** row from the left table, plus matching right-table data; nulls where there is no match
RIGHT (OUTER) JOIN	Every row from the right table, plus matches
FULL OUTER JOIN	All rows from both tables, matched where possible

Example (inner join): Invoices for customers in the East region.

```
SELECT c.Name, i.InvNo, i.Amount
FROM Customers c
INNER JOIN Invoices i ON c.CustID = i.CustID
WHERE c.Region = 'East'
ORDER BY i.InvNo;
```

Result (5 rows): Alder Supply 5001 4,200; Alder Supply 5003 3,900; Cedar Labs 5004 12,500; Cedar Labs 5006 950; Alder Supply 5007 4,200.

Example (left join): Every customer with its invoice count and total — including customers with no invoices.

```
SELECT c.Name, COUNT(i.InvNo) AS NumInv,
       SUM(i.Amount) AS Total
FROM Customers c
LEFT JOIN Invoices i ON c.CustID = i.CustID
GROUP BY c.CustID, c.Name;
```

Name	NumInv	Total
Alder Supply	3	12,300
Birch Foods	2	15,700
Cedar Labs	2	13,450
Dogwood Inc	0	(null)
Elm Partners	0	(null)

The left join keeps Dogwood and Elm with nulls on the invoice side. COUNT(i.InvNo) returns **0** for them because it skips nulls; SUM over no values returns **null**, not zero. Had the query used COUNT(*), each would show **1** — the single null-filled row.

COUNT and null rules

Expression	Counts
COUNT(*)	All rows, including rows with nulls
COUNT(column)	Rows where that column is **not null**
COUNT(DISTINCT column)	Distinct non-null values
AVG(column), SUM(column)	Ignore nulls

Example:

```
SELECT COUNT(*), COUNT(PaidDate),
       COUNT(DISTINCT CustID)
FROM Invoices;
```

Result: **7, 2, 3** — seven invoices, two with a paid date, three different customers.

```
SELECT COUNT(*), COUNT(CreditLimit),
       AVG(CreditLimit)
FROM Customers;
```

Result: **5, 4, 18,750**. AVG ignores Elm's null limit: (20,000 + 15,000 + 30,000 + 10,000) ÷ **4** = 18,750, not ÷ 5 = 15,000.

SQL for audit tests

Example (possible duplicate invoices):

```
SELECT CustID, Amount, COUNT(*) AS Times
FROM Invoices
GROUP BY CustID, Amount
HAVING COUNT(*) > 1;
```

Result: **101, 4,200, 2** — invoices 5001 and 5007 are both $4,200 to Alder Supply. Follow up: a true duplicate, or two legitimate orders?

Example (customers over their credit limit on open invoices):

```
SELECT c.Name, SUM(i.Amount) AS OpenAR,
       c.CreditLimit
FROM Customers c
JOIN Invoices i ON c.CustID = i.CustID
WHERE i.PaidDate IS NULL
GROUP BY c.CustID
HAVING SUM(i.Amount) > c.CreditLimit;
```

Result: **Birch Foods, 15,700, 15,000**. Birch's open invoices exceed its limit by $700 — the credit check control failed or was overridden. (A customer with a null limit would never appear, because a comparison with null is not true — itself a control gap worth noting.)

Other audit queries: invoices whose CustID is NOT IN the Customers table (orphans — none here, because referential integrity is enforced), gaps in invoice numbers, entries posted on weekends, and employees whose bank account matches a vendor's.

Changing data: INSERT, UPDATE, DELETE

```
UPDATE Invoices
SET PaidDate = '2025-11-05'
WHERE InvNo = 5003;
```

Watch out: An UPDATE or DELETE **without a WHERE clause** affects **every row** in the table. This is why direct production data changes must be controlled, logged and reviewed (Chapter 4). In a transaction, ROLLBACK can undo such a mistake before COMMIT; after COMMIT, recovery needs backups or logs.

SQL injection

SQL injection happens when an application builds a query by pasting user input directly into SQL text. Suppose the application builds:

```
SELECT COUNT(*) FROM Customers
WHERE Name = '<user input>';
```

If an attacker types `x' OR '1'='1`, the query becomes `... WHERE Name = 'x' OR '1'='1'`, which is always true. In the test run it returned **5** — every customer — instead of 0. With a **parameterized query** (the input passed separately as data, never as code), the same input returned **0**, because the DBMS looked for a customer literally named `x' OR '1'='1`.

Exam tip: The primary defense against SQL injection is **parameterized queries (prepared statements)**, supported by input validation, least-privilege database accounts and a web application firewall. Encryption does not stop SQL injection.

Common traps

- Putting an aggregate condition in WHERE, or a row condition in HAVING.
- Forgetting that a LEFT JOIN keeps unmatched left rows with nulls.
- Using COUNT(*) when the item asks for COUNT(column) or COUNT(DISTINCT ...).
- Dividing by all rows when AVG ignores nulls.
- Testing `= NULL` instead of `IS NULL`.
- Treating BETWEEN as exclusive — it includes both endpoints.

Key takeaways

- Primary key = unique row identifier; foreign key = link to another table; referential integrity blocks orphans.
- 1NF atomic values; 2NF no partial dependency; 3NF no transitive dependency.
- ACID: atomicity, consistency, isolation, durability; COMMIT and ROLLBACK.
- DDL defines, DML manipulates, DCL grants and revokes, TCL commits and rolls back.
- Evaluation order: FROM, WHERE, GROUP BY, HAVING, SELECT, ORDER BY.
- INNER JOIN keeps matches only; LEFT JOIN keeps every left row.
- COUNT(*) counts all rows; COUNT(column) and AVG skip nulls.
- Parameterized queries prevent SQL injection.

8. Security Frameworks and Governance

Area II opens with the structures organizations use to run a security program: frameworks that say what good looks like, governance that assigns accountability, and risk management that decides how much risk to accept. ISC tests the frameworks by name and structure, so precise lists matter.

Why frameworks?

A framework gives a common language and a checklist of outcomes so an organization can assess where it is, decide where it wants to be, and prioritize the gap. Frameworks differ in purpose:

Framework	Purpose	Certifiable?
NIST Cybersecurity Framework (CSF) 2.0	Voluntary outcomes-based framework for managing cybersecurity risk	No — NIST does not certify
NIST SP 800-53 Rev. 5	Detailed catalog of security and privacy controls (required for U.S. federal systems)	No (used for federal authorizations)
ISO/IEC 27001:2022	Requirements for an information security management system (ISMS)	**Yes** — by accredited certification bodies
COBIT 2019	Governance and management of enterprise information and technology	No (organizations can be assessed)
CIS Critical Security Controls v8.1	Prioritized set of defensive safeguards	No
COSO 2013	Internal control over objectives, including financial reporting	No
PCI DSS v4.0.1	Mandatory standard for entities handling payment card data	Validated by assessments (QSA reports or self-assessment)

Watch out: "NIST certified" is not a thing for the CSF. If an answer says an organization can be certified against CSF tiers, it is wrong. ISO/IEC 27001 is the certifiable standard.

NIST Cybersecurity Framework 2.0

NIST released CSF 2.0 in **February 2024**. It applies to organizations of every size and sector (not only critical infrastructure) and has three parts: the **Core**, **Tiers** and **Profiles**.

The Core: six Functions

Function	Code	Outcome in plain language
Govern (new in 2.0)	GV	Strategy, expectations, policy, roles, oversight and **supply chain risk management** are established and monitored
Identify	ID	The organization understands its assets, risks and improvement needs
Protect	PR	Safeguards manage risk: identity and access, awareness and training, data security, platform security, infrastructure resilience
Detect	DE	Possible attacks and compromises are found and analyzed (continuous monitoring, adverse event analysis)
Respond	RS	Actions are taken on a detected incident (management, analysis, reporting and communication, mitigation)
Recover	RC	Assets and operations affected by an incident are restored

Govern sits at the center of the CSF 2.0 wheel because it informs how all the other Functions are carried out. Each Function is divided into **Categories** (for example, GV.SC — Cybersecurity Supply Chain Risk Management) and **Subcategories** (specific outcomes).

Memory trick: "**G**ood **I**deas **P**rotect, **D**etect, **R**espond, **R**ecover" — G-I-P-D-R-R.

Watch out: In CSF 1.1, supply chain risk management sat under Identify. In CSF 2.0 it moved to **Govern (GV.SC)**. Any answer describing CSF 2.0 with five Functions is outdated.

Tiers

Tiers describe how rigorous an organization's cybersecurity risk governance and management practices are:

Tier	Name	Description
1	Partial	Ad hoc, reactive
2	Risk Informed	Management-approved practices, but not organization-wide policy
3	Repeatable	Formally approved, consistently applied, regularly updated
4	Adaptive	Continuously improved using lessons learned and predictive indicators

Tiers are **not maturity certifications** and an organization need not aim for Tier 4 everywhere; the right Tier depends on risk and resources.

Profiles

A **Current Profile** describes the outcomes the organization achieves now; a **Target Profile** describes the outcomes it wants. The gap between them drives an action plan. **Community Profiles** are baselines shared by a sector or for a common use case.

> **Example:** A regional bank completes a Current Profile showing it has no formal process to assess the security of key vendors. Its Target Profile, set by the board, calls for risk-based vendor due diligence and contract security requirements. The gap falls under **Govern — supply chain risk management (GV.SC)**, and closing it becomes a funded action item with an owner and a deadline.

COBIT 2019

COBIT 2019 (ISACA) separates **governance** from **management**.

Area	Domain	Meaning
Governance (board level)	**EDM** — Evaluate, Direct and Monitor	Board evaluates options, sets direction, monitors performance
Management	**APO** — Align, Plan and Organize	Strategy, architecture, budget, risk, security management
Management	**BAI** — Build, Acquire and Implement	Projects, requirements, solutions, change
Management	**DSS** — Deliver, Service and Support	Operations, service requests, incidents, continuity, security services
Management	**MEA** — Monitor, Evaluate and Assess	Performance monitoring, internal control, compliance, assurance

COBIT 2019 has **40 governance and management objectives** (5 in EDM and 35 across the management domains). It also uses **design factors** to tailor a governance system and **capability levels (0–5)** to assess processes.

> **Memory trick:** The board **EDM**s; management **Plans (APO), Builds (BAI), Runs (DSS) and Monitors (MEA)**.

CIS Critical Security Controls v8.1

The CIS Controls are **18** prioritized controls containing specific safeguards. They begin with knowing what you have: **Control 1** is inventory and control of enterprise assets, and **Control 2** is inventory and control of software assets. Safeguards are grouped into **Implementation Groups**:

Group	For	Scope
IG1	Every organization; small organizations with limited IT expertise	"Essential cyber hygiene" — the minimum baseline
IG2	Organizations with more complex environments and sensitive data	IG1 + additional safeguards
IG3	Organizations with significant risk and dedicated security experts	All safeguards

> **Exam tip:** You cannot protect, patch or monitor what you do not know exists. When asked which control should come first in building a security program, look for **asset inventory**.

ISO/IEC 27001 and 27002

ISO/IEC 27001:2022 specifies requirements for establishing, implementing, maintaining and continually improving an **ISMS**, using a risk-based approach. Its **Annex A** lists **93 controls** in **four themes**:

Theme	Number of controls
Organizational	37
People	8
Physical	14
Technological	34

ISO/IEC 27002 gives guidance on implementing those controls. A **Statement of Applicability** documents which Annex A controls apply and why. Certification is granted by an accredited body after an audit, followed by surveillance audits.

Policies, standards, procedures and guidelines

Document	Nature	Example
Policy	High-level management intent; **mandatory**	"All remote access must use multi-factor authentication."
Standard	Specific **mandatory** requirement supporting a policy	"Use authenticator apps or hardware keys; SMS codes not allowed for administrators."
Procedure	Step-by-step instructions	How the help desk enrolls a user in MFA
Guideline	Recommended practice; **optional**	Tips for choosing a memorable passphrase
Baseline	Minimum configuration settings for a system type	Server hardening baseline

> **Watch out:** Standards are mandatory; guidelines are optional. Do not confuse them.

Security governance roles

- **Board of directors:** oversight of cyber risk as an enterprise risk; often through the audit or risk committee.
- **Chief information security officer (CISO):** leads the security program; should have enough independence and a reporting line that lets risks reach senior management and the board.
- **Chief information officer (CIO):** runs IT operations and delivery.
- **Internal audit:** independent assurance on the program.
- **Three lines model:** first line (operational management owns and manages risk), second line (risk, compliance and security functions set frameworks and monitor), third line (internal audit gives independent assurance).

SEC disclosure of cybersecurity governance

Public companies must describe their cybersecurity risk management, strategy and governance — including board oversight and management's role and expertise — **annually** in Form 10-K under **Regulation S-K Item 106**. Material incidents are reported separately on **Form 8-K Item 1.05** (Chapter 12).

Risk management

Key terms

Term	Meaning
Asset	Anything of value: data, systems, people, reputation
Threat	Something that could cause harm (an attacker, a flood)
Vulnerability	A weakness a threat could exploit
Likelihood and impact	The two dimensions of risk
Inherent risk	Risk before considering controls
Residual risk	Risk remaining after controls
Risk appetite	The amount of risk the organization is willing to accept in pursuit of its objectives
Risk tolerance	Acceptable variation around specific objectives

Risk responses

Response	Meaning	Example
Avoid	Stop the activity	Decide not to store card numbers at all
Mitigate (reduce)	Add controls to reduce likelihood or impact	Add MFA and endpoint detection
Transfer (share)	Shift financial impact to another party	Buy cyber insurance; outsource with contractual indemnity
Accept	Retain the risk knowingly, within appetite	Accept a low risk where controls would cost more than the loss

Example: On a 1–5 scale, a vendor-payment fraud scenario has likelihood 4 and impact 5, an inherent score of 4 × 5 = **20**. After adding dual approval and call-back verification, likelihood falls to 2 while impact stays 5: residual score 2 × 5 = **10**. The board's appetite allows residual scores up to 8, so the risk is **still above appetite** — management needs a further response (for example, lowering impact through payment limits or insurance) or a documented decision by an authorized person to accept it.

Exam tip: Insurance **transfers** financial impact but does not transfer accountability or reputational harm. The organization still owns the risk.

Common traps

- Saying CSF 2.0 has five Functions or that NIST certifies organizations.
- Placing supply chain risk in Identify instead of Govern.
- Treating Tiers as maturity levels everyone must reach.
- Mixing up COBIT's governance domain (EDM) with management domains.
- Confusing a standard (mandatory) with a guideline (optional).
- Thinking insurance eliminates the risk.

Key takeaways

- CSF 2.0 (February 2024): Core with six Functions — Govern, Identify, Protect, Detect, Respond, Recover — plus Tiers (Partial, Risk Informed, Repeatable, Adaptive) and Profiles (Current, Target, Community).
- COBIT 2019: 40 objectives; EDM governance; APO, BAI, DSS, MEA management.
- CIS Controls v8.1: 18 controls; IG1 = essential cyber hygiene; Control 1 = asset inventory.
- ISO/IEC 27001:2022: certifiable ISMS; Annex A has 93 controls in 4 themes.
- Policy and standard are mandatory; guideline is optional.
- Risk responses: avoid, mitigate, transfer, accept; compare residual risk with appetite.
- SEC: S-K Item 106 annual governance disclosure; 8-K Item 1.05 for material incidents.

9. Threats, Attacks and Cyber Risk

To choose the right control you first have to recognize the threat. This chapter catalogs the attacks ISC tests — against people, systems, applications and networks — and then shows how to quantify cyber risk with single and annualized loss expectancy.

Threat actors

Actor	Typical motive
Cybercriminals and ransomware groups	Money
Nation-state actors (advanced persistent threats)	Espionage, disruption; long, stealthy campaigns
Hacktivists	Publicity for a cause
Insiders (malicious or careless)	Revenge, money — or simple mistakes
Script kiddies	Curiosity, notoriety; use ready-made tools
Competitors	Trade secrets

> **Exam tip:** Insiders already have legitimate access, so perimeter defenses do little against them. The relevant controls are least privilege, segregation of duties, monitoring of user activity and prompt removal of access.

Social engineering

Social engineering manipulates people rather than technology.

Attack	How it works
Phishing	Mass e-mails that trick recipients into clicking links, opening attachments or giving credentials
Spear phishing	Phishing targeted at a specific person or group using personal details
Whaling	Spear phishing aimed at senior executives
Smishing	Phishing by text message (SMS)
Vishing	Phishing by voice call, increasingly with AI-cloned voices
Pretexting	A fabricated story to obtain information (posing as IT support or an auditor)
Baiting	Leaving infected USB drives or offering free downloads
Quid pro quo	Offering a service in exchange for information or access
Tailgating / piggybacking	Following an authorized person through a secure door
Business e-mail compromise (BEC)	Impersonating an executive or vendor (often from a compromised or look-alike account) to redirect payments
Pharming	Redirecting users to a fake site by corrupting DNS or host files — no click on a bad link needed

Watch out: Pharming redirects traffic through DNS manipulation even when the user types the correct address. **Phishing** relies on luring the user to a bad link. Know the difference.

Example: The accounts payable team receives an e-mail that appears to come from a long-time supplier: "We have changed banks — please send all future payments to the new account below." The e-mail has no attachment and no link. This is **business e-mail compromise**. Antivirus will not catch it because there is no malware. The effective controls are:

1. **Call-back verification** to the supplier using a phone number already on file (never the one in the e-mail).
2. **Dual approval** of vendor bank account changes in the vendor master.
3. Review of a vendor master change report and a cooling-off period before the first payment to new bank details.
4. Staff training to recognize urgency and secrecy as red flags.

Malware

Type	Key feature
Virus	Attaches to a host file and needs user action to spread
Worm	**Self-replicates** across networks without user action
Trojan horse	Disguised as legitimate software
Ransomware	Encrypts data (and often steals it first — "double extortion") and demands payment
Spyware / keylogger	Secretly records activity or keystrokes
Rootkit	Hides deep in the operating system to maintain privileged, concealed access
Logic bomb	Code that triggers on a condition (a date, or an employee's removal from payroll)
Backdoor	Hidden way to bypass normal authentication
Botnet	Network of infected devices controlled remotely, often used for DDoS or spam
Fileless malware	Runs in memory using legitimate system tools, leaving few files to scan

> **Memory trick:** A **w**orm **w**alks by itself; a virus needs a ride.

Application attacks

Attack	Description	Primary defense
SQL injection	Malicious SQL inserted through input fields (Chapter 7)	Parameterized queries, input validation
Cross-site scripting (XSS)	Malicious script injected into a web page and run in other users' browsers	Output encoding, input validation, content security policy
Cross-site request forgery (CSRF)	Tricks a logged-in user's browser into sending an unwanted request	Anti-CSRF tokens, same-site cookies
Buffer overflow	Input larger than the memory set aside overwrites adjacent memory	Bounds checking, secure coding, memory-safe languages
Broken access control	Users reach data or functions they should not (for example, by changing an ID in a URL)	Server-side authorization checks on every request
Zero-day exploit	Attack on a vulnerability with no vendor patch yet	Defense in depth, behavior-based detection, segmentation

Network and other technical attacks

Attack	Description
Denial of service (DoS) / distributed DoS (DDoS)	Overwhelms a system so legitimate users cannot reach it — an attack on **availability**
Man-in-the-middle (on-path)	Attacker secretly intercepts or alters communications between two parties
Eavesdropping / sniffing	Capturing unencrypted network traffic
Spoofing	Faking an identity (IP address, e-mail sender, caller ID)
Session hijacking	Stealing a session token to take over a logged-in session
Password attacks	Brute force (all combinations), dictionary, password spraying (one common password across many accounts)
Credential stuffing	Reusing username-password pairs leaked from other breaches
Supply chain attack	Compromising a trusted vendor, software update or library to reach its customers
Advanced persistent threat (APT)	Long-term, stealthy intrusion, usually by well-resourced groups

Exam tip: Credential stuffing and password spraying are defeated mainly by **multi-factor authentication**, plus rate limiting, lockout or throttling, and screening passwords against breached-password lists.

AI-enabled threats

Generative AI makes phishing messages more convincing, produces deepfake audio or video of executives, and speeds up malware development. Attacks on AI systems themselves include **prompt injection** (instructions hidden in input that redirect a model) and **data poisoning** (corrupting training data). Controls include verification procedures that do not rely on recognizing a voice or face, and governance over AI tools and their data access.

The CIA triad and attack mapping

Security objective	Attack that threatens it most directly
Confidentiality	Eavesdropping, data exfiltration, credential theft
Integrity	Unauthorized changes, man-in-the-middle alteration, SQL injection updates
Availability	DoS/DDoS, ransomware encryption, destruction of backups

Quantitative risk analysis: SLE and ALE

Quantitative analysis puts a dollar value on a risk scenario so management can compare it with the cost of controls.

Term	Formula / meaning
Asset value (AV)	Value of the asset at risk
Exposure factor (EF)	Percentage of the asset's value lost in one occurrence
Single loss expectancy (SLE)	**SLE = AV × EF**
Annualized rate of occurrence (ARO)	Expected number of occurrences per year (once every 4 years = 0.25)
Annualized loss expectancy (ALE)	**ALE = SLE × ARO**
Value of a control	**ALE before − ALE after − annual cost of the control** (implement if positive)

Example: A customer database is valued at $400,000. A successful breach is expected to cost 25% of that value, and such a breach is expected once every two years. A proposed data loss prevention tool would reduce the expected frequency to once every ten years and costs $25,000 per year.

1. SLE = $400,000 × 25% = **$100,000**.
2. ARO before = 1 ÷ 2 = 0.5, so ALE before = $100,000 × 0.5 = **$50,000**.
3. ARO after = 1 ÷ 10 = 0.1, so ALE after = $100,000 × 0.1 = **$10,000**.
4. Annual risk reduction = $50,000 − $10,000 = $40,000.
5. Net value of the control = $40,000 − $25,000 = **$15,000** per year. It is cost-justified.

Example: A manufacturer values its production systems and data at $2,000,000. A ransomware attack is expected once every four years and would destroy or disrupt 40% of that value. Immutable offline backups would cut the exposure factor to 10% (faster recovery, no ransom needed) at an annual cost of $70,000.

1. SLE before = $2,000,000 × 40% = **$800,000**; ARO = 0.25; ALE before = $800,000 × 0.25 = **$200,000**.
2. SLE after = $2,000,000 × 10% = $200,000; ALE after = $200,000 × 0.25 = **$50,000**.
3. Control value = $200,000 − $50,000 − $70,000 = **$80,000** per year. Implement.

Note that this control changed the **exposure factor** (impact), not the frequency. Some controls reduce likelihood (ARO), others reduce impact (EF).

Watch out: If the question asks for ALE, do not stop at SLE. And always subtract the control's **annual cost** before deciding whether it is worthwhile.

Qualitative analysis

When reliable numbers are not available, organizations rate likelihood and impact on scales (for example, low / medium / high or 1–5) and plot them on a **heat map**. Qualitative analysis is faster and easier to communicate; quantitative analysis supports cost-benefit decisions. Many organizations use both.

Defense in depth

No single control stops every attack. **Defense in depth** layers controls — policies and training, physical security, perimeter defenses, network segmentation, endpoint protection, application security, data encryption, monitoring and backups — so that if one layer fails, another still protects the asset.

Common traps

- Reporting SLE when the question asks for ALE.
- Ignoring the control's annual cost in a cost-benefit question.
- Calling DNS-based redirection phishing instead of pharming.
- Calling a self-spreading program a virus instead of a worm.
- Expecting antivirus to stop BEC; the fix is process verification.
- Treating ARO as a probability that cannot exceed 1 — an event expected three times a year has ARO 3.

Key takeaways

- Social engineering targets people: phishing, spear phishing, whaling, smishing, vishing, pretexting, baiting, tailgating, BEC, pharming.
- Worms self-spread; viruses need a host and user action; Trojans are disguised; ransomware encrypts and often exfiltrates.
- SQL injection → parameterized queries; XSS → output encoding; credential stuffing → MFA.
- SLE = AV × EF; ALE = SLE × ARO; control value = ALE before − ALE after − annual cost.
- Defense in depth layers people, process and technology controls.

10. Logical and Physical Access Controls

Access controls decide who can reach which systems, data and facilities, and they create the accountability trail that shows who did what. They are the most frequently tested ITGC. This chapter covers the access control process, authentication factors and passwords, biometrics, authorization models, the user access life cycle, privileged access and physical security.

The access control process

Step	Question	Example
Identification	Who do you claim to be?	Entering a user ID
Authentication	Prove it	Password plus a code from an authenticator app
Authorization	What are you allowed to do?	Role permits viewing, not approving, invoices
Accountability (auditing)	What did you do?	Logs tie actions to the unique user ID

Memory trick: "**I A A A**" — Identify, Authenticate, Authorize, Account. Accountability depends on **unique** IDs; shared accounts break it.

Authentication factors

Factor category	Examples
Something you **know**	Password, passphrase, PIN, answers to security questions
Something you **have**	Smart card, hardware security key, phone with authenticator app, one-time-password token
Something you **are**	Fingerprint, face, iris, voice (biometrics)
Context (supplementary)	Location, device, time of day — used in risk-based or adaptive authentication

Multi-factor authentication (MFA) requires factors from **two or more different categories**.

Watch out: A password plus a security question is **two knowledge factors** — that is not MFA. A password plus a fingerprint, or a smart card plus a PIN, is MFA.

Strength of MFA methods

Not all second factors are equal. SMS codes can be intercepted through SIM swapping; push notifications can be abused by "MFA fatigue" (repeated prompts until the user approves). **Phishing-resistant** methods — FIDO2/WebAuthn security keys and passkeys, or smart-card certificates — bind authentication to the real site, so a fake login page cannot capture a usable credential.

Single sign-on and federation

Single sign-on (SSO) lets a user authenticate once to an identity provider and reach many applications. Federation standards (such as SAML and OpenID Connect) pass assertions between the identity provider and applications. SSO improves usability and lets the organization disable access in one place, but it concentrates risk: the identity provider account must be protected with strong MFA.

Passwords

Current NIST digital identity guidance (SP 800-63B, Revision 4, 2025) emphasizes **length and screening** over complexity rules and routine expiration:

Practice	Current guidance
Minimum length	At least **15** characters when a password is the only factor; at least **8** when used as part of MFA
Maximum length	Allow at least 64 characters
Composition rules (must include symbol, number, etc.)	**Should not be imposed**
Periodic forced changes	**Not required**; change when there is evidence of compromise
Screening	Check new passwords against lists of breached, common and context-specific passwords
Other	Allow paste (password managers); no password hints; limit failed attempts (rate limiting)

Passwords must be stored as **salted, slow hashes** (Chapter 11), never in plain text or with reversible encryption.

Keyspace

The number of possible passwords (the keyspace) = **(character set size) ^ length**. Length increases keyspace much faster than a larger character set.

Example:

Password type	Keyspace
4-digit PIN	10^4 = 10,000
6-digit PIN	10^6 = 1,000,000
8 lowercase letters	26^8 ≈ 2.09 × 10^11
8 characters from 94 printable characters	94^8 ≈ 6.10 × 10^15
16 lowercase letters	26^16 ≈ 4.36 × 10^22

Doubling the length of an all-lowercase password from 8 to 16 characters gives about **7.2 million times** more combinations than an 8-character password drawn from all 94 characters. Length wins.

Biometrics

Biometric systems make two kinds of error:

Error	Definition	Type
False acceptance rate (FAR) — Type II error	**False accepts ÷ impostor attempts**	**Security** failure: an impostor gets in
False rejection rate (FRR) — Type I error	**False rejects ÷ legitimate attempts**	**Convenience** failure: a valid user is turned away
Crossover error rate (CER, equal error rate)	The setting where FAR = FRR	Used to compare devices: **lower CER = more accurate**

Making matching stricter lowers FAR but raises FRR; loosening it does the opposite.

Example: In a pilot, a fingerprint system processed 10,000 legitimate attempts and 2,000 impostor attempts. It wrongly rejected 150 legitimate users and wrongly accepted 6 impostors.

1. FRR = 150 ÷ 10,000 = **1.5%**.
2. FAR = 6 ÷ 2,000 = **0.3%**.
3. Using all 12,000 attempts as the denominator would wrongly give FAR = 6 ÷ 12,000 = 0.05%, understating the security risk six-fold.

Memory trick: FAR lets the **f**raudster **a**rrive; **FRR r**ejects the **r**ightful user.

Authorization models

Model	How access is decided	Typical use
Role-based access control (RBAC)	Permissions assigned to roles; users get roles based on job	Most business applications and ERPs
Attribute-based access control (ABAC)	Rules evaluate attributes of user, resource and context (department, data classification, time, location)	Fine-grained and cloud access policies
Discretionary access control (DAC)	The data owner decides who gets access	File shares, personal folders
Mandatory access control (MAC)	System enforces labels (clearance vs classification); users cannot change them	Military and government systems
Rule-based access control	Global rules apply to everyone	Firewall rules, time-of-day restrictions

Core principles:

- **Least privilege:** give only the access needed to do the job.
- **Need to know:** access to specific data only when the job requires it.
- **Segregation of duties:** roles are designed so no single user holds incompatible permissions (for example, create vendor + approve payment). SoD conflict rule sets in ERP systems flag such combinations.
- **Default deny:** anything not explicitly allowed is denied.

The user access life cycle

Stage	Control
Joiner (provisioning)	Access requested and approved by the manager **and** the data or application owner; granted by security administration (not by the requester) based on a role
Mover (transfer)	Access for the new role added **and** old access **removed**, to prevent access creep
Leaver (deprovisioning)	Access disabled promptly — ideally automatically, triggered by the HR termination record; same day for involuntary terminations
Periodic review (recertification)	Owners or managers review user lists at set intervals (often quarterly for sensitive systems) and confirm or remove access

Example: An auditor compares the HR termination list for the year (64 terminated employees) with the ERP user list. Seven terminated employees still have active accounts, and logs show that two of those accounts were used after the termination date.

1. The seven active accounts are a **deprovisioning failure** — a deviation in the access ITGC.
2. The two used accounts are more serious: someone used credentials of people no longer employed. The auditor must find out what transactions were entered and whether they affect financial data.
3. Better design: automatic disablement triggered by HR, and a timely periodic access review that would have caught the leftovers.

Exam tip: A periodic access review is a **detective** control; prompt deprovisioning is a **preventive** control. The review is only effective if the reviewer has a complete and accurate user list and actually removes inappropriate access.

Privileged access management

Privileged accounts (system administrators, database administrators, domain admins, service accounts) can bypass application controls. Controls:

- Separate administrator accounts from everyday accounts.
- Store privileged credentials in a **password vault** with check-out, automatic rotation and session recording.
- **Just-in-time** elevation: grant admin rights for a limited time for an approved task.
- Log and independently review privileged activity (the administrator should not review their own logs).
- Restrict and monitor **emergency ("firefighter") access**; review every use.
- Service and bot accounts: unique, least privilege, non-interactive, credentials rotated.

Watch out: Shared administrator accounts ("admin" used by five people) destroy accountability. Even with logging, you cannot tell which person acted.

Physical and environmental controls

Logical controls are useless if someone can walk out with the server. Physical controls protect facilities and equipment.

Control	Classification	Notes
Fences, gates, locks, guards	Preventive (guards also detective)	Layered perimeter
Badge readers with logging	Preventive and detective	Ties entry to an individual
Mantrap (access control vestibule)	Preventive	Two interlocking doors; only one person passes at a time — defeats tailgating
CCTV	**Mainly detective** (deterrent when visible)	Records events for later review
Visitor logs and escorts	Preventive / detective	Visitors badged and accompanied
Asset tags and cable locks	Preventive / detective	Track and secure equipment

Environmental controls:

Hazard	Controls
Fire	Smoke detectors; clean-agent (gas) suppression for data centers; pre-action (dry-pipe) sprinklers that fill with water only after detection; fire extinguishers
Power	UPS, generators, surge protection
Temperature and humidity	Precision HVAC, sensors, alarms
Water	Raised floors, water sensors, location away from pipes and flood zones

> **Exam tip:** Data centers favor **clean-agent** suppression or **pre-action** sprinklers because a standard wet-pipe sprinkler can leak or discharge onto equipment. Halon is no longer produced because of its ozone impact.

Common traps

- Treating two knowledge factors as MFA.
- Using all attempts as the denominator for FAR or FRR.
- Calling CCTV a preventive control — it is mainly detective.
- Recommending forced 30- or 90-day password changes as current best practice.
- Removing new access but forgetting to remove old access when an employee transfers.
- Letting the person who requests access also approve or grant it.

Key takeaways

- Identification, authentication, authorization, accountability; accountability needs unique IDs.
- MFA = factors from different categories; phishing-resistant methods (security keys, passkeys) are strongest.
- NIST SP 800-63B Rev. 4: length (15+ characters for single-factor passwords) and breached-password screening; no composition rules or routine expiration.
- Keyspace = (character set)^length; length matters most.
- FAR = false accepts ÷ impostor attempts; FRR = false rejects ÷ legitimate attempts; lower CER = better device.
- RBAC is the business default; least privilege and SoD drive role design.
- Joiner–mover–leaver controls plus periodic reviews; privileged access needs vaulting, just-in-time elevation and independent log review.
- Mantraps stop tailgating; CCTV is detective; data centers use clean-agent or pre-action suppression.

11. Encryption, PKI and Network Security

This chapter covers the cryptographic tools that protect confidentiality, integrity and authenticity — symmetric and asymmetric encryption, hashing, digital signatures and public key infrastructure — and the network defenses that control traffic: firewalls, segmentation, IDS/IPS, VPNs, wireless security, monitoring and zero trust.

Cryptography vocabulary

Term	Meaning
Plaintext / ciphertext	Readable data / encrypted data
Algorithm (cipher)	The mathematical method (public and well-tested)
Key	The secret value that controls the algorithm
Key length	Size of the key in bits; each extra bit **doubles** the keyspace
Encryption at rest	Protects stored data (disks, databases, backups)
Encryption in transit	Protects data moving over networks (TLS, VPN)

Example: A 256-bit key has 2^256 possible values; a 128-bit key has 2^128. Going from 128 to 129 bits doubles the keyspace (2^129 ÷ 2^128 = 2). Going from 128 to 256 bits multiplies it by 2^128, about **3.4 × 10^38**.

Symmetric encryption

One **shared secret key** both encrypts and decrypts. Symmetric algorithms such as **AES** (Advanced Encryption Standard, 128-, 192- or 256-bit keys) are fast and suit bulk data.

The weakness is **key distribution**: each pair of parties needs its own secret key, delivered securely.

Number of keys for n parties to communicate pairwise = n(n − 1) ÷ 2

Asymmetric (public key) encryption

Each party has a **key pair**: a **public key** shared with anyone and a **private key** kept secret. What one key encrypts, only the other can decrypt. Algorithms include **RSA** and **elliptic curve cryptography (ECC)**, which achieves similar strength with shorter keys.

Asymmetric encryption is slow but solves key distribution: **total keys for n parties = 2n** (one pair each).

Example: How many keys are needed for secure communication?

Parties (n)	Symmetric: n(n − 1) ÷ 2	Asymmetric: 2n
10	10 × 9 ÷ 2 = **45**	**20**
50	50 × 49 ÷ 2 = **1,225**	**100**
200	200 × 199 ÷ 2 = **19,900**	**400**

Watch out: Do not forget to divide by 2 in the symmetric formula — the key Alice shares with Bob is the same key Bob shares with Alice.

Which key does what?

Goal	Sender uses	Receiver uses
Confidentiality (only Bob can read)	Bob's **public** key to encrypt	Bob's **private** key to decrypt
Authenticity / non-repudiation (prove Alice sent it)	Alice's **private** key to sign	Alice's **public** key to verify

Memory trick: "**Encrypt to** the receiver's **public**; **sign with** your own **private**." Nobody ever uses someone else's private key — they don't have it.

Hybrid encryption

In practice, systems combine both: asymmetric cryptography (or a key-agreement method) securely establishes a random **symmetric session key**, and the fast symmetric key encrypts the actual data. **TLS** — the protocol behind HTTPS — works this way.

Hashing

A hash function turns input of any size into a fixed-length digest (for example, **SHA-256** produces 256 bits). Good hash functions are:

- **One-way** — you cannot recover the input from the digest.
- **Deterministic** — the same input always gives the same digest.
- **Collision-resistant** — it is infeasible to find two inputs with the same digest.
- **Sensitive** — a one-character change produces a completely different digest.

Hashing provides **integrity**, not confidentiality. It is not encryption, because there is no key and no way to decrypt. MD5 and SHA-1 are considered broken for security purposes.

Password storage: store a **salted** hash using a deliberately **slow** algorithm (such as bcrypt, scrypt, Argon2 or PBKDF2). A unique salt per user defeats precomputed "rainbow tables" and makes identical passwords produce different hashes.

> **Watch out:** "The passwords were hashed, so they were encrypted" is wrong terminology. Hashing is one-way; encryption is reversible with the key.

Digital signatures

1. The sender hashes the message.
2. The sender encrypts (signs) the hash with the sender's **private key** — that is the signature.
3. The receiver decrypts the signature with the sender's **public key** to recover the hash.
4. The receiver hashes the received message and compares. A match proves **integrity** (message unchanged), **authenticity** (signed by the private-key holder) and **non-repudiation** (the sender cannot credibly deny signing).

A digital signature does **not** provide confidentiality — the message can still be read. To get both, sign with the sender's private key and encrypt with the receiver's public key.

Tool	Confidentiality	Integrity	Authenticity	Non-repudiation
Symmetric encryption	Yes	Limited	Limited (shared key)	No
Asymmetric encryption to receiver's public key	Yes	No	No	No
Hash	No	Yes	No	No
Message authentication code (keyed hash, HMAC)	No	Yes	Yes (between key holders)	No
Digital signature	No	Yes	Yes	**Yes**

Public key infrastructure (PKI)

How does Bob know a public key really belongs to Alice? PKI answers that.

Component	Role
Certificate authority (CA)	Trusted party that issues and signs **digital certificates** binding a public key to an identity
Registration authority (RA)	Verifies the identity of applicants before the CA issues certificates
Digital certificate (X.509)	Contains the subject's identity, public key, validity dates, issuer and the CA's signature
Certificate revocation	Certificates withdrawn before expiry (for example, if the private key is compromised) are published in a certificate revocation list (CRL) or checked online (OCSP)
Root and intermediate CAs	Chain of trust: browsers trust root CAs, which sign intermediate CAs, which sign server certificates

Exam tip: If a private key is stolen, the certificate must be **revoked**. Simply waiting for it to expire leaves the attacker able to impersonate the owner until then.

Key management

Keys must be generated securely, stored in protected hardware (**hardware security modules**, HSMs, or cloud key management services), rotated, backed up (encryption keys lost = data lost), separated from the data they protect, and destroyed securely. Split knowledge and dual control prevent one person from holding a master key alone.

Protecting payment and sensitive data

Technique	How it works	Reversible?
Encryption	Mathematical transformation with a key	Yes, with the key
Tokenization	Replaces the value with a random surrogate; the mapping is kept in a secure vault	Only through the vault
Masking	Hides part of the value for display (for example, showing only the last four card digits)	No (for the displayed copy)
Hashing	One-way digest	No

Exam tip: Tokenization can **reduce PCI DSS scope**: systems that store only tokens, never card numbers, may fall outside the cardholder data environment.

Network security

Firewalls and segmentation

Device / concept	Function
Packet-filtering firewall	Allows or blocks traffic by address, port and protocol
Stateful firewall	Tracks connections; allows return traffic for sessions it permitted
Next-generation firewall (NGFW)	Adds application awareness, user identity and intrusion prevention
Web application firewall (WAF)	Inspects web traffic to block application attacks such as SQL injection and XSS
Proxy server	Intermediary that hides internal addresses and filters content
DMZ (screened subnet)	Network zone between the internet and the internal network for public-facing servers (web, e-mail)
Network segmentation / microsegmentation	Splits the network into zones so an intruder cannot move freely (limits lateral movement)

Firewall rule sets should **deny by default**, be reviewed periodically, and be changed only through change management.

Detection and monitoring

Tool	Function
Intrusion detection system (IDS)	**Detects** and alerts; passive
Intrusion prevention system (IPS)	Sits inline and **blocks** malicious traffic; preventive
Signature-based detection	Matches known attack patterns; misses new attacks
Anomaly (behavior)-based detection	Flags deviations from normal; can catch new attacks but produces more false positives
Security information and event management (SIEM)	Collects and correlates logs from many sources; alerts and supports investigation
Endpoint detection and response (EDR / XDR)	Monitors endpoints for malicious behavior and can isolate devices
Honeypot	Decoy system to attract and study attackers
Data loss prevention (DLP)	Detects and blocks sensitive data leaving through e-mail, uploads or devices

> **Memory trick:** IDS = **D**etect (alarm). IPS = **P**revent (bouncer).

Secure remote access and wireless

- **VPN:** encrypted tunnel over a public network (IPsec or TLS-based); should require MFA.
- **Zero trust network access** increasingly replaces broad VPN access with per-application access decisions.
- **Wireless:** use WPA3 (or at least WPA2-Enterprise), change default settings, separate guest networks, and detect rogue access points. WEP is obsolete and insecure.

Zero trust

Zero trust (NIST SP 800-207) assumes **no implicit trust** based on network location. Every request is authenticated, authorized and encrypted, using identity, device health and context, with least-privilege, per-session access. "Never trust, always verify."

> **Watch out:** In zero trust, being "inside the corporate network" is **not** a reason to trust a request.

Common traps

- Encrypting with the sender's private key for confidentiality (that is a signature).
- Calling hashing "encryption."
- Counting $n(n - 1)$ symmetric keys instead of $n(n - 1) \div 2$.
- Expecting a digital signature to keep a message confidential.
- Choosing an IDS when the requirement is to block traffic (IPS).
- Letting a compromised certificate simply expire instead of revoking it.

Key takeaways

- Symmetric (AES): one shared key, fast, key distribution problem; keys = $n(n - 1) \div 2$.
- Asymmetric (RSA, ECC): key pairs, slow, solves distribution; keys = $2n$.
- Confidentiality: encrypt with the receiver's public key. Signature: sign with the sender's private key.
- Hash = integrity; digital signature = integrity + authenticity + non-repudiation; neither gives confidentiality.
- Store passwords as salted, slow hashes; protect keys in HSMs; revoke compromised certificates.
- Tokenization can shrink PCI DSS scope.
- IDS detects, IPS blocks, WAF protects web apps, DMZ hosts public servers, SIEM correlates logs.
- Zero trust: never trust based on location; verify every request.

12. Vulnerability Management and Incident Response

Preventive controls will never be perfect. Vulnerability management finds and fixes weaknesses before attackers use them; incident response limits the damage when something gets through. This chapter covers scanning, penetration testing, CVSS scoring and prioritization, patch and configuration management, the incident response life cycle, digital forensics, and the SEC's four-business-day incident disclosure rule.

The vulnerability management cycle

1. **Asset inventory** — know what you have (hardware, software, cloud resources). Unknown assets are never scanned or patched.
2. **Discover** — scan for known vulnerabilities and misconfigurations.
3. **Prioritize** — rank by severity, exploitability and asset importance.
4. **Remediate** — patch, reconfigure, or apply a compensating control; or formally accept the risk.
5. **Verify** — rescan to confirm the fix.
6. **Report and monitor** — track metrics such as time to remediate and overdue items.

Vulnerability scanning

Scan type	Description
Unauthenticated (non-credentialed)	Sees systems as an outsider would; finds exposed services
Authenticated (credentialed)	Logs in to inspect installed software and settings; finds far more and produces fewer false positives
External / internal	From the internet / from inside the network
Agent-based	Software on each host reports continuously

> **Exam tip:** Credentialed scans give a more complete and accurate picture of missing patches and insecure settings, which is why they are generally preferred for internal systems.

Scanners make mistakes: a **false positive** reports a vulnerability that is not really there (wasting effort); a **false negative** misses a real vulnerability (dangerous, because no one acts).

Penetration testing

A penetration test goes beyond scanning: authorized testers try to **exploit** vulnerabilities to show what an attacker could actually achieve. A written authorization ("rules of engagement") defining scope, timing and methods is essential — without it, testing is an attack.

Approach	Tester's knowledge
Black-box	None — simulates an outside attacker
Gray-box	Partial (for example, a normal user account)
White-box	Full — source code, architecture diagrams, credentials

Team	Role
Red team	Attackers simulating real adversaries
Blue team	Defenders who detect and respond
Purple team	Red and blue working together to improve detection

Other techniques: **bug bounty** programs (rewarding outside researchers), **social engineering tests** (simulated phishing) and **tabletop exercises** for incident response.

Prioritizing with CVSS and context

The Common Vulnerability Scoring System (CVSS) rates technical severity from 0.0 to 10.0. The qualitative bands are:

CVSS score	Severity
0.0	None
0.1–3.9	Low
4.0–6.9	Medium
7.0–8.9	High
9.0–10.0	Critical

> **Watch out:** Boundary scores belong to the **higher** band's lower edge: 7.0 is **High**, not Medium; 9.0 is **Critical**, not High; 4.0 is **Medium**.

The CVSS base score measures severity, not risk. Good prioritization adds context:

- **Known exploitation:** Is the vulnerability being actively exploited? CISA's **Known Exploited Vulnerabilities (KEV)** catalog lists those. Exploit-prediction scores (such as EPSS) estimate likelihood.

- **Exposure:** Is the system reachable from the internet?
- **Asset criticality and data sensitivity:** Does it process financial or personal data?
- **Compensating controls:** Is the vulnerable service blocked by a firewall or disabled?

Example: Three open findings, with capacity to fix only one this week:

Finding	CVSS	Severity	Context
A — internal print server	9.1	Critical	No known exploit; segmented network; no sensitive data
B — internet-facing payment web server	7.5	High	Listed in CISA KEV (actively exploited)
C — test server	5.3	Medium	Isolated lab

Fix **B** first. Although A has the higher score, B is internet-facing, processes payment data and is being actively exploited — the highest real-world risk. A is next; C can follow the normal patch cycle.

Patch and configuration management

- **Patch management:** obtain patches from trusted sources, test them, deploy within timelines set by severity (critical faster than low), track exceptions, and verify. Patches to production follow change management (Chapter 4); emergency patches follow the emergency change path.
- **End-of-life systems** no longer receive patches; isolate, add compensating controls, and plan replacement.
- **Configuration management:** apply secure baselines (for example, CIS Benchmarks), disable unnecessary services and default accounts, and monitor for **configuration drift**.
- **Hardening** reduces the attack surface: fewer services, ports, accounts and privileges.

Incident response

An **event** is any observable occurrence (a failed login). An **incident** is an event that actually or potentially jeopardizes the confidentiality, integrity or availability of systems or data, or violates security policy. A **breach** usually means unauthorized access to or disclosure of protected data — a legal term that can trigger notification duties (Chapter 13).

The classic life cycle

The long-standing NIST incident response life cycle has four phases:

Phase	Key activities
1. Preparation	Policy, plan, team (CSIRT) with defined roles, contact lists, tools, logging, training, tabletop exercises
2. Detection and analysis	Monitor alerts (SIEM, EDR, user reports), validate, determine scope, classify severity, document
3. Containment, eradication and recovery	Isolate affected systems; remove the cause (malware, compromised accounts); restore from clean backups; monitor closely
4. Post-incident activity	Lessons learned, root cause, update plans and controls, retain evidence

NIST SP 800-61 Rev. 3 (April 2025) reorganizes this guidance around the **CSF 2.0 Functions**: preparation activities fall under Govern, Identify and Protect, while incident handling maps to Detect, Respond and Recover, with continuous improvement throughout. The underlying activities are the same, so the classic phases remain the best way to answer "what comes next" questions.

> **Memory trick:** "**P**lease **D**on't **C**ry; **P**lan" — Preparation, Detection and analysis, Containment-eradication-recovery, Post-incident.

Containment choices

Short-term containment (disconnecting a host from the network, disabling an account, blocking an IP) stops the spread. Long-term containment may move services to clean systems while investigation continues. Containment should preserve evidence: **isolate rather than power off** where possible.

Digital forensics

- **Order of volatility:** collect the most volatile evidence first — CPU registers and cache, then RAM and running processes and network connections, then temporary files, then disk, then remote logs, then archives and backups.
- **Chain of custody:** document who collected, handled and stored each item and when, so evidence remains admissible.
- **Forensic images and hashes:** make bit-by-bit copies; record hash values to prove the copy is unchanged; analyze the copy, not the original.
- **Legal hold:** preserve relevant records once litigation is reasonably anticipated.

> **Watch out:** Powering off a compromised machine destroys memory-resident evidence (running malware, encryption keys, network connections). Capture volatile data first.

SEC cybersecurity incident disclosure

Public companies (SEC registrants) must disclose a **material** cybersecurity incident on **Form 8-K, Item 1.05**:

Rule	Detail
Trigger	Determination that a cybersecurity incident is **material**
Deadline	Within **four business days after the materiality determination** (not after discovery)
Timing of determination	Must be made **without unreasonable delay** after discovery
Content	Material aspects of the nature, scope and timing of the incident and its material impact or reasonably likely material impact
Delay	Permitted only if the U.S. Attorney General determines immediate disclosure poses a substantial risk to national security or public safety
Later information	Information not available at filing is added by an amended 8-K
Annual disclosure	Regulation S-K **Item 106** in Form 10-K: cyber risk management, strategy and governance

Example: A company discovers suspicious activity on Monday, October 6, 2025. After investigation, its disclosure committee determines on **Friday, October 17, 2025** that the incident is material.

1. The clock starts at the materiality determination, Friday, October 17.
2. Count four **business** days: Monday Oct. 20 (1), Tuesday Oct. 21 (2), Wednesday Oct. 22 (3), Thursday Oct. 23 (4).
3. The Form 8-K Item 1.05 is due by **Thursday, October 23, 2025**.

Counting from discovery, or counting calendar days (which would give Tuesday, October 21), would both be wrong.

Measuring the program

Metric	Meaning
Mean time to detect (MTTD)	Average time from compromise to detection
Mean time to respond / contain	Average time from detection to containment
Patch compliance	Share of systems patched within policy timelines
Overdue critical vulnerabilities	Count and age of critical items past due

Common traps

- Counting the 8-K deadline from discovery or in calendar days.
- Putting a boundary CVSS score (7.0, 9.0) in the lower band.
- Prioritizing by CVSS score alone, ignoring active exploitation and exposure.
- Powering off systems before capturing volatile evidence.
- Starting a penetration test without written authorization.
- Skipping the lessons-learned step after an incident.

Key takeaways

- Vulnerability management: inventory, scan (credentialed preferred), prioritize, remediate, verify.
- CVSS bands: Low 0.1–3.9, Medium 4.0–6.9, High 7.0–8.9, Critical 9.0–10.0; add exploitation (CISA KEV), exposure and asset value.
- Black-box = no knowledge, gray-box = partial, white-box = full; purple team = red + blue together.
- Incident life cycle: preparation; detection and analysis; containment, eradication and recovery; post-incident activity. SP 800-61 Rev. 3 aligns it with CSF 2.0.
- Forensics: order of volatility, chain of custody, hash-verified images.
- SEC 8-K Item 1.05: four business days after determining materiality; S-K Item 106 annual governance disclosure.

13. Privacy and Data Protection Rules

Security protects data from unauthorized access; **privacy** governs how personal information is collected, used, shared, kept and disposed of — even by authorized people. Privacy rules apply by **location** of the people, **sector** of the organization and **type** of data. ISC tests which rule applies, its key requirements and its deadlines and penalties.

Privacy foundations

Personal information categories

Term	Meaning
Personal information / personal data / PII	Information that identifies or can reasonably be linked to an individual
Sensitive personal information	Higher-risk data: health, biometrics, government IDs, financial account data, precise location, racial or ethnic origin, religion, sexual orientation
Pseudonymized data	Identifiers replaced with codes, with the key kept separately — **still personal data** because it can be re-identified
Anonymized data	Irreversibly de-identified so no one can be re-identified — outside most privacy laws

Watch out: Pseudonymized data is **not** anonymous. Under the GDPR it remains personal data, although pseudonymization is a recognized safeguard.

Common privacy principles

Most laws and frameworks share the same core ideas: **notice** (tell people what you collect and why), **choice and consent**, **collection limitation / data minimization**, **purpose limitation** (use data only for the stated purposes), **access and correction** rights, **retention limits**, **security**, **disclosure limits** for third parties, **data quality**, and **accountability**, including monitoring and enforcement. **Privacy by design** builds these into systems from the start, with privacy-protective defaults.

The AICPA and CIMA **Privacy Management Framework** (PMF) organizes privacy controls for management and assurance engagements, and privacy is one of the five Trust Services Categories in SOC 2 (Chapter 14).

GDPR (European Union)

The General Data Protection Regulation applies to:

- Organizations **established in the EU** that process personal data, wherever the processing happens; and
- Organizations **outside the EU** that offer goods or services to, or monitor the behavior of, people in the EU.

Role	Meaning
Data subject	The individual the data is about
Controller	Decides the **purposes and means** of processing
Processor	Processes data **on behalf of** the controller under a written contract
Supervisory authority	The national data protection regulator
Data protection officer (DPO)	Required for public authorities and for organizations whose core activities involve large-scale monitoring or large-scale processing of special-category data

Principles and lawful bases

GDPR principles: lawfulness, fairness and transparency; purpose limitation; data minimization; accuracy; storage limitation; integrity and confidentiality; and accountability (the controller must be able to **demonstrate** compliance).

Processing needs one of **six lawful bases**:

1. Consent (freely given, specific, informed, unambiguous; as easy to withdraw as to give)
2. Contract (necessary to perform a contract with the person)
3. Legal obligation
4. Vital interests (protecting someone's life)
5. Public task
6. Legitimate interests (balanced against the person's rights)

> **Memory trick:** "**C**an **C**ats **L**ick **V**ery **P**ointy **L**emons?" — Consent, Contract, Legal obligation, Vital interests, Public task, Legitimate interests.

Data subject rights

Access, rectification, erasure ("right to be forgotten"), restriction of processing, data portability, objection, and rights related to solely automated decision-making. Controllers generally respond within **one month**, extendable by two further months for complex or numerous requests.

Breach notification and fines

Requirement	Rule
Notice to the supervisory authority	Without undue delay and, where feasible, within **72 hours** after becoming aware of a personal data breach — unless the breach is unlikely to result in a risk to people
Notice to individuals	Without undue delay when the breach is likely to result in a **high** risk to them
Processor duty	Notify the controller without undue delay
Higher fine tier	Up to **€20 million or 4%** of total worldwide annual turnover of the preceding year, **whichever is higher**
Lower fine tier	Up to **€10 million or 2%**, whichever is higher

Example: A controller becomes aware of a reportable breach on **Monday, September 8, 2025 at 3:00 p.m.** 72 hours later is **Thursday, September 11, 2025 at 3:00 p.m.** — the target for notifying the supervisory authority. Note that the clock runs in hours, including the weekend if there is one, and starts at **awareness**, not when the breach occurred.

Example: Maximum higher-tier fine:

Worldwide turnover	4% of turnover	Fixed amount	Maximum fine (higher)
€900 million	€36 million	€20 million	**€36 million**
€300 million	€12 million	€20 million	**€20 million**

For a lower-tier violation at €900 million turnover: 2% = €18 million versus €10 million, so the maximum is **€18 million**.

Watch out: The GDPR cap is the **higher** of the two amounts, not the lower.

Cross-border transfers

Transfers of personal data outside the European Economic Area require a mechanism such as an **adequacy decision** (for example, the EU–U.S. Data Privacy Framework for certified U.S. companies), **standard contractual clauses** or **binding corporate rules**.

HIPAA (U.S. health information)

HIPAA applies to **covered entities** — health plans, health care clearinghouses and health care providers that conduct standard transactions electronically — and to their **business**

associates (vendors that handle protected health information, PHI, on their behalf), which must sign a **business associate agreement**.

Rule	Content
Privacy Rule	Uses and disclosures of PHI; patient rights; **minimum necessary** standard
Security Rule	Administrative, physical and technical safeguards for electronic PHI, based on risk analysis
Breach Notification Rule	Notices after a breach of **unsecured** PHI

Breach notice	Deadline
Affected individuals	Without unreasonable delay, and **no later than 60 calendar days** after discovery
Prominent media outlets	When **more than 500 residents of a state or jurisdiction** are affected; same 60-day limit
HHS — 500 or more individuals	Contemporaneously with individual notice (within 60 days)
HHS — fewer than 500 individuals	Logged and reported within 60 days after the end of the calendar year
Business associate to covered entity	Without unreasonable delay, no later than 60 days after discovery

> **Example:** A clinic discovers a breach of unsecured PHI on **March 3, 2025**. The outside date for individual notice is 60 calendar days later: **May 2, 2025**. "No later than 60 days" is a ceiling — notifying on day 59 without a reason for the delay can still be "unreasonable delay."

PHI encrypted to the HHS guidance standard is **"secured"**; if the keys were not also compromised, its loss is generally not a reportable breach. That is a strong argument for encrypting laptops and portable media.

> **Watch out:** HIPAA does **not** cover all health data. A consumer fitness app that is not a covered entity or business associate is outside HIPAA (though other laws, such as FTC rules and state laws, may apply).

GLBA (U.S. financial institutions)

The Gramm-Leach-Bliley Act covers "financial institutions" broadly — banks, but also lenders, mortgage brokers, investment advisers, many auto dealers and **tax preparers**

(including CPA firms that prepare tax returns).

- **Privacy Rule:** privacy notices to customers and opt-out rights for certain sharing with nonaffiliated third parties.
- **Safeguards Rule** (FTC version, as amended): a written information security program with a designated **qualified individual**, written risk assessments, access controls, **encryption** of customer information in transit and at rest, **MFA** for anyone accessing information systems, service provider oversight, testing (continuous monitoring or annual penetration tests and vulnerability assessments at least every six months), incident response planning, and a written report to the board at least annually.
- **FTC notification:** for a "notification event" involving unencrypted customer information of **500 or more consumers**, notify the FTC as soon as possible and **no later than 30 days** after discovery (effective May 13, 2024).

> **Example:** A tax practice discovers on **June 10, 2025** that an intruder accessed unencrypted records of 1,200 clients. Because 1,200 ≥ 500, it must notify the FTC no later than **July 10, 2025** (30 days). State breach laws may separately require notice to the clients and state regulators.

CCPA / CPRA (California)

The California Consumer Privacy Act, as amended by the California Privacy Rights Act, applies to for-profit businesses doing business in California that meet **any one** of these tests:

Threshold	Amount
Annual gross revenue	Above an inflation-adjusted threshold — **$26,625,000** from January 1, 2025 (originally $25 million)
Personal information volume	Buys, sells or shares the personal information of **100,000 or more** California consumers or households
Revenue from data	Derives **50% or more** of annual revenue from selling or sharing personal information

Consumer rights: to **know/access**, **delete**, **correct**, **opt out of the sale or sharing** of personal information (including for cross-context behavioral advertising), **limit the use of sensitive personal information**, and **non-discrimination** for exercising rights. Businesses generally respond within 45 days (extendable by another 45). The **California Privacy Protection Agency** enforces the law alongside the Attorney General; administrative fines are up to **$2,663** per violation or **$7,988** per intentional violation or violation involving minors under 16 (amounts adjusted January 1, 2025).

PCI DSS (payment cards)

PCI DSS is a contractual industry standard (from the PCI Security Standards Council, enforced through card brands and acquiring banks), not a law. **Version 4.0.1** is current; the future-dated requirements of v4.0 became mandatory on **March 31, 2025**. It contains **12 requirements** under six goals: build and maintain a secure network; protect account data; maintain a vulnerability management program; implement strong access control measures; regularly monitor and test networks; maintain an information security policy.

Data element	Storage after authorization
Primary account number (PAN)	Allowed only if needed, and must be rendered **unreadable** (encryption, truncation, tokenization, hashing)
Cardholder name, expiration date, service code	Allowed with protection
Sensitive authentication data — full track data, card verification code (CVV2/CVC2), PIN / PIN block	**Never** stored after authorization, even if encrypted

> **Exam tip:** The fastest way to reduce PCI DSS burden is to **not store card data**: use tokenization or a payment processor's hosted payment page so card numbers never touch your systems.

Choosing the right rule

Situation	Rule
U.S. company selling online to customers in France	GDPR (offering goods to people in the EU)
Hospital's billing vendor handling patient records	HIPAA (business associate)
CPA firm preparing individual tax returns	GLBA Safeguards Rule (and IRS security expectations)
Large retailer with California customers and $40 million revenue	CCPA/CPRA
Restaurant accepting credit cards	PCI DSS
Public company with a material breach	SEC Form 8-K Item 1.05, plus state breach laws

All 50 U.S. states also have their own **breach notification laws**, with differing definitions and deadlines; many exempt encrypted data.

Common traps

- Taking the **lower** of the GDPR fixed amount and percentage.
- Treating pseudonymized data as anonymous.
- Starting the GDPR 72-hour clock when the breach happened rather than at awareness.
- Applying HIPAA to non-covered consumer apps, or GLBA to restaurants.
- Thinking PCI DSS is a law, or that CVV2 may be stored if encrypted.
- Forgetting that CPA firms preparing tax returns are GLBA financial institutions.

Key takeaways

- GDPR: controllers and processors; six lawful bases; 72 hours to the authority after awareness; fines up to the higher of €20 million / 4% (or €10 million / 2%).
- HIPAA: covered entities and business associates; individual notice within 60 days; more than 500 residents of a state → media notice; 500+ → HHS at the same time; encrypted PHI is "secured."
- GLBA Safeguards Rule: qualified individual, written program, encryption, MFA; FTC notice within 30 days for events affecting 500+ consumers.
- CCPA/CPRA: revenue over $26,625,000 (2025 adjusted), 100,000+ consumers or households, or 50%+ revenue from selling or sharing data; rights to know, delete, correct, opt out and limit.
- PCI DSS v4.0.1: 12 requirements; never store sensitive authentication data after authorization.

14. SOC Engagements: Types and Planning

Many organizations outsource payroll, hosting, claims processing, payments and software to **service organizations**. Their customers (**user entities**) and the user entities' auditors need assurance about the service organization's controls. System and Organization Controls (SOC) engagements provide it. This chapter covers the report types, Type 1 versus Type 2, the parties and their responsibilities, subservice organizations, complementary controls and planning. Chapter 15 covers testing and reporting.

The parties

Party	Role
Service organization	Provides the outsourced service; its management prepares the **system description** and a written **assertion**, and designs and operates the controls
Service auditor	CPA who performs the examination and issues the opinion; must be **independent** of the service organization
User entity	Customer that uses the service
User auditor	Auditor of a user entity's financial statements; uses the SOC 1 report under **AU-C 402**
Subservice organization	A vendor used **by the service organization** to perform part of the service (for example, a data center used by a payroll provider)

Watch out: Management of the service organization — not the service auditor — prepares the description and the assertion. The auditor gives an opinion on them.

The SOC report family

Report	Standard	Subject	Intended users
SOC 1	SSAE 18, **AT-C 320**	Controls at a service organization **relevant to user entities' internal control over financial reporting (ICFR)**; control objectives specified by management	**Restricted:** management of the service organization, user entities and **user auditors**
SOC 2	**AT-C 205** using the **Trust Services Criteria** (TSP section 100) and description criteria (**DC 200**)	Controls relevant to **security, availability, processing integrity, confidentiality and/or privacy**	**Restricted:** management, user entities and others with sufficient knowledge (customers, prospects, regulators, business partners)
SOC 3	AT-C 205, Trust Services Criteria	Same as SOC 2, but a short report	**General use** — can be posted on a website; no description of tests or results
SOC for Cybersecurity	AT-C 205	An entity's cybersecurity risk management program (any entity, not only service organizations)	General use
SOC for Supply Chain	AT-C 205	Controls over the production, manufacture or distribution of goods	Customers and business partners

Memory trick: SOC 1 = 1 thing — the financial statements. SOC 2 = the 5 Trust Services Categories. SOC 3 = for everyone (3 = free).

Exam tip: A user **auditor** auditing financial statements wants a **SOC 1 Type 2**. A customer's IT security team evaluating a cloud vendor wants a **SOC 2 Type 2**. A prospect browsing the vendor's website gets a **SOC 3**. If detailed tests and results are needed, a SOC 3 is never enough.

Type 1 versus Type 2

	Type 1	Type 2
Point in time or period	**As of a specified date**	**Throughout a specified period** (commonly 6 to 12 months)
Opinion covers	Fairness of the description and **suitability of design** (and implementation as of the date)	Fairness of the description, suitability of design **and operating effectiveness**
Tests of operating effectiveness	No	**Yes**, with a description of tests and results
Usefulness to a user auditor who wants to rely on controls	Understanding only; **cannot** support reliance on operating effectiveness	Can support reliance, for the period covered

> **Memory trick: Type 1 = one day** (design). **Type 2 = two dimensions** (design + operating effectiveness over time).

Trust Services Criteria (SOC 2)

The **2017 Trust Services Criteria** (with revised points of focus issued in 2022) are organized into five categories:

Category	Focus	Required?
Security	Protection against unauthorized access, disclosure and damage — the **common criteria**	**Always included**
Availability	System is available for operation and use as committed or agreed	Optional
Processing integrity	Processing is complete, valid, accurate, timely and authorized	Optional
Confidentiality	Information designated as confidential is protected	Optional
Privacy	Personal information is collected, used, retained, disclosed and disposed of to meet objectives	Optional

The **common criteria** (CC1–CC9) apply to every SOC 2:

Series	Topic
CC1	Control environment
CC2	Communication and information
CC3	Risk assessment
CC4	Monitoring activities
CC5	Control activities
CC6	Logical and physical access controls
CC7	System operations
CC8	Change management
CC9	Risk mitigation (including vendor and business partner risk)

CC1–CC5 mirror the five **COSO** components. Additional criteria apply for the optional categories (A1 availability, PI1 processing integrity, C1 confidentiality, P1–P8 privacy).

> **Watch out: Confidentiality** protects any information designated confidential (trade secrets, business plans, customer contracts). **Privacy** applies specifically to **personal information** about individuals. A SOC 2 that covers confidentiality does not automatically cover privacy.

SOC 1 control objectives versus SOC 2 criteria

In a **SOC 1**, management **specifies its own control objectives** (for example, "Controls provide reasonable assurance that payroll changes are authorized and processed completely and accurately"). In a **SOC 2**, the criteria are the **AICPA's Trust Services Criteria**, which management cannot rewrite — it chooses which categories are in scope and designs controls to meet the criteria.

Subservice organizations: carve-out or inclusive

When the service organization relies on a subservice organization, the description must address it using one of two methods:

Method	What happens	Requirements
Carve-out	The subservice organization's controls are **excluded** from the description and the auditor's testing	The description identifies the subservice organization, the nature of its services and the **complementary subservice organization controls (CSOCs)** the service organization assumes are in place; user entities need separate assurance (often the subservice organization's own SOC report)
Inclusive	The subservice organization's relevant controls are **included** in the description and tested	The subservice organization's management must provide its **own written assertion**, and the service auditor needs access to test its controls

Example: CloudPay, a payroll service provider, hosts its application at a third-party data center. The data center will not allow CloudPay's service auditor on site.

1. Because the data center will not participate (no access, no assertion), the **inclusive method is not possible**.
2. CloudPay uses the **carve-out method**: its description names the data center, describes its hosting services and lists the CSOCs CloudPay assumes (for example, physical access restriction and environmental protection).
3. A user auditor relying on the CloudPay report should also obtain the **data center's SOC report** (or perform other procedures) to cover the carved-out controls.

Complementary user entity controls (CUECs)

A service organization's controls often achieve their objectives only if user entities do their part. Those **complementary user entity controls** are listed in the description. Examples:

- User entities review and approve payroll change reports before each payroll is processed.
- User entities promptly notify the service organization of terminated employees who had access to the system.
- User entities restrict and review access to the service's web portal for their own staff.

The user auditor must evaluate whether the user entity **actually performs** the relevant CUECs. If a CUEC is not in place, the service organization's controls may not protect that user entity even under an unmodified opinion.

Watch out: CUECs are performed by the **customers** (user entities). **CSOCs** are performed by **carved-out subservice organizations**. Do not mix them up.

Planning a SOC engagement

Preconditions and acceptance

Before accepting, the service auditor confirms:

- Independence from the service organization.
- The competence and capabilities to perform the engagement (including IT specialists).
- That management **acknowledges responsibility** for the description, the assertion, the selection of criteria (SOC 2) or control objectives (SOC 1), and the design and operation of controls.
- That the criteria are **suitable** and available to users.
- That management will provide a **written assertion**, access to evidence, and written representations.
- The scope: report type, Type 1 or Type 2, period or date, in-scope systems, categories (SOC 2), and the treatment of subservice organizations.

The terms are documented in an **engagement letter**.

> **Exam tip:** Under AT-C 320 (SOC 1), management's **written assertion** is required. If management refuses to provide it, the service auditor should **withdraw** where possible (and if withdrawal is not possible, disclaim an opinion).

Understanding and risk assessment

The service auditor obtains an understanding of the system — services, infrastructure, software, people, procedures and data — and assesses the risks that the description is not fairly presented and that controls are not suitably designed or operating effectively. **Materiality** in a SOC engagement is considered for the description (could an omission or misstatement influence users?) and for controls, mainly qualitatively.

The service auditor also:

- Reads relevant reports from internal audit and regulators.
- Inquires about fraud, noncompliance and uncorrected errors.
- Decides whether to use the work of the **internal audit function** (evaluating its objectivity and competence) — the service auditor remains solely responsible for the opinion.
- Plans tests of the ITGCs that support automated controls.

SSAE 21 changes

SSAE No. 21 (effective for reports dated on or after June 15, 2022) added **AT-C 206, direct examination engagements**, in which the practitioner measures the subject

matter against criteria without a responsible party's assertion, and amended AT-C 105 and 205. SOC 1 under AT-C 320 still requires management's written assertion.

Report periods and bridge letters

A user entity's fiscal year often does not line up with the service organization's report period. The gap is commonly covered by a **bridge letter** (gap letter) — a written representation from **service organization management** (not the service auditor) stating whether there have been material changes to the system or controls since the period ended.

> **Example:** A user entity's fiscal year ends **December 31, 2025**. The service organization's SOC 1 Type 2 report covers **October 1, 2024 – September 30, 2025**.
>
> 1. The report covers nine months of the user's fiscal year (January 1 – September 30, 2025).
> 2. The uncovered gap is **October 1 – December 31, 2025: 3 months**.
> 3. The user auditor obtains a **bridge letter from service organization management** for the gap and, depending on risk, performs additional procedures (for example, testing the user entity's own monitoring controls or inquiring about changes). The longer the gap, the less the report alone supports reliance.

> **Watch out:** A bridge letter is **not** audited and carries no opinion. It is management's representation and gives less assurance than the report itself.

Common traps

- Recommending a SOC 3 when detailed test results are needed.
- Choosing a SOC 2 for a user auditor's financial statement audit (SOC 1 is designed for ICFR).
- Thinking a Type 1 supports reliance on operating effectiveness.
- Believing the service auditor prepares the description or the bridge letter.
- Confusing CUECs (customers' controls) with CSOCs (carved-out vendors' controls).
- Assuming the inclusive method is possible without the subservice organization's assertion and access.
- Treating Security as optional in a SOC 2 — it is always in scope.

Key takeaways

- SOC 1 (AT-C 320) = controls relevant to user entities' ICFR; management-specified control objectives; restricted use.

- SOC 2 (AT-C 205, TSP 100, DC 200) = Trust Services Categories; Security (CC1–CC9) always included; restricted to knowledgeable users.
- SOC 3 = general-use version of SOC 2 without detailed tests.
- Type 1 = as of a date (description and design); Type 2 = throughout a period (adds operating effectiveness).
- Carve-out excludes subservice controls (list CSOCs); inclusive includes and tests them (needs the subservice organization's assertion).
- CUECs are performed by user entities and must be evaluated by user auditors.
- Management prepares the description, assertion and bridge letter; SSAE 21 added AT-C 206 direct examinations.

15. SOC Testing, Reporting and Opinions

This chapter finishes Area III. It covers how a service auditor tests controls in a Type 2 engagement, how deviations are evaluated and reported, the structure of a SOC report, the four opinion types and when each applies, and how a user auditor uses the report under AU-C 402.

Evidence for operating effectiveness

To conclude that a control operated effectively **throughout the period**, the service auditor gathers evidence about how it was applied, by whom and how consistently.

Procedure	Description	Sufficient alone?
Inquiry	Asking personnel how the control works	**Never**
Observation	Watching the control being performed	Limited to the moment observed
Inspection	Examining documents, approvals, logs, system settings	Often, combined with inquiry
Reperformance	Independently executing the control	Strongest evidence

> **Watch out: Inquiry alone is never sufficient** evidence of operating effectiveness. It must be combined with observation, inspection or reperformance.

Testing by type of control

Control type	Approach
Automated application control	Test **once** per relevant scenario (for example, one approved and one rejected transaction) — **provided** ITGCs over change management and access were effective throughout the period
IT general controls	Test across the period (for example, a sample of changes, new users and terminations); they support reliance on automated controls
Manual controls	Sample across the period; sample size increases with frequency of operation
IT-dependent manual controls	Test the manual review **and** the completeness and accuracy of the report used

Sample size by frequency

Professional standards do not prescribe sample sizes. Audit firms set them in their methodologies; a typical pattern looks like this:

Control frequency	Illustrative sample size
Annual	1
Quarterly	2
Monthly	2–5
Weekly	5–15
Daily	20–40
Many times per day	25–60

The principles to know: **more frequent** controls need **larger** samples; higher planned reliance and higher risk need larger samples; samples must be drawn from the **whole period**, not just the end.

Deviations

A **deviation** is an instance where a control was not performed as designed. The **deviation rate** is:

Deviation rate = number of deviations ÷ number of items tested

Example: The service auditor tests a daily reconciliation control by selecting 40 days from the 12-month period and finds 2 days without evidence of review.

1. Deviation rate = 2 ÷ 40 = **5.0%**.
2. The population (about 250 business days) is **not** the denominator; dividing 2 by 250 would understate the rate.
3. The auditor investigates the **cause** (staff turnover? system outage?), considers whether the deviations are isolated or systemic, and compares the rate with the tolerable rate, allowing for sampling risk.
4. If the deviations exceed what the plan tolerated, the auditor may extend testing or test other controls that address the same objective; if not enough, the control is not operating effectively.

Example: A different control was tested on 40 items with 1 deviation (2.5%). The auditor's methodology planned for zero expected deviations, so the auditor extends the sample. If the extended sample shows no further deviations and the cause was an isolated one-day absence, the auditor may still conclude the control operated effectively — but the deviation **must still be described** in the tests and results section.

Deviations and the opinion are separate questions

A deviation in one control does **not** automatically modify the opinion. The opinion asks whether each **control objective** (SOC 1) or **criterion** (SOC 2) was achieved. If other controls compensated, the objective can still be achieved — the opinion stays unmodified, but every deviation is still reported in the tests-and-results section so users can judge it themselves.

Structure of a SOC report

Section	Content	Prepared by
1	Independent service auditor's report (opinion)	Service auditor
2	Management's written assertion	Service organization management
3	Description of the system (including CUECs and, for carve-outs, CSOCs)	Service organization management
4	Control objectives (SOC 1) or criteria (SOC 2), related controls, the **service auditor's tests and results** (Type 2)	Controls by management; tests and results by the service auditor
5 (optional)	Other information provided by the service organization (for example, planned remediation, business continuity summary)	Service organization management — **not covered by the opinion**

> **Watch out:** "Other information" from management is **not** covered by the service auditor's opinion. The auditor reads it for material inconsistencies with the description.

Elements of the service auditor's report (Type 2)

- Title including the word "independent" and addressee.
- Scope: the description, the period, the control objectives or criteria; treatment of subservice organizations; reference to CUECs.
- Management's responsibilities and the service auditor's responsibilities; statement that the examination followed AICPA attestation standards; independence and ethics.
- Inherent limitations (controls may not prevent or detect all errors; projections to future periods are risky).
- Reference to the description of tests of controls and results.
- **Opinion** on the three matters: fair presentation of the description, suitability of design, operating effectiveness throughout the period.
- **Restricted use** paragraph (SOC 1 and SOC 2).
- Signature, city and state, date.

Opinion types

Opinion	When	Typical wording
Unmodified	Description fairly presented, controls suitably designed and operating effectively in all material respects	"In our opinion, in all material respects, ..."
Qualified	A misstatement or deviation is **material but not pervasive** (for example, one control objective not achieved), or a material but not pervasive scope limitation	"**Except for** the matter described ..., in all material respects ..."
Adverse	Misstatements or deviations are **material and pervasive**	"...the description is **not fairly presented** / controls were **not** suitably designed or operating effectively"
Disclaimer	The auditor **cannot obtain sufficient appropriate evidence** and the possible effects could be material and pervasive (for example, management refuses access)	"We **do not express an opinion** ..."

Situations that lead to modification:

- The description is not fairly presented (it omits significant controls or describes controls that do not exist).
- Controls are not suitably designed to achieve an objective or criterion.
- Controls did not operate effectively, so an objective or criterion was not achieved.
- A scope limitation prevents obtaining evidence.

Example: A SOC 1 Type 2 report on a payroll processor has eight control objectives. For the objective on authorizing payroll master file changes, the auditor found that 6 of 25 sampled changes had no approval, and no other control addressed the objective. The other seven objectives were achieved.

1. One objective was **not achieved** — material.
2. The problem is confined to one objective, so it is **not pervasive**.
3. Opinion: **qualified** ("except for" the effects on the payroll master file change objective).
4. If most objectives had failed, the opinion would be **adverse**. If management had refused access to the evidence for most controls, the result would be a **disclaimer**, not an adverse opinion.

> **Memory trick:** Material + isolated = **"except for."** Material + everywhere = **adverse.** Couldn't look = **disclaim.**

Other reporting issues

- **Subsequent events:** the service auditor inquires about events up to the report date that could affect the report and considers disclosing them.
- **Written representations:** management provides a representation letter. Refusal is a scope limitation.
- **Fraud or noncompliance** identified during the engagement is evaluated for its effect on the description and opinion and communicated to management.
- **Intentional misstatement of the description** or refusal to correct it may lead to an adverse opinion or withdrawal.

The user auditor: AU-C 402

A user auditor auditing a user entity's financial statements uses a SOC 1 report to understand and, where appropriate, rely on controls at the service organization. The user auditor evaluates:

Question	Why it matters
Is it a **Type 2**?	A Type 1 gives understanding only and cannot support reliance on operating effectiveness
Does the **period** cover the user's fiscal year?	Gaps need a bridge letter and possibly other procedures
Are the relevant **systems, locations and control objectives** in scope?	Out-of-scope controls give no assurance
Is the service auditor **competent and independent**?	Reliability of the report
What **deviations** were found, and what is the **opinion**?	Deviations in relevant controls may require extra work even under an unmodified opinion
Are the **CUECs** in place at the user entity?	The service organization's controls may depend on them
Are there **carved-out** subservice organizations?	The user auditor may need their SOC reports

Example: A user auditor receives a SOC 1 Type 2 report with an unmodified opinion. Section 4 shows 3 deviations in 40 samples for the control over approving rate changes to the user entity's billing data, compensated by another control the service auditor tested. One listed CUEC requires the user entity to review a monthly rate change report — and the user entity has not been doing so.

1. The unmodified opinion does not end the analysis: the deviations relate to a control relevant to the user's revenue.
2. The missing CUEC means part of the control system assumed by the report is not working at the user entity.
3. The user auditor should assess the effect on control risk, likely reduce reliance on that control, and perform additional substantive procedures on billing rates — and communicate the CUEC gap to the user entity's management.

Referring to the service auditor

In the user auditor's report:

- The user auditor **must not refer** to the service auditor's work in a report containing an **unmodified opinion**.
- If reference is relevant to understanding a **modification** of the user auditor's opinion, the report must make clear that the reference does not reduce the user auditor's responsibility.

Common traps

- Leaving deviations out of the report because the opinion is unmodified.
- Choosing adverse when the auditor could not obtain evidence — that is a disclaimer.
- Relying on inquiry alone as evidence of operating effectiveness.
- Computing the deviation rate with the population as the denominator.
- Treating "other information" as covered by the opinion.
- Relying on a Type 1 report for operating effectiveness.
- Ignoring CUECs because the service auditor's opinion was clean.

Key takeaways

- Type 2 testing: inquiry plus observation, inspection or reperformance; inquiry alone is never enough.
- Automated controls: test once per scenario if ITGCs are effective; manual controls: sample size rises with frequency.
- Deviation rate = deviations ÷ items tested; investigate cause; all deviations are reported in Section 4.

- Opinion modification depends on whether objectives or criteria are achieved, not on individual deviations.
- Qualified = material, not pervasive ("except for"); adverse = material and pervasive; disclaimer = cannot obtain evidence.
- Report sections: auditor's report, management's assertion, system description, controls with tests and results, optional other information (not covered by the opinion).
- AU-C 402: check type, period, scope, deviations, CUECs and carve-outs; never refer to the service auditor in an unmodified user auditor's opinion.

Final Review Checklist

Use this checklist in the last week before your exam. For each line, ask yourself whether you could explain the rule and work a quick example without notes. Anything you cannot do quickly, reread in its chapter and drill in the matching bank chapter.

Numbers and lists sheet

Item	Value
Exam format	82 MCQs (41 + 41), 6 TBSs (1, 3, 2); MCQs 60% / TBSs 40%; 4 hours; pass 75
Content areas	I Information systems and data 35–45%; II Security, confidentiality and privacy 35–45%; III SOC 15–25%
COSO 2013	5 components, 17 principles; Principle 11 = general controls over technology
NIST CSF 2.0	Govern, Identify, Protect, Detect, Respond, Recover; Tiers: Partial, Risk Informed, Repeatable, Adaptive
COBIT 2019	40 objectives; EDM + APO, BAI, DSS, MEA
CIS Controls v8.1	18 controls; IG1, IG2, IG3
ISO/IEC 27001:2022	Annex A: 93 controls, 4 themes
PCI DSS v4.0.1	12 requirements; never store CVV2, full track or PIN after authorization
CVSS	Low 0.1–3.9, Medium 4.0–6.9, High 7.0–8.9, Critical 9.0–10.0
Deadlines	GDPR 72 hours; SEC 8-K 4 business days after materiality determination; GLBA FTC 30 days (500+ consumers); HIPAA 60 days
GDPR fines	Higher of €20M or 4% / €10M or 2%
CCPA thresholds (2025)	$26,625,000 revenue; 100,000 consumers or households; 50% of revenue from selling or sharing
Passwords (SP 800-63B Rev. 4)	15+ characters single-factor, 8+ with MFA; no composition rules; no routine expiry; breached-password screening
SOC 2 common criteria	CC1–CC9; CC1–CC5 mirror COSO

Formula sheet

Formula	Expression
Allowed downtime	Minutes in period × (1 − availability)
Serial / parallel availability	Multiply / 1 − product of failure rates
Availability from reliability	MTBF ÷ (MTBF + MTTR)
Recovery test	RTO + WRT ≤ MTD; worst-case data loss = backup interval
RAID usable disks	RAID 5: n − 1; RAID 6: n − 2; RAID 10: n ÷ 2
Benford	P(d) = log10(1 + 1/d)
z-score	(x − mean) ÷ standard deviation
Risk	SLE = AV × EF; ALE = SLE × ARO; control value = ALE before − ALE after − annual cost
Biometrics	FAR = false accepts ÷ impostor attempts; FRR = false rejects ÷ legitimate attempts
Keyspace	(character set size)^length
Keys	Symmetric n(n − 1) ÷ 2; asymmetric 2n
Deviation rate	Deviations ÷ items tested

Area I — Information systems and data management (Chapters 1–7)

☐ I can list the five NIST cloud characteristics and tell IaaS, PaaS and SaaS apart by who patches the operating system.

☐ I know the customer always owns its data, identities and access settings under the shared responsibility model.

☐ I can convert an SLA percentage into allowed downtime and compute serial and parallel availability.

☐ I can name the transaction cycles and the incompatible duties within each.

☐ I can compute a three-way-match payment and the three batch totals, and I know only a hash total catches a changed identifier.

☐ I can rank system conversion methods by risk (parallel lowest, direct highest).

☐ I know the controls needed for RPA bots, AI models and blockchain, and what a blockchain does not prove.

☐ I can list COSO's five components and say what Principle 11 covers.

☐ I can distinguish ITGCs from application controls and explain why ITGCs support a test of one.

☐ I can classify a control as preventive, detective, corrective, directive or compensating, and as automated, IT-dependent manual or manual.

☐ I can match each edit check to the error it catches and compute a Luhn check digit.

☐ I know developers must not migrate their own code to production.

☐ I can order the SDLC phases and tell analysis (what) from design (how).

☐ I can distinguish unit, integration, system, regression and user acceptance testing.

☐ I know the rules for normal, standard and emergency changes, and why to sample from the production change log.

☐ I can define BIA, RTO, RPO, WRT and MTD and test a plan with RTO + WRT ≤ MTD.

☐ I can choose between cold, warm and hot sites based on RTO and cost.

☐ I know which backup sets an incremental or differential restore needs.

☐ I can compute RAID usable capacity, and I know RAID is not a backup.

☐ I know the 3-2-1 rule and why immutable or offline copies matter against ransomware.

☐ I can tell data owner, steward and custodian apart.

☐ I can define each data quality dimension, especially validity versus accuracy.

☐ I can distinguish data warehouse, data mart and data lake (schema-on-write versus schema-on-read).

☐ I know which destruction method fits which media and that legal holds override retention schedules.

☐ I can name the four types of analytics and apply Benford's law and z-scores, including when Benford does not apply.

☐ I can define primary key, foreign key and referential integrity.

☐ I can identify 1NF, 2NF and 3NF violations.

☐ I can explain ACID and classify SQL commands as DDL, DML, DCL or TCL.

☐ I can predict the result of a query using WHERE, GROUP BY, HAVING, INNER JOIN and LEFT JOIN.

☐ I know COUNT(*) versus COUNT(column), and that AVG ignores nulls.

☐ I know parameterized queries are the primary defense against SQL injection.

Area II — Security, confidentiality and privacy (Chapters 8–13)

☐ I can list the six CSF 2.0 Functions and know that supply chain risk sits under Govern.

☐ I can describe CSF Tiers and Profiles, and I know NIST does not certify organizations.

☐ I can separate COBIT's governance domain from its management domains.

☐ I know CIS Control 1 is asset inventory and IG1 is essential cyber hygiene.

☐ I know ISO/IEC 27001 is certifiable and has 93 Annex A controls in 4 themes.

☐ I can tell policies, standards, procedures and guidelines apart.

☐ I can apply the four risk responses and compare residual risk with risk appetite.

☐ I can identify each social engineering attack, including BEC and pharming, and the control that fits.

☐ I can distinguish viruses, worms, Trojans, ransomware, rootkits and logic bombs.

☐ I can match application and network attacks to their primary defenses.

☐ I can compute SLE, ALE and the net value of a control.

☐ I can explain identification, authentication, authorization and accountability.

☐ I can recognize true MFA and the phishing-resistant methods.

☐ I can state current NIST password guidance and compute a keyspace.

☐ I can compute FAR and FRR with the correct denominators and explain the CER.

☐ I can compare RBAC, ABAC, DAC and MAC.

☐ I know the joiner–mover–leaver controls and the controls over privileged access.

☐ I can classify physical controls (mantrap preventive, CCTV detective) and data center fire suppression choices.

☐ I can choose the right key for confidentiality and for a digital signature.

☐ I can compute symmetric and asymmetric key counts.

☐ I can explain hashing, salting, digital signatures and what each provides.

☐ I can describe PKI components and when to revoke a certificate.

☐ I can distinguish encryption, tokenization, masking and hashing.

☐ I can distinguish firewall types, WAF, DMZ, IDS, IPS, SIEM and DLP, and I can explain zero trust.

☐ I can describe the vulnerability management cycle and prefer credentialed scans.

☐ I can place any CVSS score in its band and prioritize using exploitation, exposure and asset value.

☐ I can distinguish black-box, gray-box and white-box tests, and red, blue and purple teams.

☐ I can order the incident response phases and explain order of volatility and chain of custody.

☐ I can compute an SEC 8-K Item 1.05 due date in business days from the materiality determination.

☐ I can tell pseudonymized from anonymized data.

☐ I know GDPR's scope, roles, six lawful bases, rights, 72-hour rule and fine tiers.

☐ I know HIPAA's covered entities, business associates, 60-day rule, media and HHS thresholds, and the encryption safe harbor.

☐ I know GLBA covers tax preparers and the Safeguards Rule elements, including the 30-day FTC notice.

☐ I know the CCPA/CPRA thresholds and consumer rights.

☐ I know PCI DSS's 12 requirements and which card data may never be stored.

Area III — SOC engagements (Chapters 14–15)

☐ I can choose between SOC 1, SOC 2, SOC 3, SOC for Cybersecurity and SOC for Supply Chain for a given user.

☐ I know the standards: AT-C 320 for SOC 1, AT-C 205 for SOC 2 and SOC 3, AU-C 402 for user auditors, AT-C 206 for direct examinations.

☐ I can explain Type 1 versus Type 2 and why a Type 1 cannot support reliance on operating effectiveness.

☐ I can list the five Trust Services Categories and the nine common criteria series.

☐ I can explain carve-out versus inclusive and the requirements of each.

☐ I can tell CUECs from CSOCs and explain why user auditors must check CUECs.

☐ I know who prepares the description, the assertion and the bridge letter.

☐ I can compute a report-period gap and know what covers it.

☐ I know inquiry alone is never sufficient and how testing differs for automated, manual and IT-dependent manual controls.

☐ I can compute a deviation rate and explain why deviations are reported even under an unmodified opinion.

☐ I can list the sections of a SOC report and know that other information is not covered by the opinion.

☐ I can choose between qualified, adverse and disclaimer opinions.

☐ I know what a user auditor evaluates in a SOC 1 report and when reference to the service auditor is allowed.

The night before

☐ Reread every chapter's Key takeaways.

☐ Rewrite the numbers sheet and the formula sheet from memory.

☐ Work one SQL query, one ALE, one availability and one deviation-rate example without notes.

☐ Confirm your exam appointment, ID requirements and travel time.

☐ Sleep. Recall on a remembering-heavy section like ISC depends on rest.

Now practice it

Every chapter in this guide has matching practice tests, simulations and mock exams at **UPractice.org**.