CS Exit Ethiopia

Revision Outlines

Computer Science Exit Examination

1 selected course | 18 November 2026

Selected courses

01. Computer Security

https://csexitethiopia.com

abelasnake2003@gmail.com | Telegram: @Abechu7

Course contents

Each course begins on a new page. PDF bookmarks provide question, study-note section and revision-topic navigation.

Computer Security 3

Computer Security

Revision Outlines

CIA and the property actually harmed | Exam focus

- CIA: property actually harmed
  - Confidentiality: restricted disclosure
  - Credit-card copying violates secrecy
  - Intact original ≠ preserved confidentiality
  - Integrity: unauthorized/improper change prevention
  - Altered balance; inserted false message
  - Availability: timely usable authorized service
  - Crashes; resource starvation; denied access
  - One event may harm multiple goals
- Identity, evidence and information assurance
  - Authentication: identity or claim establishment
  - Authorization: permitted actions
  - Accountability: responsible entities linked to actions
  - Reliable identity and protected logs
  - Non-repudiation: origin/receipt evidence
  - Signatures ≠ physically impossible denial
  - Assurance: confidence, risk management, protection
  - Detection; response; restoration; information lifecycle
  - Data at rest, in use, in transit
  - Accidental corruption and intentional attacks

Threat, vulnerability, risk and control | Exam focus

- Asset, threat, vulnerability, attack, control
  - Assets: data, software, hardware, service, reputation
  - Threat: potentially harmful circumstance
  - Vulnerability: exploitable weakness
  - Attack: actual attempt
  - Control: reduced likelihood or impact

- Software flaw ≠ active attack
- Risk: likelihood and consequence
- Unmeasured labels ≠ universally meaningful product

• Control purposes and categories
- Preventive: block; detective: reveal
- Corrective/recovery: restore safe operation
- Deterrent: discourage attempts
- Encryption: disclosure prevention
- Audit log: detection and accountability evidence
- Physical; technical; administrative categories
- Categories ≠ competing stages
- Firewall cannot replace policy or locks

Attack classifications and attacker intent

• Attack effects and 4D
- Interception: observe/copy
- Modification: alter
- Interruption: block
- Fabrication: false insertion
- Disclosure; deception; disruption; usurpation
- Usurpation: unauthorized control
- DoS: disruption and availability loss
- DoS need not obtain control

• Passive versus active attacks
- Passive: unchanged target exchange
- Sniffing; traffic analysis
- Active: inject, alter, replay, disrupt
- Passive/active ≠ offline/online
- Encrypted traffic still exposes timing/size
- Traffic analysis infers relationships
- Neither universally easy to prevent/detect

• Attacker intent and trust position
- White hat: authorization and scope

- Black hat: malicious or unauthorized
- Competitor: commercial secrets
- Hacktivist: cause
- Hacker label ≠ legality
- Insider/outsider: origin and trust position
- Origin ≠ attack mechanism

Malware: propagation, disguise and payload

• Malware replication and disguise
- Virus: attaches to program/document host
- Infected-host execution spreads virus
- Worm: independent replication, commonly networked
- Trojan: useful disguise, malicious behavior
- Trojan need not self-replicate
- Labels describe different properties
- Package may contain multiple components

• Malware payloads and defenses
- Spyware: monitoring; keylogger: keystrokes
- Adware: unwanted ads, possible tracking
- Ransomware: encryption/locking and extortion
- Logic bomb: trigger-dependent
- Backdoor: ordinary-access bypass
- Rootkit: hidden presence/privileged activity
- Zombie/bot: remote commands; botnet: collection
- Malware: umbrella, not replication method
- Update; restrict execution; isolate compromise
- Tested offline backups
- Antivirus never guarantees complete protection

Social engineering, spoofing and sessions

• Social engineering
- Phishing: trusted-source impersonation
- Secret theft or induced unsafe action

- Spear phishing: specific person
- Verify through independently known channel
- Shoulder surfing: observed entry
- Pretexting: invented disclosure reason
- Authorized monitoring ≠ necessarily attack

• Spoofing, replay and session attacks

- Masquerade: another identity
- ARP poisoning: local IPv4-to-MAC associations
- DNS spoofing/cache poisoning: false resolutions
- IP spoofing: false source address
- Intermediary: interception and possible alteration
- Replay: reuse valid old message
- Replay defenses: nonces, timestamps, sequence checks
- Session hijacking: stolen/predicted valid token
- Protect transport; regenerate IDs after login
- Restricted cookies; per-request server authorization

DoS, wireless attacks and layer-specific defenses

• Resource-exhaustion attacks

- DoS: exhaust or disable resource
- DDoS: multiple sources
- Bots; reflected/amplified responses
- SYN flood: incomplete-handshake state
- DNS flooding: query capacity
- NXDOMAIN: nonexistent-name resolver stress
- Fragment floods: reassembly/processing exhaustion
- Rate limits; SYN cookies; filtering
- Upstream capacity controls
- Defense depends on bottleneck

• Wireless attacks and defense layers

- Rogue AP: imitate/bypass approved service
- Bluesnarfing: Bluetooth information theft
- Bluejacking: unsolicited Bluetooth messages

- Phishing not wireless-specific
- Authenticated IPsec: IP impersonation resistance
- DNSSEC: DNS-data authentication
- TLS: application-transport authentication/protection
- Bounds-safe code: overflow prevention
- HTTPS cannot repair unsafe memory copy

Identification, authentication and factors | Exam focus

- Access-control sequence and direction
  - Identify → authenticate → authorize → record
  - Identification: username/claim, not proof
  - Identification omitted: authentication first available stage
  - User-to-host: user proves identity to host
  - Host-to-user: host proves identity to user
  - Arrow: prover → verifier
- Authentication factors
  - Knowledge: password/PIN
  - Possession: token/card
  - Inherence: biometric, fingerprint
  - Behavioral: typing rhythm, handwriting
  - Textbook category: something one does
  - One biometric ≠ three factors
  - Password plus PIN: one factor class
  - Card plus PIN: two factors
  - OTP factor depends on generation/delivery

Passwords and delegated authorization

- Password attacks and protection
  - Brute force: enumerate guesses
  - Dictionary: prioritize likely words
  - Credential stuffing: reuse leaked credentials
  - Online: server rate limits
  - Offline stolen verifiers: no server limits

- Short secrets vulnerable in either setting
- Unique long secrets; password manager
- Rate limiting; suitable MFA
- Salted slow hashes, not reversible plaintext
- Public salt defeats shared precomputed lookup

• OAuth authorization framework
- Client: limited access tokens
- Authorization server: authentication/owner approval
- Resource server: protected-data token validation
- Authorization server may delegate identity checks
- OAuth alone ≠ complete login identity protocol
- OpenID Connect: added identity layer

Access control and least privilege | Exam focus

• Subjects, objects and ACLs
- Subjects request operations on objects
- Object ACL: allowed subjects/actions
- Network ACL: address, protocol, port matching
- Direct effects: classification and filtering
- ACL ≠ guaranteed availability/malware detection

• Access-control models
- DAC: owner grants permissions
- Mandatory: enforced labels and policy
- Ordinary owners cannot override mandatory policy
- RBAC: permissions→roles; users→roles
- ABAC: subject/object/action/environment attributes
- Rule-based: explicit conditions
- Time or source-network rules

• Privilege restrictions
- Least privilege: needed rights, needed time
- Separation of duties: split sensitive steps
- Deny by default
- Mediate every access

- Revoke stale permissions
- SELECT allowed; UPDATE denied
- Access control supports integrity
- Business-state validity needs additional enforcement

Confidentiality and integrity models | Exam focus

- Bell-LaPadula confidentiality
  - Simple security: no read up
  - Star: no write down
  - Strong star: same-level writes only
  - TS>S>C>U
  - S subject may read C
  - S subject cannot write C
  - Compartments and DAC may further deny
- Integrity models
  - Strict Biba: no read down
  - Strict Biba: no write up
  - Prevent low-integrity input contaminating higher results
  - Clark-Wilson: certified well-formed transformations
  - Integrity checks; separation of duties
  - Clark-Wilson ≠ confidentiality model
  - MAC: enforcement category
  - Mandatory policy need not be Bell-LaPadula

Physical protection and resilient design

- Physical protection and threat models
  - Theft; unauthorized connections
  - Heat; humidity; water; surges; fire
  - UPS: temporary conditioned power
  - Faulty installation/battery: possible hazard
  - Disconnected machine: fewer network paths
  - Insiders; removable media; supply-chain attacks
  - Physical access remains a threat

- Most-secure claim requires threat model

• Resilient security design
  - Defense in depth: independent layers
  - Minimization: remove unnecessary software/services
  - Compartmentalization: limit blast radius
  - Fail securely: errors never silently grant
  - Uniformity: easier patching and prediction
  - Uniformity: correlated failures
  - Diversity: reduced common-mode compromise
  - Diversity: higher administration cost
  - Protection proportional to assets and threats

Cryptographic vocabulary and security assumptions

• Cryptographic vocabulary
  - Plaintext: original; ciphertext: encrypted representation
  - Key: algorithm input
  - C=E_k(P); P=D_k(C)
  - Symmetric five: plaintext, algorithm, secret key
  - Ciphertext and decryption algorithm
  - Cryptanalysis: unauthorized recovery or breaks
  - Cryptology: cryptography plus cryptanalysis
  - Steganography: hidden existence
  - Encryption: hidden meaning

• Cryptographic security assumptions
  - Known algorithm; protected keys
  - Implementation secrecy not main argument
  - b-bit key → 2^b possibilities
  - Uniform exhaustive-search average: half keyspace
  - Algorithm; key size; randomness
  - Mode and implementation matter
  - Symmetric ≠ inherently less secure
  - Speed ratios not fixed constants

Substitution, shift and affine ciphers

- Substitution versus transposition
  - Substitution: replace symbols
  - Transposition: rearrange positions
  - Monoalphabetic: one mapping throughout
  - Frequency leakage despite large nominal keyspace
  - Classical ciphers: teaching, not modern protection
- Caesar, multiplicative and affine arithmetic
  - Alphabet mapping: A=0…Z=25
  - Caesar: C=(P+k) mod26
  - Inverse: P=(C−k) mod26
  - k=3: A→D; HELLO→KHOOR
  - k=25: ATTACK→ZSSZBJ
  - Wrap negatives into 0-25
  - Multiplicative: C=kP mod26; gcd(k,26)=1
  - Decrypt with modular inverse of k
  - Affine: C=aP+b mod26; gcd(a,26)=1
  - Inverse relation, not repeated numeric multiplication

Vigenère, Playfair and Hill

- Vigenère and autokey
  - C_i=(P_i+K_i) mod26
  - Repeated or otherwise supplied shifts
  - Same letter may encrypt differently
  - Repeated short keys leak periodicity
  - Autokey extends initial key with message
- Playfair method
  - Keyed 5×5 grid; commonly merged I/J
  - Encrypt letter pairs
  - Specify duplicate fillers and padding
  - Same row: right
  - Same column: down
  - Rectangle: other column, original row

- Decrypt by reversed shifts
- Hill cipher and known plaintext
  - Letter vectors: C=KP mod26
  - Invertible K modulo26
  - gcd(det(K),26)=1
  - Linear structure
  - Enough independent known blocks reveal key
  - K=CP^-1 mod26
  - Claimed known-plaintext resistance: false
  - Polygraphic substitution ≠ modern strength

Transposition and one-time pads

- Transposition procedures and conventions
  - Rail fence: specified zigzag, read rails
  - Columnar: write rows, ordered columns
  - Symbols and counts preserved
  - Specify rows; column order; padding; direction
  - Inconsistent source-example conventions
  - Two rails ABCDEF: ACE and BDF
  - Ciphertext ACEBDF
- One-time pad conditions
  - XOR with independent uniformly random secret
  - Key at least message length
  - Never reuse key
  - Perfect secrecy requires every condition
  - Reuse: C1 XOR C2=P1 XOR P2
  - Vernam XOR ≠ automatically OTP
  - Predictable/reused keystream loses guarantee
  - Key delivery and storage costs

Symmetric keys, AES and distribution

- Symmetric keys and distribution
  - Shared secret; efficient bulk protection

- n users: n(n−1)/2 pairwise keys
- Trusted KDC changes architecture
- Session key: temporary communication context
- Master/key-encryption key: protects subordinate keys
- Physical distribution; authenticated protected channel
- Trusted third-party distribution
- Uncompromised old key protects new key
- Stolen old key prevents secrecy restoration

• AES and cipher construction

- AES block: 128 bits
- Keys: 128/192/256 bits
- Rounds: 10/12/14
- Substitution/permutation design
- SubBytes; ShiftRows; MixColumns; AddRoundKey
- Final round omits MixColumns
- Block cipher: mode and nonce/IV handling
- Authenticated encryption detects tampering
- Stream cipher: data plus keystream
- Nonce uniqueness when construction requires

DES and Feistel mechanics

• DES Feistel mechanics

- Block: 64 bits
- Key: 56 effective bits, eight parity
- 16 rounds; initial permutation
- $L_i=R_{(i-1)}$
- $R_i=L_{(i-1)} \text{ XOR } f(R_{(i-1)},K_i)$
- Right-half expansion: 48 bits
- XOR with 48-bit subkey
- Eight S-boxes: six→four bits each
- Permutation: 32 output bits
- S-box row: outer input bits
- S-box column: inner four bits

- Feistel inversion and DES limitations
  - Nonlinear S-boxes resist simple linear descriptions
  - Same structure, reversed round keys
  - Round function need not be invertible
  - Final swap/permutation
  - DES and triple DES: legacy
  - Not choices for new protection
  - Three 56-bit keys ≠ 168-bit effective security

Public keys and hybrid protection

- Public-key operation direction
  - Confidentiality to B: B's authenticated public key
  - B decrypts with B's private key
  - A's public key targets A, not B
  - A need not possess B's private key
  - A signs: A's private signing key
  - Others verify: A's public key
  - Signing and encryption are distinct operations
  - Not every public-key algorithm encrypts
- Hybrid protection and Diffie-Hellman
  - Public keys require authentic identity bindings
  - Public-key secret establishment/encapsulation
  - Symmetric bulk-data protection
  - Unauthenticated DH permits intermediary substitution
  - Public p,g; private a,b
  - A=g^a modp; B=g^b modp
  - B^a=A^b=g^(ab) modp
  - Authenticate exchange; derive separate traffic keys
- Sender key-pair requirements
  - One-way confidentiality: recipient authenticated public key
  - Sender key pair not inherently needed
  - Sender signature requires sender key pair
  - Separate recipient role may require pair

RSA arithmetic and key-validity conditions

- Textbook RSA construction
  - Distinct primes p,q
  - n=pq; φ=(p−1)(q−1)
  - 1<e<φ; gcd(e,φ)=1
  - d=e^-1 modφ
  - Public (n,e); private includes d
  - 0≤m<n
  - c=m^e modn; m=c^d modn
  - Large exponents: modular repeated squaring
  - Real RSA requires standardized padding
  - Raw arithmetic ≠ secure deployment
- RSA valid versus defective examples
  - p=7,q=19 → n=133,φ=108
  - Smallest admissible e>1: 5
  - e=7 and other coprimes also valid
  - e=5 → d=65
  - 325≡1 mod108
  - p=3,q=13,e=3,m=5
  - 5^3 mod39=8
  - gcd(3,24)=3 → no inverse
  - Arithmetic result ≠ valid RSA key
  - State defect; reject claimed key pair

Hashes, MACs and signatures

- Hash versus keyed MAC
  - Arbitrary input → fixed-length digest
  - Deterministic; preimage and collision resistance
  - Collisions necessarily exist
  - Not mathematically unique fingerprints
  - Unkeyed hash can be attacker-recomputed
  - Hash alone needs trusted reference
  - MAC: integrity, shared-secret authentication

- Both holders can create MAC
- Limited third-party non-repudiation

• Signatures and password hashes
- Private-key message/encoded-digest signing
- Public-key verification
- Signing ≠ generic private-key encryption
- DSA/ECDSA counterexamples
- Origin/integrity: identity and custody assumptions
- Signature does not conceal content
- Passwords: salted, deliberately expensive hash
- MD5/SHA-1 history ≠ modern collision-resistant choice

Certificates, TLS and HTTPS

• Certificate trust validation
- Issuer signature binds key to subject
- Chain to trusted root
- Intended hostname; time validity
- Applicable key usage and revocation policy
- Genuine certificate ≠ honest business

• TLS and HTTPS protection boundaries
- TLS succeeds obsolete SSL
- Authenticated protected channel
- Handshake: parameters, authentication, key establishment
- Server authentication; optional client authentication
- Records: symmetric authenticated encryption
- Messages vary with TLS version
- ChangeCipherSpec narrative not universal TLS1.3
- HTTPS=HTTP over TLS
- HTTP paths, headers, content protected
- IP endpoints/timing metadata remain visible
- Compromised decrypted endpoint remains exposed

SSH, IPsec, email protection and SET

- SSH, IPsec and VPN
  - SSH: remote login, commands, SFTP
  - Host keys; multiple client-authentication methods
  - IPsec: network-layer IP protection
  - AH: integrity/origin authentication, no encryption
  - ESP: confidentiality and integrity possible
  - IKE: security associations and keys
  - Transport mode: upper-layer payload
  - Tunnel mode: inner IP, outer header
  - VPN: logical private connection
  - Security depends on authenticated tunnel
- Email protection and SET
  - PGP/OpenPGP: symmetric content protection
  - Public-key key handling and signatures
  - S/MIME: certificate-based signed/enveloped MIME
  - Separate/clear signatures need not conceal
  - Envelope: encrypted content, wrapped recipient keys
  - SET: Secure Electronic Transaction
  - Visa/Mastercard-led card-transaction specification
  - Certificates and signatures
  - SET ≠ payment processor or ordinary HTTPS
- S/MIME sign-then-encrypt construction
  - Sign MIME entity
  - Package content and signature
  - Encrypt signed entity with fresh symmetric key
  - Protect key copy per recipient
  - Recipient: recover key → decrypt → verify
  - Validated signer certificate/public key
  - Signing ≠ generic private-key encryption
  - Other nesting orders possible
  - Specify order and protection coverage

Firewalls and protected boundaries | Exam focus

- Firewall types
  - Traffic-policy enforcement
  - Stateless filtering: source/destination, protocol, ports
  - Stateful inspection: permitted connection context
  - Application proxy: terminate, relay, inspect
  - Circuit gateway: connection relay, commonly SOCKS
  - No full application-content interpretation
  - Textbook session/transport association
  - Circuit gateway need not encrypt
  - Gateway ≠ automatically VPN
- Firewall placement and limits
  - Host firewall: individual machine
  - Network firewall: segmentation and DMZ
  - DMZ: public-facing services
  - Explicit ordered least-privilege rules; appropriate logging
  - Stateful/next-generation devices also filter ports
  - Port-filter question nonexclusive
  - Bypass paths; insiders; uninspected encrypted content
  - Infected removable media
  - Hardware firewall ≠ fixed port count

Detection, prevention and diagnostic measures

- Detection versus prevention
  - IDS: observe and alert
  - IPS: enforce blocking, usually inline
  - Host sensors: host events
  - Network sensors: traffic
  - Signature detection: known patterns
  - Anomaly detection: baseline deviations, possible novelty
  - Neither detects every attack
  - Anomaly false positives
  - Signatures can sometimes match variants

- Diagnostic measures and obfuscation
  - TP: real attack detected
  - FN: attack missed
  - FP: innocent activity flagged
  - TN: innocent activity accepted
  - Recall=TP/(TP+FN)
  - Precision=TP/(TP+FP)
  - 90/100 spam; five innocent flags
  - Recall=90%; precision=90/95≈94.7%
  - Obfuscation: harder code understanding/analysis
  - No reliable secret protection or vulnerability removal

Memory safety and secure input handling | Exam focus

- Buffer overflows and defenses
  - Write beyond allocated bounds
  - Data/control corruption; possible code execution
  - Execution neither definition nor inevitable outcome
  - Eight-byte array ≠ eight characters plus zero
  - Bounds-check lengths and indexes
  - Consider integer overflow
  - Safe abstractions; validate library returns
  - Canaries; non-executable memory; address randomization
  - Exploit mitigation ≠ repaired underlying write
- Trust-boundary validation and patches
  - Expected type, length, range, structure
  - Server validation: clients controllable
  - Destination-specific output encoding
  - Validation ≠ query parameterization
  - Validation ≠ HTML-context encoding
  - Patches: known weakness repair
  - Zero-day may lack vendor fix
  - Segmentation and mitigations still useful

SQL injection, XSS and request forgery | Exam focus

- SQL injection prevention
  - Unsafe strings let values alter syntax
  - Tautology changes matching rows
  - Prepared statements bind data values
  - Allow-list nonparameterizable identifiers
  - Least-privilege application database account
  - Stored procedures/ORMs require safe SQL
  - Unsafe dynamic SQL remains vulnerable
- XSS mechanisms and defenses
  - Attacker script under victim-site origin
  - Stored: persistent; reflected: response-returned
  - DOM-based: unsafe browser sinks
  - Safe text insertion
  - Context-specific encoding
  - Careful allowed-HTML sanitization
  - CSP: added protection, not sole fix
- CSRF, cookie limits and fuzzing
  - Ambient credentials enable unwanted authenticated action
  - Unpredictable request tokens
  - Suitable cookie/origin controls
  - HttpOnly: restricted script cookie access
  - HttpOnly ≠ all XSS prevention
  - Fuzzing: unexpected-input fault discovery
  - Testing technique ≠ injection category

Races, testing and secure development | Exam focus

- Race conditions
  - Timing/interleaving-dependent shared-state correctness
  - Check-then-open: replacement between check/use
  - Two workers read100, each writes90
  - Lost debit
  - Protect whole invariant

- Suitable atomic operations, locks, transactions
- Unrelated-variable lock ineffective

• Security testing and development
- Vulnerability assessment: find/prioritize weaknesses
- Authorized penetration test: demonstrate consequences
- Static: inspect code; dynamic: execute
- Fuzzing varies inputs
- Host/network/database scanners: different exposure
- False results possible
- Correct → patch → test → recheck
- Testing ≠ proof of absence
- Threat modeling; review; least privilege
- Dependency management; safe error handling

Trusted OS requirements and reference mediation | Exam focus

• Trusted operating system requirements
- Enforce stated security policy
- Identification/authentication; process and memory isolation
- Controlled privileges; object-access enforcement
- Protected auditing; safe resource reuse
- Trusted boot/update paths; recovery behavior
- TCB: policy-critical components
- Smaller analyzable boundaries simplify assurance

• Reference monitor and assurance
- Abstract access-checking mechanism
- Complete mediation; tamper resistance
- Small/simple enough for analysis
- Security kernel: core enforcement implementation
- Firewall mediates traversing traffic only
- Firewall ≠ every host-memory access
- Assurance: justified confidence in implementation
- Functionality: claimed control behavior
- GUI/multitasking alone ≠ trust

Independent evaluation, certification and assurance | Exam focus

- Common Criteria evaluation roles
  - TOE: Target of Evaluation
  - Specified configuration and environment
  - Security Target: threats, assumptions, objectives
  - Functional and assurance requirements
  - Protection Profile: reusable product-category requirements
  - Evaluator: evidence, testing, vulnerability analysis
  - Certification body: oversight/review and decision
  - Sponsor commissions; developer supplies product/evidence
- Evaluation Assurance Levels and scope
  - EAL: assurance-requirement package
  - EAL1-7: increasing evaluation rigor
  - EAL ≠ universal security-function ranking
  - Higher EAL across scopes ≠ safer
  - CCRA: mutual recognition, not worldwide laboratory
  - Certificate: specified product/configuration and claims
  - Certification ≠ future-attack immunity

Database secrecy, permissions and key boundaries | Exam focus

- Database protection and permissions
  - Confidentiality; integrity; availability; accountability
  - Authenticate connections
  - Roles: necessary table/column/row privileges only
  - Restrict views; audit sensitive changes
  - Report permission ≠ underlying full-table access
  - Application authorization plus database rights
- Database encryption and replication limits
  - At-rest encryption: stolen storage/backups
  - Keys must remain separate
  - Transport encryption: connection protection
  - Neither blocks fully authorized queries
  - Neither guarantees correctness

- Access controls, encryption, protected transmission
- Secrecy question may have multiple answers
- Replication: availability plus malicious-update copying
- Separate recovery controls still needed

Database integrity and reliability mechanisms | Exam focus

- Database integrity mechanisms
  - Valid data and valid transitions
  - NOT NULL: mandatory values
  - UNIQUE/primary key: duplicate-identifier prevention
  - Foreign key: reference preservation
  - CHECK: declared predicates
  - Atomic transaction: transfer debit and credit
  - Concurrency control: safe interleavings
  - Roles/ACLs: who may update
  - Constraints/transaction logic: authorized update validity
- Logging, backup and reliability
  - Write-ahead logging: recovery records before data
  - Crash recovery
  - Backups; restore tests; separately protected logs
  - Deletion/corruption recovery
  - Replication: alternate service, not backup replacement
  - Redundancy needs appropriate failover
  - Independent failure handling
  - Confidentiality versus reliability versus integrity constraints

Inference control and multilevel databases | Exam focus

- Inference control
  - Permitted outputs plus knowledge reveal secrets
  - Two-person salary total minus known salary
  - Blocked direct SELECT still leaks indirectly
  - Restrict small groups and overlapping queries
  - Track query history

- Coarsen/suppress results
- Carefully designed noise
- Utility-leakage tradeoffs

• Multilevel database security
- Different item labels; mandatory policy
- Subject clearances; classifications; compartments
- Bell-LaPadula-style read/write restrictions
- Prevent direct downward confidentiality flow
- Trusted declassification: separately controlled action
- Views alone ≠ multilevel isolation proof
- Covert channels; query results; error messages
- Assess direct access and indirect inference

Policy, incident response and recovery objectives

• Policy versus procedure
- Policy: responsibilities and rules
- Procedure: operational steps
- Coordinate protection; reduce risk
- Acceptable use; access review
- Incident handling; recovery
- Policy ≠ merely passive system use
- Awareness: reporting and independent verification
- Enforcement and usable procedures

• Incident response and recovery objectives
- Prepare roles/tools
- Detect/analyze → contain → eradicate
- Restore verified service; learn from evidence
- Preserve useful protected logs
- Disaster recovery: restore IT
- Business continuity: sustain business functions
- Workarounds and alternate facilities
- RTO: recovery-time target
- RPO: tolerable lost-data age

- Backup schedule meets recovery point
- Restore tests demonstrate recoverability

Interpret security mechanisms and conditions precisely

• Flawed security-question interpretation
- Overlapping/missing mechanisms: no invented unique answer
- Asymmetric question: multiple wrong key directions
- Signature ≠ generic private-key whole-message encryption
- Body biometric ≠ three factors
- SSL/SSH both describe protected channels
- Remote-shell scope distinguishes SSH
- Malformed option labels unrecoverable from theory

• Missing distinctions in security questions
- Arbitrary code execution: multiple mechanisms
- Execution consequence ≠ uniquely buffer overflow
- Database requirements: availability and confidentiality
- ACL/RBAC both restrict unauthorized writes
- Neither alone enforces all integrity constraints
- Evaluation organizations depend on national scheme
- CCRA: recognition, not evaluator
- Supported answer requires missing distinction