CS Exit Ethiopia

Complete Study Notes

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 6

How to use these study notes

START01 A practical study cycle

Start with a small set of questions before reading.

Mark the exact thing that stopped you: an unknown term, a rule, a calculation, a trace or a choice between alternatives.

Use the question lookup to open the relevant section.

Read one section, close it, and explain the rule in your own words.

Solve the question again and say why each other choice fails.

Then change one assumption or input and solve the variation.

This checks understanding beyond recognition.

A useful 60-minute session: 10 minutes of cold questions, 20 minutes of focused reading, 20 minutes of closed-book solving and 10 minutes updating an error log.

Adjust the timing to your needs.

Revisit mistakes after a gap, for example the next day, several days later and the following week.

These intervals are a practical starting plan, not a fixed learning law.

Record the section ID, your mistaken rule, the corrected rule and one boundary case.

Keep the log short enough to review.

Mix related courses during revision: programming with data structures; databases with security; networking with administration.

Explain how the same word changes meaning across contexts.

START02 What to emphasize

Emphasize definitions, key differences, algorithm costs, SDLC models, databases, networking and security.

Use short oral questions to check understanding and correct common misconceptions.

Combine those habits with the tracing, calculation, design and diagnosis demanded by the Ministry learning outcomes.

Allocate revision by both exam share and your error log.

A course with fewer items can still deserve more time if your foundations are weak.

The blueprint is a planning baseline, not a prediction of exact questions.

Definitions: distinguish nearby concepts, not just their names.

Calculations: write units, formulas, assumptions and intermediate steps.

Code and SQL: check validity before tracing the text; state any changes before analyzing a corrected version.

Algorithms and protocols: know the input, decision rule, state change, termination and failure cases.

Design choices: state the criterion.

A choice can change when cost, reliability, consistency or latency matters more.

Course group	Items in the 100-item blueprint	Revision implication
Web Programming	9	Reserve extra practice for HTML, CSS, JavaScript, PHP, client/server behavior and safe data handling.
Data Structures and Algorithms	7	Trace operations and compare their assumptions and costs.
Each of the other fourteen courses	6 each	Study all sixteen; do not abandon a whole course based on a short note or an alleged repeated-question list.

START03 How to handle exam traps

Underline NOT, FALSE, BEST and EXCEPT before comparing choices.

Check whether the stem supplies enough facts to select one answer.

A long scenario is not automatically an analysis question, and a familiar answer letter is not evidence.

When code is invalid, a diagram is absent, or two choices express the same rule, identify the clarification needed before answering.

Learn the correct rule and the smallest clarification needed.

Do not memorize an arbitrary key as a universal fact.

Distinguish worst case from average case, and an upper bound from a tight bound.

Check language version, SQL dialect, addressing assumptions, graph weights and tie-breaking rules.

For subnetting and scheduling, calculate rather than guess from a familiar number.

Use timed mixed papers only after you can explain your errors.

Revisit the reading sections behind recurring mistakes.

Computer Security

Complete Study Notes

SEC01 | CIA and the property actually harmed

Confidentiality restricts disclosure: copying a credit-card database violates secrecy even if the original stays intact.

Integrity restricts unauthorized or improper changes: altering a balance or inserting a false message damages correctness.

Availability supplies timely, usable service to authorized users: crashes, resource starvation and denied access violate it.

One event can harm several goals.

Authentication establishes an identity or other claim; authorization decides permitted actions.

Accountability links actions to responsible entities through reliable identity and protected logs.

Non-repudiation supplies evidence supporting origin or receipt; signatures do not make denial physically impossible.

Information assurance includes confidence, risk management, protection, detection, response and restoration across the information lifecycle.

Data security protects information at rest, in use and in transit, including accidental corruption as well as intentional attacks.

Computer Security module

SEC02 | Threat, vulnerability, risk and control

An asset is something valuable: data, software, hardware, service or reputation.

A threat is a potentially harmful circumstance; a vulnerability is an exploitable weakness; an attack is an actual attempt; a control reduces likelihood or impact.

A software flaw is not automatically an active attack.

Risk concerns likelihood and consequence, not a universally meaningful numeric product of two unmeasured labels.

Preventive controls block events, detective controls reveal them, corrective/recovery controls restore safe operation, and deterrents discourage attempts.

Encryption is preventive for disclosure; an audit log is principally detective/accountability evidence.

Physical, technical and administrative are control categories, not competing stages.

An external firewall cannot replace a written policy or a locked equipment room.

Computer Security module

SEC03 | Attack classifications and attacker intent

Interception copies or observes data, modification alters it, interruption blocks it, and fabrication inserts false material.

The 4D classification uses disclosure, deception, disruption and usurpation, the last meaning unauthorized control.

DoS primarily demonstrates disruption/availability loss; it need not give control of the victim.

Passive attacks observe without changing the target exchange: sniffing and traffic analysis.

Active attacks inject, alter, replay or disrupt.

These are not synonyms for offline and online.

Traffic analysis can infer relationships from timing/size even when content is encrypted.

Neither category is universally easy to prevent or detect.

White-hat testing requires authorization and scope; black-hat activity is malicious or unauthorized.

A competitor may seek commercial secrets; hacktivists pursue a cause. “Hacker” alone does not establish legality.

Insider versus outsider describes origin/trust position, not the specific mechanism.

Computer Security module

SEC04 | Malware: propagation, disguise and payload

A virus replicates by attaching to a host program or document; execution of the infected host spreads it.

A worm replicates independently, commonly across networks.

A Trojan disguises malicious behavior as something useful and does not require self-replication.

These labels describe different properties, so a malware package may contain several components.

Spyware monitors users; a keylogger records keystrokes.

Adware displays unwanted advertising and may also track users.

Ransomware denies access through encryption/locking and extortion.

A logic bomb waits for a trigger.

A backdoor bypasses ordinary access paths; a rootkit hides malicious presence or privileged activity.

A zombie/bot follows remote commands; a botnet is a collection.

Malware is the umbrella term, not a specific replication method.

Update, restrict execution, isolate compromise and maintain tested offline backups; no antivirus guarantees complete protection.

Computer Security module

SEC05 | Social engineering, spoofing and sessions

Phishing impersonates a trusted source to obtain secrets or induce an unsafe action; spear phishing targets a specific person.

Verify a request through an independently known channel.

Shoulder surfing observes entry; pretexting invents a reason for disclosure.

Monitoring software used with authorization is not necessarily an attack merely because it records activity.

Masquerade claims another identity.

ARP spoofing poisons local IPv4-to-MAC associations; DNS spoofing/cache poisoning supplies false name resolutions; IP spoofing falsifies a source address.

An intermediary attack intercepts and can alter exchanges.

A replay reuses a valid old message; nonces, timestamps and sequence checks can reject it.

Session hijacking steals or predicts a valid session token.

Protect transport, regenerate session IDs after login, restrict cookies and check authorization server-side on every sensitive request.

Computer Security module

SEC06 | DoS, wireless attacks and layer-specific defenses

DoS exhausts or disables a resource; distributed DoS uses multiple sources, often bots or reflected/amplified responses.

A SYN flood leaves many incomplete TCP handshakes occupying state.

DNS flooding targets query capacity; NXDOMAIN attacks generate queries for nonexistent names to stress resolvers.

Fragment floods abuse reassembly or packet-processing resources.

Rate limits, SYN cookies, filtering and upstream capacity controls address different bottlenecks.

Rogue access points imitate or bypass approved wireless service.

Bluesnarfing steals Bluetooth information; bluejacking sends unsolicited Bluetooth messages.

Phishing is not specific to wireless.

Match a defense to its layer: authenticated IPsec can resist IP impersonation, DNSSEC authenticates DNS data, TLS authenticates/protects application transport, and bounds-safe code prevents overflows.

HTTPS cannot fix an unsafe memory copy in a server.

Computer Security module

SEC07 | Identification, authentication and factors

A common access sequence is identify the subject, authenticate the claim, authorize the operation, then record relevant events.

Identification is the username/claim, not its proof.

When a sequence begins with authentication, distinguish it from the preceding identification stage.

User-to-host means a user proves identity to a host; host-to-user means the host proves its identity to a user; the arrow names the prover and verifier.

Knowledge is a password/PIN, possession is a token/card, and inherence is a biometric such as a fingerprint.

Behavioral biometrics include typing rhythm or handwriting; some textbooks call these “something one does.” A body characteristic is not “three-factor authentication”: one biometric is one factor.

Password plus PIN remains one factor class; card plus PIN is two.

An OTP’s factor depends on its delivery/generation mechanism.

Computer Security module

SEC08 | Passwords and delegated authorization

Brute force enumerates guesses; dictionary attacks prioritize likely words; credential stuffing tries previously leaked credentials on another service.

Online attempts face server rate limits; offline attacks against stolen verifiers do not.

Short passwords can fail in either setting.

Use unique long secrets and password managers, rate limiting and suitable MFA; store salted slow password hashes, not reversible plaintext passwords.

A salt prevents shared precomputed lookup, but is not secret.

OAuth is an authorization framework: a client obtains limited access tokens, the authorization server authenticates/obtains approval from the resource owner, and the resource server validates tokens for protected data.

The authorization server can delegate identity checks.

OAuth alone does not define a complete end-user login identity protocol; OpenID Connect adds an identity layer.

Computer Security module

IETF RFC 6749

SEC09 | Access control and least privilege

Subjects request operations on objects.

An ACL associates an object/resource with allowed subjects or actions; a network ACL matches traffic such as address, protocol and port.

Classification and filtering are direct effects; high availability or malware detection is not guaranteed by an ACL.

DAC lets owners grant permissions; mandatory control enforces labels and policy that ordinary owners cannot override.

RBAC gives permissions to roles and assigns users to roles.

ABAC evaluates subject, object, action and environment attributes.

Rule-based controls apply explicit conditions, such as time or source network.

Least privilege gives only needed rights, for needed time.

Separation of duties splits sensitive steps across parties.

Deny by default, mediate every access and revoke stale permissions.

Allowing SELECT but denying UPDATE is an access-control decision that helps integrity; it does not by itself enforce all valid business-state transitions.

Computer Security module

SEC10 | Confidentiality and integrity models

Bell-LaPadula protects confidentiality.

The simple-security rule prohibits reading above a subject's level; the star rule prohibits writing down.

The strong-star variant permits writing only at the same level.

With TS>S>C>U, an S subject can read C under the simple rule but cannot write C under star; compartments and discretionary restrictions may still deny access.

Strict Biba protects integrity by preventing read down and write up: untrusted lower-integrity input should not contaminate higher-integrity results.

Clark-Wilson protects integrity through certified well-formed transformations, integrity checks and separation of duties.

It is not a confidentiality model. “MAC” is an enforcement category and does not force every system to use Bell-LaPadula rather than another label policy.

Paul Krzyzanowski, Rutgers: Computer Security exam guide

SEC11 | Physical protection and resilient design

Protect equipment against theft, unauthorized connections, heat, humidity, water, electrical surges and fire.

A UPS supplies temporary conditioned power; it is normally protective, but faulty installation/battery failure can itself be hazardous.

A disconnected machine loses many network attack paths yet remains exposed to insiders, removable media, supply-chain attacks and physical access.

There is no universal “most secure computer” without a threat model.

Defense in depth layers independent controls.

Minimization removes unnecessary software and services; compartmentalization limits blast radius.

Fail securely means errors do not silently grant permission.

Uniform configurations simplify patching and prediction but create correlated failures; diversity may reduce common-mode compromise while raising administration cost.

Choose protection proportional to asset value and credible threats.

Computer Security module

SEC12 | Cryptographic vocabulary and security assumptions

Plaintext is the original message, ciphertext its encrypted representation, a key is input to an algorithm, encryption computes C=E_k(P), and decryption recovers P=D_k(C).

The five conventional symmetric ingredients are plaintext, encryption algorithm, secret key, ciphertext and decryption algorithm.

Cryptanalysis seeks recovery or other breaks without the authorized secret; cryptology includes cryptography and cryptanalysis.

Steganography hides that a message exists; encryption hides its meaning.

Assume the algorithm is known and protect keys: secrecy of implementation is not the main security argument.

A b-bit key has 2^b possibilities; exhaustive search averages roughly half the space if keys are uniformly random.

Security depends on algorithm, key size, randomness, mode and implementation.

Symmetric cryptography is not inherently less secure than public-key cryptography, and speed ratios are not fixed constants.

Computer Security module

SEC13 | Substitution, shift and affine ciphers

Substitution replaces symbols; transposition changes their positions.

A monoalphabetic substitution preserves one mapping throughout the message and leaks frequency patterns despite a large nominal key space.

Classical ciphers are teaching devices, not modern data protection.

With A=0 through Z=25, Caesar encryption is C=(P+k) mod26 and decryption P=(C-k) mod26.

A shift of 3 maps A→D and HELLO→KHOOR.

Key25 moves back one: ATTACK→ZSSZBJ.

Ensure negative results wrap into 0-25.

A multiplicative cipher uses C=kP mod26 with gcd(k,26)=1; decryption uses k's modular inverse.

Affine uses C=aP+b mod26 and requires gcd(a,26)=1.

The inverse-key relation, not simply the same numeric multiplication, recovers plaintext.

Computer Security module

SEC14 | Vigenère, Playfair and Hill

Vigenère repeats or otherwise supplies shifts K_i and computes C_i=(P_i+K_i) mod26.

The same plaintext letter may encrypt differently by position; repeated short keys still leak periodic structure.

An autokey scheme extends its initial key using message material.

Playfair uses a keyed 5×5 grid, commonly combining I/J, and encrypts letter pairs.

Specify duplicate-letter fillers and padding.

Same row moves right, same column moves down, and rectangle pairs take the other column in their original row; decryption reverses shifts.

Hill uses C=KP mod26 for letter vectors.

K needs an inverse modulo26, equivalently determinant coprime to26.

It is linear: enough independent known plaintext/ciphertext blocks reveal K=CP^-1 mod26.

Hill is vulnerable to known-plaintext attacks; polygraphic substitution alone is not modern strength.

Computer Security module

Chris Christensen, Northern Kentucky University: Hill cipher

SEC15 | Transposition and one-time pads

Rail-fence writes along a specified zigzag and reads rails; columnar transposition writes rows and reads columns in a specified order.

They preserve the symbols and their counts.

Always state rows, column order, padding and direction before computing a transposition cipher.

Two-rail zigzag for ABCDEF writes ACE on the first rail and BDF on the second, producing ACEBDF.

A one-time pad XORs bits with an independent uniformly random secret key at least as long as the message and never reused.

Its perfect secrecy depends on every condition.

Reusing a pad leaks C1 XOR C2=P1 XOR P2.

A Vernam XOR construction is not automatically a one-time pad: a predictable/reused keystream loses that guarantee.

Key delivery and storage remain substantial costs.

Computer Security module

SEC16 | Symmetric keys, AES and distribution

Symmetric encryption uses a shared secret and efficiently protects bulk data.

Pairwise independent keys for n users require n(n-1)/2 shared keys; a trusted key distribution center changes that architecture.

A session key is temporary for a communication context; a master/key-encryption key protects subordinate keys.

Distribute keys physically, via an authenticated protected channel, or through a trusted third party.

An uncompromised old key can protect a new key, although this does not restore secrecy after the old key is stolen.

AES is a 128-bit block cipher with 128/192/256-bit keys and 10/12/14 rounds.

Its substitution/permutation design uses SubBytes, ShiftRows, MixColumns and AddRoundKey; the final round omits MixColumns.

Block ciphers need a mode and correct nonce/IV handling; authenticated encryption also detects tampering.

A stream cipher combines data with a keystream.

Never reuse a nonce when the chosen construction requires uniqueness.

Computer Security module

SEC17 | DES and Feistel mechanics

DES has a 64-bit block, 56 effective key bits (eight other bits serve parity), and 16 Feistel rounds.

After initial permutation, each round computes L_i=R_(i-1), R_i=L_(i-1) XOR f(R_(i-1),K_i).

Its right-half expansion makes 48 bits, XOR mixes a 48-bit subkey, eight S-boxes each map six bits to four, and permutation rearranges the 32 output bits.

The outer S-box input bits select a row and the inner four select a column.

The nonlinear S-boxes resist simple linear descriptions.

Feistel inversion uses the same overall structure with reversed round keys; f need not itself be invertible.

Final swapping/permutation completes DES.

DES and triple DES are legacy algorithms, not choices for new protection.

Triple DES repeats DES operations but does not provide 168-bit effective security merely because three 56-bit keys are supplied.

Computer Security module

SEC18 | Public keys and hybrid protection

For confidentiality to B, encrypt with B’s authenticated public encryption key; B decrypts with B’s private key.

Do not encrypt to A’s public key if only B should read it, and never assume A possesses B’s private key.

For a signature, A signs with A’s private signing key and others verify with A’s public key.

These are distinct operations; not every public-key algorithm supports encryption.

Public keys ease distribution but require authentic bindings to identities.

Hybrid systems establish or encapsulate a symmetric secret with public-key mechanisms, then protect bulk data symmetrically.

Diffie-Hellman agrees on a secret but unauthenticated DH permits intermediary substitution.

With public p,g, A=g^a modp and B=g^b modp, both derive B^a=A^b=g^(ab) modp.

Authenticate the exchange and derive separate traffic keys.

For one-way public-key confidentiality the sender need not own a key pair; it needs the recipient’s authenticated public encryption key.

A sender key pair is needed when the protocol requires its signature or separate recipient role.

Computer Security module

SEC19 | RSA arithmetic and key-validity conditions

For textbook RSA choose distinct primes p,q; n=pq and φ=(p-1)(q-1).

Choose 1<e<φ with gcd(e,φ)=1, then compute d=e^-1 modφ.

Public key is (n,e), private includes d. Encrypt 0≤m<n by c=m^e modn; decrypt by m=c^d modn.

Use modular repeated squaring for large exponents.

Real RSA requires appropriate standardized padding; raw arithmetic is not a secure deployment recipe.

With p=7,q=19, n=133 and φ=108, e=5 is the smallest admissible e>1, but 7 and other coprime choices also work.

For e=5, d=65 since 325≡1 mod108.

With p=3,q=13,e=3,m=5, 5^3 mod39=8 arithmetically, but gcd(3,24)=3, so e has no inverse and this is not a valid RSA key.

Check key validity before treating arithmetic results as a working encryption/decryption pair.

Computer Security module

SEC20 | Hashes, MACs and signatures

A cryptographic hash maps arbitrary-length input to a fixed-length digest.

It is deterministic and designed to resist preimages and collisions; collisions must exist, so hashes are not mathematically unique fingerprints.

An attacker can recompute an unkeyed hash of modified data, so a hash alone needs a trusted reference.

A keyed MAC provides integrity and shared-secret authentication; both holders can create it, limiting third-party non-repudiation.

A signature algorithm signs a message or its encoded digest with a private key and verifies with the public key.

It is not generically “encrypting with a private key,” especially for DSA/ECDSA.

Verification checks origin/integrity under key-identity and key-custody assumptions; it does not conceal content.

Password verification uses a salt and deliberately expensive password hash.

Historical MD5/SHA-1 examples do not justify their use for new collision-resistant signatures.

Computer Security module

SEC21 | Certificates, TLS and HTTPS

A certificate binds a public key to a named subject through an issuer signature.

Validate the chain to a trusted root, intended hostname, time validity and applicable key usage/revocation policy.

A genuine certificate is not proof that a business is honest.

TLS succeeds obsolete SSL and supplies an authenticated protected channel.

A handshake negotiates parameters, authenticates the server and sometimes the client, and establishes keys; record protection uses symmetric authenticated encryption.

Exact messages depend on TLS version: the old ChangeCipherSpec narrative is not a universal TLS1.3 sequence.

HTTPS is HTTP over TLS.

It protects HTTP paths, headers and content on the channel, but does not hide all metadata such as IP endpoints and traffic timing, nor protect data after a compromised endpoint decrypts it.

Computer Security module

SEC22 | SSH, IPsec, email protection and SET

SSH protects remote login, commands and file transfer such as SFTP; it supports host keys and several client-authentication methods.

IPsec protects IP traffic at network layer.

AH provides integrity/origin authentication, not encryption; ESP can provide confidentiality and integrity; IKE negotiates security associations and keys.

Transport mode protects upper-layer payload; tunnel mode encapsulates an inner IP packet inside an outer header.

A VPN is a logical private connection; its actual security depends on the chosen authenticated tunnel.

PGP/OpenPGP combines symmetric content protection with public-key key handling and signatures.

S/MIME uses certificates for signed and/or enveloped MIME content.

Signatures can be sent separately/clear-signed without confidentiality; enveloped data encrypts content and wraps content keys for recipients.

SET, Secure Electronic Transaction, was a Visa/Mastercard-led specification for protecting card transactions using certificates and signatures; it is distinct from a payment-processing system or ordinary HTTPS.

One signed-and-encrypted S/MIME construction first signs the MIME entity, packages its content and signature, then encrypts that signed entity with a fresh symmetric content key.

Protect a copy of that key for each recipient.

A recipient recovers the key, decrypts the signed entity and verifies its signature using the validated signer certificate/public key.

Signing is not generic private-key encryption.

S/MIME permits other nesting orders; specify the order and resulting coverage.

Computer Security module

IBM KeyWorks service-provider interface

S/MIME4.0 specification

SEC23 | Firewalls and protected boundaries

A firewall enforces traffic policy.

Stateless packet filtering matches headers such as source/destination, protocol and ports.

Stateful inspection tracks permitted connection context; application proxies terminate and relay application exchanges and can inspect their content.

A circuit gateway relays connections, commonly SOCKS, without full application-content interpretation; textbooks associate it with session/transport functions.

Such a gateway need not encrypt and is not automatically a VPN.

Host firewalls protect individual machines; network firewalls segment networks and can create a DMZ for public-facing services.

Rules should be explicit, ordered as implemented, least-privilege and logged appropriately.

Stateful and next-generation devices can also filter ports; port filtering is not exclusive to one firewall category.

Bypass paths, insiders, encrypted uninspected content and infected removable media limit firewall protection.

Hardware firewall does not imply any fixed number of ports.

Computer Security module

SEC24 | Detection, prevention and diagnostic measures

An IDS observes and alerts; an IPS can enforce blocking, usually inline.

Host-based sensors examine host events, network-based sensors inspect traffic.

Signature detection recognizes known patterns; anomaly detection compares with a baseline and may detect novel behavior.

Neither guarantees every attack is detected; anomaly detection can raise many false positives, and signatures can sometimes match variants.

A true positive is a real attack detected, false negative is a missed attack, false positive is innocent activity flagged, and true negative is innocent activity accepted.

Recall=TP/(TP+FN); precision=TP/(TP+FP).

Detecting 90 of 100 spam messages with five innocent messages flagged gives recall90%, precision90/95≈94.7%.

Obfuscation makes code harder to understand and can hamper analysis, but does not reliably protect embedded secrets or remove vulnerabilities.

Computer Security module

SEC25 | Memory safety and secure input handling

A buffer overflow writes beyond allocated bounds and can corrupt data or control information; code execution is a possible consequence, not its definition or an inevitable outcome.

An array of eight bytes cannot store eight text characters plus a terminating zero.

Bounds-check lengths and indexes, consider integer overflow, use safe abstractions and validate library return values.

Canaries, non-executable memory and address randomization reduce exploitability but do not repair the underlying write.

Validate expected type, length, range and structure at trust boundaries.

Server validation is essential because clients are controllable.

Encode output for its destination; validation is not a universal substitute for query parameterization or HTML-context encoding.

Security patches repair known weaknesses; a zero-day may lack a vendor fix, but segmentation and other mitigations can still help.

MITRE CWE-787

SEC26 | SQL injection, XSS and request forgery

SQL injection occurs when untrusted values alter SQL syntax through unsafe string construction.

A tautology can change which rows match.

Bind data values in prepared statements, allow-list identifiers that cannot be parameters, and restrict the application’s database account.

Stored procedures/ORMs are safe only when they avoid unsafe dynamic SQL.

XSS causes attacker-controlled script to execute in a victim’s browser under the site’s origin.

Stored content persists; reflected content returns in a response; DOM-based XSS can arise from unsafe browser sinks.

Use safe text insertion and context-specific encoding; carefully sanitize allowed HTML.

CSP is added protection, not the sole fix.

CSRF exploits ambient credentials to submit an unwanted authenticated action.

Use unpredictable request tokens and suitable cookie/origin controls.

HttpOnly restricts script access to cookies but does not stop every XSS effect.

Fuzzing sends unexpected inputs to discover faults; it is a testing technique, not itself a category of injection.

OWASP SQL Injection Prevention Cheat Sheet

OWASP Cross Site Scripting

SEC27 | Races, testing and secure development

A race occurs when correctness depends on timing/interleaving over shared state.

In a check-then-open file operation, another actor may replace the file between the check and use.

In a balance update, two workers can both read100 and each write90, losing one debit.

Protect the whole invariant with suitable atomic operations, locks or database transactions; locking an unrelated variable does not help.

Vulnerability assessment finds and prioritizes weaknesses; authorized penetration testing attempts to demonstrate consequences.

Static analysis inspects code, dynamic testing executes it, and fuzzing varies inputs.

Host/network/database scanners see different exposures and have false results.

Correct findings, patch, test and recheck; testing is not proof of absence.

Build threat modeling, review, least privilege, dependency management and safe error handling into development.

Computer Security module

MITRE CWE-362

SEC28 | Trusted OS requirements and reference mediation

A trusted OS is relied on to enforce a stated security policy.

Relevant mechanisms include identification/authentication, process and memory isolation, controlled privileges, object access enforcement, protected auditing, safe resource reuse, trusted boot/update paths and recovery behavior.

The trusted computing base comprises components whose failure can violate policy; smaller, analyzable boundaries simplify assurance.

A reference monitor is an abstract access-checking mechanism: it must mediate every relevant access, resist tampering and be sufficiently small/simple to analyze.

A security kernel implements core enforcement.

A firewall can mediate its traversing traffic but cannot enforce every host-memory access.

Assurance is justified confidence that controls implement the claims; functionality is what controls claim to do.

A GUI or multitasking support alone does not establish trust.

Computer Security module

NIST SP800-53 Rev5 AC-25 Reference Monitor

SEC29 | Independent evaluation, certification and assurance

Under Common Criteria, define the Target of Evaluation (TOE), its configuration and environment.

A Security Target (ST) states threats, assumptions, objectives, functional requirements and assurance requirements; a Protection Profile describes reusable requirements for a product category.

Evaluators examine developer evidence, test and analyze vulnerabilities against those claims.

A certification body oversees/reviews the evaluation and issues the certification decision; sponsors commission work and developers supply product/evidence.

An Evaluation Assurance Level packages assurance requirements; EAL1-7 express increasing evaluation rigor, not a universal ranking of security functions.

A higher number does not make a product with different scope automatically safer.

CCRA is a mutual-recognition arrangement, not a single worldwide testing laboratory.

A certificate applies to a specified product/configuration and claims, not immunity to all future attacks.

Common Criteria 2022 Part 1

Indian Common Criteria Certification Scheme FAQ

SEC30 | Database secrecy, permissions and key boundaries

Database security needs confidentiality, integrity, availability and accountability together.

Authenticate connections, give roles only necessary table/column/row privileges, restrict views and audit sensitive changes.

A user allowed to read a report need not read the underlying full table.

Check application-level authorization as well as database rights.

Encryption at rest protects stolen storage/backups when keys remain separate; transport encryption protects the connection.

Neither prevents a fully authorized query or guarantees correctness.

Strong access controls, encryption and protected transmission can all contribute to secrecy; compare their roles and assumptions.

Replication is useful for availability but also copies malicious updates unless separate recovery controls exist.

Computer Security module

SEC31 | Database integrity and reliability mechanisms

Integrity requires valid data and valid transitions.

NOT NULL rejects missing mandatory values; UNIQUE/primary keys prevent duplicate identifiers; foreign keys preserve references; CHECK enforces declared predicates.

Transactions group a transfer's debit and credit atomically; concurrency control prevents unsafe interleavings.

Roles/ACLs limit who may update, but constraints and transaction logic determine whether an authorized update is valid.

Write-ahead logging persists recovery records before dependent data-file changes, supporting recovery after a crash.

Backups, restore tests and separately protected logs recover from deletion/corruption; replication provides alternate service capacity but does not replace backups.

Redundancy improves availability only with appropriate failover and independent failure handling.

Distinguish confidentiality controls from reliability controls and from integrity constraints.

PostgreSQL constraints documentation

PostgreSQL write-ahead logging documentation

SEC32 | Inference control and multilevel databases

Inference reveals protected facts by combining individually permitted outputs with other knowledge.

If a report gives a two-person salary total and one salary is known, subtraction reveals the other.

Blocking direct SELECT of a salary column does not stop that inference.

Restrict small groups and overlapping queries, track query history, coarsen/suppress results, or introduce carefully designed noise; each trades utility against leakage.

A multilevel database stores items at different labels and enforces mandatory policy using subject clearances, classifications and compartments.

Bell-LaPadula-style read/write restrictions prevent direct downward confidentiality flow; trusted declassification is a separate controlled action.

Views alone are not proof of multilevel isolation.

Covert channels, query results and error messages may still reveal information, so evaluate both direct access and indirect inference.

University of South Carolina, Information Security Lab

SEC33 | Policy, incident response and recovery objectives

A security policy states responsibilities and rules; a procedure supplies the steps.

Policies coordinate protection, reduce risk and define acceptable use, access review, incident handling and recovery.

They do not prescribe merely “passive system use.” Awareness training teaches reporting and independent verification; enforcement and usable procedures make it operational.

Prepare response roles/tools, detect and analyze, contain the incident, eradicate the cause, restore verified service and learn from the evidence.

Preserve useful protected logs.

Disaster recovery restores IT after disruption; business continuity sustains business functions through workarounds and alternate facilities.

RTO is a recovery-time target; RPO measures the tolerable age of lost data.

A backup schedule must satisfy the required recovery point, and a restore test must demonstrate that data can actually be recovered.

Computer Security module

SEC34 | Interpret security mechanisms and conditions precisely

Compare each security mechanism with the stated requirement and conditions.

Check asymmetric-encryption key directions carefully; signatures do not generically encrypt whole messages with private keys.

Body biometrics do not imply three factors.

SSL and SSH can both describe protected channels; SSH specifically supports remote-shell services.

Read the full option text before matching it to a security mechanism.

Undefined “arbitrary code execution” does not uniquely identify buffer overflow: several mechanisms can produce that consequence.

A database-security requirement can include both availability and confidentiality.

Generic ACLs and RBAC both restrict unauthorized writes but neither independently enforces all integrity constraints.

Evaluation organizations must be identified by the actual national scheme; CCRA is recognition, not an evaluator.

An answer is supported only after the missing distinction is supplied.

Computer Security module

SEC-S01 | Command injection and the data-to-command boundary

OS command injection occurs when an application lets untrusted input affect a command executed by the host operating system, often through unsafe construction of a shell command. The command runs with the application's effective privileges. This is an injection flaw in command interpretation, unlike fuzzing, which supplies varied inputs to discover defects.

Prefer a suitable library API over constructing a shell command. Where command execution is required, keep the executable and arguments under strict control and validate against the expected permitted structure; merely blocking a few suspicious characters is not a general proof of safety.

OWASP - Command Injection

SEC-S02 | WannaCry: ransomware and worm propagation

WannaCry, also known as WannaCrypt, is ransomware with worm-like propagation. It encrypts files and demands payment to restore access. Its name identifies malicious software rather than a network-security protocol. Malware names must be distinguished from defensive controls such as intrusion detection, intrusion prevention and physical protection.

Microsoft Defender Security Research Team - WannaCrypt ransomware worm targets out-of-date systems

CERT-EU - WannaCry Ransomware Campaign

SEC-S03 | Distinguish network roles and protocol names

A router forwards IP packets between networks; forwarding alone is different from detecting suspicious behavior or enforcing a firewall's security policy. SNMP means Simple Network Management Protocol and provides a framework for monitoring and controlling managed devices through management information. It is not a remote-shell service.

AAA names authentication, authorization and accounting functions, rather than a particular remote-shell channel. SSH expands to Secure Shell and provides secure remote login and related network services. PGP expands to Pretty Good Privacy; its message-protection design combines conventional and public-key encryption with digital signatures. Protocol purpose matters when several technologies all contribute to security.

RFC 1812 - Requirements for IP Version 4 Routers

RFC 3411 - SNMP Management Frameworks

RFC 2903 - Generic AAA Architecture

RFC 4251 - The Secure Shell Protocol Architecture

RFC 1991 - PGP Message Exchange Formats (historical terminology)