



VICTORIA UNIVERSITY OF
WELLINGTON
TE HERENGA WAKA

EXAMINATIONS – 2026

TEST 1

FRONT PAGE

CYBR 271

CODE SECURITY

Time Allowed: 50 minutes

Instructions: Mark all questions using the item-specific guidance. The paper is marked out of 50.

MARKING SCHEDULE – TUTOR USE

Assessed learning objective:

Apply a secure design methodology to identify and analyse potential security threats in various computing environments, using case studies to illustrate practical applications.

General marking rules:

- Award marks in increments of 0.5; do not use smaller fractions. The final script total may therefore end in .5.
- MCQs and Section B items are dichotomous: award 1 for the keyed or explicitly accepted answer and 0 otherwise. Do not award 0.5 for these items.
- For other 1-mark responses, award 1 when the requirement is met, 0.5 for a relevant but materially incomplete or underspecified response, and 0 for an incorrect, irrelevant or absent response. Apply the item-specific guidance below.
- For multi-mark responses, score each listed component independently. A component may receive half credit only where the guidance identifies partial evidence; never exceed the item maximum.
- Mark security competence, not prose polish. Accept technically sound alternatives and clear paraphrases; record an unlisted accepted answer for consistent use.
- For category/principle plus evidence items, score the classification and the scenario link independently: 1 for an accurate classification; 1 for a specific, correctly linked scenario fact; 0.5 where the fact is relevant but is not what enables the named category, or where the link is only implied. Exact quotation is not required.
- **Follow-through** applies to C1(b)(iii)–(iv), C3(b) and D1(c): credit reasoning consistent with a reasonable earlier answer and do not penalise the same error twice. C3(c) and D1(d) are independent items. The 2026 paper does not tell students that follow-through exists, so apply it generously.
- Where a prompt permits assumptions, accept a reasonable stated assumption. Do not import unsupported assumptions into the model answer.

“Different category / different principle” requirements (2026 wording):

- C1(a), C2(a) and D1(b) require the student’s answers to come from *different* categories or principles. This is stated in the question, so it is a marked requirement, not a marker convention.
- Where two answers fall in the same category or principle, mark the stronger on merit and award 0 for the duplicate. Do not spread the marks across both, and do not apply an overall cap.
- Judge the category by the security effect the student describes, not only by the label. Two answers labelled differently but describing the same effect from the same fact are a duplicate.
- Two answers in different categories may draw on the same scenario fact where the effects genuinely differ. Plain HTTP supports both information disclosure (reading) and tampering (altering); those are distinct.

Using the exemplars:

- Exemplars are *illustrative, not required wording*. An answer that reaches the same security point by a different route earns the same marks. Never reserve full marks for the exemplar you personally prefer.
- Exemplars are longer than a student could write in the time available. They show the full set of ideas being rewarded, not the expected length. A two-clause answer containing both ideas scores the same as a paragraph.

MARKING SCHEDULE

- **Second-order items** (C1(b)(ii)–(iv), C2(c), D1(d)) target a known cohort weakness: naming a control and stopping. A correct control with no consequence, adaptation or new dependency is a half-mark answer.
- **Trust boundary items** (D1(a), C2(a)) target the second known weakness. Accept any coherent account of why a mis-drawn boundary or element type misdirects the analysis; do not require the wording used here or recitation of the element-specific STRIDE table.
- Do not penalise spelling, British/American variants, or an abbreviation used in lectures (PnG, EoP, RCE, SCA, SBOM).

Moderation before marking:

- Each marker independently marks the same five scripts, then compares awards on C1(a), C2(b), C3(c) and D1(a). Reconcile against the exemplars before marking further scripts.
- Mark question by question across the pile, not script by script.
- Log any accepted answer not listed here in the shared sheet, with the item number, so it is applied to every script.
- Any script where D1 scores 0, or the total falls below 15, goes to the coordinator before marks are entered.

SECTION A – MULTIPLE CHOICE

(5 marks; suggested 4 minutes)

Select the single best answer.

A1. A shopping app sends an order to its server. An attacker changes the total price while the request is in transit. Which STRIDE category best fits? **(1 mark)**

- A. Spoofing
- B. Tampering**
- C. Information disclosure
- D. Elevation of privilege

Marking guidance: Award 1 mark for **B, Tampering**; 0 for any other option. The price is modified in transit; the attacker is not merely reading or impersonating.

Distractor note (queries and item analysis): A attracts students who read “changes the order” as impersonating the customer; C attracts students who assume anything on the wire is a confidentiality problem. The taught definition is decisive: tampering is the unauthorised modification of data, including alteration of data in transit over an open network.

A2. Every newly created user account receives administrator rights that are rarely removed. Which secure-design principle is most directly violated? **(1 mark)**

- A. Economy of mechanism
- B. Open design
- C. Least privilege**
- D. Defence in depth

Marking guidance: Award 1 mark for **C, Least privilege**; 0 for any other option. Default administrator rights exceed what ordinary users need.

Distractor note: D attracts students who treat “rarely removed” as a missing second layer. P8 is decisive: a subject is granted only the privileges needed for an operation, and operates at the lowest access level possible at any given application state.

A3. A seller denies having posted a listing. Which control would most directly support investigation and make the action difficult to deny? **(1 mark)**

- A. An integrity-protected audit log recording user, action and time**
- B. Requiring MFA before a seller can post a listing
- C. Encryption of listing photographs
- D. Nightly backups of the listings database

Marking guidance: Award 1 mark for **A**; 0 for any other option. The stem describes repudiation, whose taught control is non-repudiation: an accountable, tamper-resistant record of user, action and time.

Distractor note: *B is the diagnostic distractor for this cohort.* MFA authenticates the seller at the moment of posting but leaves no durable record afterwards, so it does not help investigate a later denial. Students who select B are applying the first-order habit the paper is designed to probe (“MFA solves it”); flag the B rate in the item analysis. D attracts students who equate a backup with an audit trail; a copy preserves the data, not the attribution.

A4. A developer deliberately inserts a hidden backdoor into the product’s own source code. Which supply-chain weakness best describes this? **(1 mark)**

- A. Compromised third-party library
- B. Third-party misconfiguration
- C. Insider threat in development**
- D. Compromised update channel

Marking guidance: Award 1 mark for **C, Insider threat in development**; 0 for any other option. An employee altered the product’s own source code.

Distractor note: A and D attract students who assume any supply-chain question must involve an external party. All four options are taught categories; only the insider one involves the organisation’s own developer and its own source.

A5. A threat model identifies a risk that a signed-in customer could read another customer’s records. Testing time is limited. Which action best uses the threat model to guide testing? **(1 mark)**

- A. Prioritise adversarial authorisation tests derived from that threat**
- B. Add input-validation tests for the listing and messaging forms
- C. Scan third-party dependencies before the next release
- D. Increase load testing of the customer-record pages

Marking guidance: Award 1 mark for **A**; 0 for any other option. The taught sequence runs threat model → security test plan, and testing time is limited.

Distractor note: B, C and D are all legitimate security activities, which is the point of the item: it tests whether the student lets the identified threat set the priority rather than reaching for any security activity. The threat is one customer reading another’s records, which is an authorisation question; none of B, C or D exercises authorisation. B attracts students who default to input validation as the universal answer; D confuses availability testing with the stated threat.

SECTION B – VERY SHORT ANSWER

(3 marks; suggested 2 minutes)

B1–B3: Write exactly one word in each answer box. Do not include working or justification.

B1. A phishing campaign targets a clinic that has no MFA. In the risk equation $\text{Risk} = \text{Threat} \times \text{_____}$, one term has been replaced by a blank. Write the missing term only (one word). **(1 mark)**

Marking guidance: Award 1 mark for **vulnerability**. Accept “vulnerabilities”. Award 0 otherwise; no half mark.

Accept: vulnerability; vulnerabilities.

Award 0: risk; likelihood; probability; impact; asset; exploit; control; “no MFA”; “lack of MFA”. The last two name an instance of the missing term rather than the term, and the stem asks for one word.

Borderline: “weakness” is the lecture gloss for vulnerability, not the term in the equation. Award 0, but tally these responses. If more than three scripts give it, stop and refer to the coordinator rather than deciding case by case.

B2. A team gives the Affected users category a score of 10 because it assumes every account could be affected. Later evidence shows that the exploit could reach only a small subset of users. Should the score be increased, reduced or left unchanged? Write one word only. **(1 mark)**

Marking guidance: Award 1 mark for **reduced**; 0 otherwise; no half mark.

Accept: reduced; reduce; reduces; lower; lowered; decrease; decreased.

Award 0: increased; unchanged; “keep at 10”; “it depends”.

Award 0: a bare number such as “5”. The 2026 stem asks the student to choose between three named responses and to write one word; a score is not one of them. Tally these and refer to the coordinator if more than three scripts give a number.

Note: the item tests whether the student scores the population the exploit *reaches*, not the population the system holds. The anchor for “some users, but not all” is 5, but no explanation or score is required or rewarded.

B3. A threat is scored 5 for Damage, 5 for Reproducibility, 5 for Exploitability, 5 for Affected users and 10 for Discoverability. Later evidence shows that it is very hard to discover and requires source code or administrator access. Using the reference table, what is the corrected overall risk band? Write the band name only (one word). **(1 mark)**

Marking guidance: Award 1 mark for **Medium**; 0 otherwise; no half mark.

Working (marker reference only): the new evidence matches the 0 anchor for Discoverability word for word, so $5 + 5 + 5 + 5 + 0 = 20 \rightarrow \text{Medium (11–24)}$.

Accept: Medium. Also accept “Medium” written alongside the total, for example “Medium (20)”; do not penalise the extra number.

Award 0: High, which is what the student gets by leaving Discoverability at 10 for a total of 30; Low; and a bare number such as “20”, since the stem asks for the band name.

Note: a student who reduces Discoverability to 5 rather than 0 still lands in Medium and scores 1. The band is what is marked.

DREAD scoring reference

Score each category from 0 to 10. The table gives the taught anchor descriptions.

Category	0	5	10
Damage potential	Nothing	Individual user data compromised	Complete system or data destruction
Reproducibility	Very hard or impossible, even for administrators/DBAs	One or two steps; may need an authorised user	A web browser alone is enough
Exploitability	Advanced programming and networking knowledge	Malware or suitable tools are available online	A web browser alone is enough
Affected users	None	Some users, but not all	All users
Discoverability	Very hard or impossible; needs source code or administrator access	Can be found by guessing or monitoring network traces	Visible in the browser, address bar, form or a hidden variable

Overall score bands: Critical 40–50; High 25–39; Medium 11–24; Low 0–10.

SECTION C – STRUCTURED SHORT ANSWER**(27 marks; suggested 22 minutes)****C1 – Community clinic booking system****(9 marks)****A community clinic uses an online appointment system:**

- Patients sign in with an email address and hashed password; the booking page uses plain HTTP.
- There is no limit on failed login attempts.
- A receptionist dashboard shows each patient's name, date of birth and reason for visit.
- Appointment data is held in one database; reminders are sent by SMS.

- (a) **(4 marks)** Identify two threats from different STRIDE categories. For each threat, name the category and identify one scenario fact that enables it.

Marking guidance: Mark each of two threats out of 2:

- **Category (1):** 1 for the correct STRIDE label; 0.5 if the label is absent or imprecise but the described security effect unambiguously matches a category; 0 for an incorrect category.
- **Enabling fact (1):** 1 where the student names a scenario fact that genuinely enables the threat; 0.5 where the fact is relevant to the scenario but is not what enables the named category, or where the connection is left implicit; 0 for a generic or invented fact.

The two threats must come from different categories. Where both fall in the same category, mark the stronger out of 2 and award 0 for the second.

2/2 — Information disclosure (HTTP). “The booking page runs over plain HTTP, so anyone on the same network can read the email address, password and reason for visit as they are submitted.”

2/2 — Information disclosure (SMS). “Reminders go out by SMS, so appointment details land on a device the clinic does not control; a shared handset, a lock-screen preview or a recycled number exposes them.”

2/2 — Tampering. “Because bookings travel over plain HTTP, an attacker on the network path can change the appointment time or the patient identifier before the request reaches the server, and neither side detects it.”

2/2 — Spoofing. “There is no limit on failed login attempts, so an attacker can run a password list against a known patient email until one works and then sign in as that patient.”

1.5/2. “Information disclosure — HTTP is not encrypted so data could be seen.” Category 1; fact 0.5, since neither the data nor the observer is identified.

1/2. “Someone could log in as a patient because the passwords are weak.” Category 0.5 (spoofing effect unambiguous, unlabelled); fact 0, since password strength is not a stated fact and the enabling fact (no login limit) is not used.

0/2. “The system is insecure and patient data could be leaked.” No category, no fact.

Adjudications.

- *Same fact, different categories.* Disclosure and tampering both argued from plain HTTP are acceptable: the categories differ and the effects genuinely differ. Two *disclosure* answers, one from HTTP and one from SMS, are a duplicate under the 2026 wording; mark the stronger and award 0 for the second.
- *Repudiation.* Award 0. The scenario says nothing about logging, so the threat is not case-supported.
- *Denial of service.* Award 0 for “one database is a single point of failure” with no mechanism. Award up to 2 where the student describes unlimited login attempts loading the authentication service to the point of unavailability, since that is a mechanism drawn from a stated fact.

- *Receptionist dashboard.* Do not award disclosure merely because an authorised receptionist sees patient information; that is the intended function. The scenario does not state that the dashboard lacks role restriction, so an answer asserting it scores 0 on the fact.
- *Elevation of privilege.* Not supported by any stated fact. Award 0.

(b) *The clinic adds this control: after five failed logins, the account locks and can be unlocked only by phoning reception.*

i. **(1 mark)** What immediate threat does this reduce?

Marking guidance: Award 1 for online password guessing or brute force, or for the resulting spoofing/account takeover. Award 0.5 for wording that identifies the mechanism but not the threat; 0 for a generic claim.

- 1. "Online password guessing against a known patient email, and so the spoofing/account-takeover threat from part (a)."
- 1. "Brute-forcing a patient account, because the attacker now gets five attempts instead of unlimited ones."
- 0.5. "It stops people trying lots of passwords." Mechanism right; no threat named.
- 0. "It makes the login more secure." / "It stops hackers."

Note: do not require both the mechanism and the STRIDE label. Either "brute force" or "spoofing" alone earns 1 when tied to the login path.

ii. **(1 mark)** Identify one new security risk or attacker adaptation introduced by the lockout and phone-unlock process.

Marking guidance: Award 1 for a specific new risk or adaptation; 0.5 for a relevant but vague downside with no attacker action and no operational consequence; 0 for a risk that existed before the control.

- 1. "An attacker locks a patient out deliberately by submitting five wrong passwords against their email address."
- 1. "The phone unlock becomes the target: an attacker rings reception posing as the patient and has the account unlocked or the password reset."
- 1. "Reception becomes a bottleneck; a burst of lockouts pulls staff off clinical work to handle unlock calls."
- 0.5. "Patients might get locked out and be annoyed." Relevant, but no attacker action and no operational consequence.
- 0. "Passwords could still be phished." Pre-existing; the control did not introduce it.

Note: the 2026 stem names *both* the lockout and the phone-unlock process, so social engineering of reception is squarely in scope and should not be treated as a tangent. This item is where first-order answers surface: "more secure but less convenient" scores 0, because a trade-off in the abstract is neither a risk nor an adaptation.

iii. **(2 marks)** For your answer in (ii), describe one concrete action of the attacker and one consequence or way the attack could be scaled up.

Marking guidance: Concrete action (1): 1 for a specific, case-grounded action; 0.5 for a plausible but underspecified one. **Consequence or scaling (1):** 1 for a clear causal consequence or scaling step; 0.5 if relevant but the causal link is only implied.

2/2 — lockout as the goal. “Patient email addresses are guessable or harvested from the clinic’s own correspondence. The attacker submits five wrong passwords per address, each account locks, and those patients can no longer book online. Scripted across a list, every lockout becomes a phone call, so reception is saturated and the unlock channel itself stops working.”

2/2 — lockout as the setup. “The attacker locks the target’s account on purpose, waits, then rings reception posing as the locked-out patient. Reception, dealing with a backlog, unlocks or resets the account without a strong identity check, so the lockout has opened a social-engineering path to takeover.”

1/2. “The attacker locks lots of accounts, which is bad for patients.” Action 0.5 (nothing described beyond “locks”); consequence 0.5 (no causal step).

0.5/2. “Denial of service.” Naming the threat is not an action; award 0.5 only where an action is implied.

Follow-through. The 2026 stem ties this item to (ii), so mark the action against the risk the student named there. Where (ii) scored 0 because the risk was pre-existing or unrelated, still mark (iii) on its own merits if the chain is coherent and case-grounded; do not deduct twice. Where (iii) develops a different risk from the one given in (ii), mark it on merit rather than for consistency.

Do not grade on ambition. The second exemplar is the more sophisticated answer; both score 2/2.

- iv. **(1 mark)** Name one additional control or monitoring measure that would reduce the risk described in (ii)–(iii).

Marking guidance: Award 1 for a concrete measure that reduces the student’s own (ii)–(iii) risk; 0.5 for a sound direction too vague to implement; 0 for a measure that addresses a different risk.

1 — against deliberate lockout. Per-IP or per-device rate limiting; progressive delay between attempts; CAPTCHA after a threshold of failures; throttling or locking the source rather than the account; time-limited lockouts that expire automatically; alerting on a spike in lockouts.

1 — against abuse of the phone-unlock path. A scripted identity check before reception unlocks; call-back to a number already on file; secure self-service recovery that takes reception out of the loop.

0.5. “Monitor the logs.” Direction sound; nothing to implement. “Raise the limit to ten attempts” earns 0.5 only where the student states the trade-off it makes.

0. HTTPS/TLS on the booking page; MFA at login; stronger password rules. Each is a reasonable security measure but none reduces the lockout or phone-unlock risk.

Change from the 2025 schedule: the 2026 stem reads “the risk described in (ii)–(iii)” and no longer offers “or another risk that remains”. Controls aimed at other scenario weaknesses, in particular HTTPS and MFA, now score 0. Markers who used the earlier schedule should note this before starting.

Where (ii)–(iii) scored 0 outright, judge (iv) against deliberate account lockout as the default risk.

C2 – Staff file-sharing service**(9 marks)****A file-sharing service stores customer documents for staff:**

- Any signed-in user can create a public share link; links never expire and link creation is not logged.
- Every staff account can open every customer folder, regardless of role.
- The administrator console can delete accounts and change permissions. It is public-internet facing and password-only.

- (a) **(4 marks)** Name two different secure-design principles from the course framework that this design violates. For each, identify the design detail that demonstrates the violation.

Marking guidance: Mark each of two principles out of 2: **principle (1)** – 1 for a correct principle named from the course framework, or 0.5 where the principle is accurately described but not named or is given an imprecise label; **design detail (1)** – 1 for a specific design detail that demonstrates the violation, or 0.5 where the detail is relevant but the demonstration is left implicit.

2/2 — P8 Least privilege. “Every staff account can open every customer folder regardless of role, so staff routinely hold access far beyond what any single operation requires.”

2/2 — P1 Minimise attack surface. “The administrator console, which can delete accounts and change permissions, is reachable from the public internet, so the highest-value path into the system is exposed to everyone.” Also accept: “Public share links never expire, so every link ever created stays a permanent path to the data and the set of paths in only grows.”

2/2 — P2 Defence in depth. “The exposed administrator console is password-only. There is a single layer on the most damaging function, so a credential compromise meets nothing behind it.”

2/2 — P7 Secure the weakest link. Award only where the student explains that an attacker spends the least effort available and so goes at the public password-only console rather than any stronger control, making that path the one that determines the system’s security.

1.5/2. “Least privilege — the permissions are too broad.” Principle 1; detail 0.5, since no design detail is cited.

1.5/2. “Everyone can open everything, which is more access than anyone needs for their job.” Principle 0.5 (accurately described, unnamed); detail 1.

0/2. “Confidentiality is violated” / “it breaks CIA” / “it is not zero trust”. None is a principle in the course framework, which the 2026 stem requires.

Do not accept.

- *P9 Fail securely* for links that never expire: nothing in the scenario fails.
- *P10 Psychological acceptability*: no usability-versus-security trade-off is described.
- *P6 Open design*: nothing depends on the secrecy of the design.
- *P11 Validate inputs*: no input handling is described.
- *P5 Separation of duty* for all-staff folder access. P5 concerns not granting a privilege on a single condition, and its taught examples are two-signature approval and two-factor authentication. The correct label here is least privilege. Award 0 for the principle; the detail mark cannot be earned either, since the link runs to the wrong principle.

Duplicates: the two principles must be different *and* rest on different design details. Least privilege and defence in depth both argued from the exposed console is one answer; mark the stronger and award 0 for the second.

- (b) **(3 marks)** Give one plausible persona non grata. Present your answer under these four headings: Actor and access; Motivation; Concrete misuse; and Design weakness enabling the misuse.

Marking guidance: The 2026 paper supplies a four-row table. Mark box by box: **Actor and access (0.5); Motivation (0.5); Concrete misuse (1); Design weakness enabling the misuse (1)**. For either 1-mark box award 0.5 where the idea is relevant but vague or only implied. An empty box scores 0.

Placement: content that clearly answers one component but sits in a neighbouring box still earns that component's mark. Do not penalise a student for filling the table untidily.

3/3 — insider. *Actor and access:* a support engineer holding an ordinary staff account (0.5). *Motivation:* money, by selling client documents to a competitor (0.5). *Concrete misuse:* signs in, opens customer folders unrelated to any ticket assigned to them, creates public share links for the documents and passes the links on (1). *Design weakness:* every staff account can open every customer folder regardless of role, and link creation is not logged, so nothing stops the access and nothing records it (1).

3/3 — external. *Actor and access:* an external attacker with no account (0.5). *Motivation:* resale or extortion (0.5). *Concrete misuse:* password-sprays common passwords across staff usernames at the administrator console until one works, then changes permissions and deletes accounts (1). *Design weakness:* the administrator console is public-internet facing and password-only (1).

2/3. Actor and motivation given; misuse “they look at files they should not” (0.5, not concrete); weakness cited and linked (1).

1.5/3. “A hacker” (0, no role or access level); “wants to steal data” (0.5); “breaks into the system” (0, no action); “it is not secure enough” (0, no stated design detail). Compare with “an attacker on the public internet with no credentials”, which earns the actor mark.

0/3. A misuse the scenario does not support: SQL injection, malware on a staff laptop, physical theft from a data centre.

Notes.

- Do not require misuse-case diagram notation or the *threatens/mitigates* relationship. This is the persona non grata route into misuse cases, not a UML exercise.
- The taught persona prompts are goals, skills and resources. A student who gives skills or resources instead of a role (“holds a valid staff login, needs no special tooling”) earns the actor mark.
- “Curious” or “disgruntled” is a sufficient motivation. “Malicious” alone is not: it restates that they are a misuser.
- The design weakness must be one of the three stated bullets. Award 0.5 where a correct weakness is named but not connected to the misuse described.

- (c) **(2 marks)** Recommend one control for misuse of the over-broad folder access. State whether its primary function is preventative, detective or responsive.

Marking guidance: Award **control (1):** 1 for a concrete control relevant to over-broad folder access; 0.5 for a sound but underspecified direction, or for a control aimed at a different weakness in the scenario. Award **classification (1):** 1 where the stated primary function is correct for the control described; 0 for a wrong classification.

2/2 — preventative. Role-based or per-customer folder authorisation; access scoped to the customers a staff member is assigned to; just-in-time grants tied to an open ticket and expiring afterwards.

2/2 — detective. Alerting when a staff account opens folders for customers outside its assignments; logging folder access with periodic review of cross-customer patterns.

2/2 — responsive. A defined revocation and containment procedure once misuse is found: suspend the account, invalidate its share links, notify the affected customers.

1/2. Correct control, wrong classification. The common instance is calling access logging or alerting *preventative*; a log records, it does not stop. Award the control mark and 0 for the classification.

1/2. “Restrict access” (0.5, no mechanism) classified preventative (0.5 at best, since there is little to classify). Forced expiry of share links is also 0.5, since it addresses the link weakness rather than the folder access named in the stem.

0/2. MFA on the administrator console: off-target for staff folder access.

Mixed controls: the 2026 stem asks for the *primary* function, which settles most borderline cases. “Alert and automatically suspend the account” may be classified detective or responsive; award 1 for either where the student’s choice fits the control as described. A log alone is detective, never enforcement of least privilege.

C3 – Mobile banking dependencies**(9 marks)****A mobile banking app imports 40 open-source libraries:**

- An image-processing library is pinned to a version released two years ago; that version has a public remote-code-execution CVE.
 - The other dependencies are unpinned; each build downloads their latest published versions.
 - The team has no software bill of materials (SBOM).
 - A released version of the app exposes a web address intended for developers to inspect internal server logs. Anyone can open it without signing in.
- (a) **(4 marks)** Identify two supply-chain weaknesses. For each, name the weakness and identify the dependency-management decision or related supply-chain practice that creates it. Not every security weakness in the scenario is supply-chain related.

Marking guidance: Mark each of two weaknesses out of 2: **weakness (1)** – 1 for an accurate name or description, 0.5 for an imprecise label that still identifies the correct mechanism; **decision or practice (1)** – 1 where the student names the dependency-management decision or supply-chain practice that creates it, 0.5 where a related fact is cited but the decision itself is not identified.

2/2 — known-vulnerable dependency shipped in the app. “The image-processing library carries a published remote-code-execution CVE. The decision that creates it is pinning to a version released two years ago and never revisiting the pin, so a fixed release exists upstream and the app does not take it.”

2/2 — uncontrolled ingestion of upstream releases. “Whatever was published most recently enters the build. The practice that creates it is leaving the other 39 dependencies unpinned, so a malicious or broken upstream release reaches the app with no review and no record of what changed.”

2/2 — no component inventory. “The team cannot say which components and versions are in a shipped build. The practice that creates it is not producing or maintaining an SBOM, so when a new CVE is published they cannot determine whether they are affected or how urgently to respond.”

1.5/2. “Old library with a CVE.” Weakness 1; decision 0.5, since the pinning decision is not identified.

1/2. “No SBOM means attackers can inject code.” Weakness 0.5: the absence of an SBOM is a detection and response weakness, not an attack mechanism. Practice 0.5.

0/2. “They use 40 libraries, which is too many.” Volume alone is not a weakness.

Adjudications.

- *The debug-log page scores 0 here.* It is the team’s own code, exposed by a release decision, and no third party is involved. The 2026 paper warns students that not every weakness in the scenario is supply-chain related, so apply this without leniency. The page is the correct subject of part (c), so do not treat its appearance here as evidence the student has misread the case.
- *Taught vocabulary.* The lecture list is compromised third-party libraries, malicious code injection, exploiting trusted update channels, insider threats in development, and misconfiguration of third-party libraries. Any of these, correctly applied, earns the weakness mark.
- *Untaught vocabulary.* Dependency confusion and typosquatting are not taught. Do not require them; accept them where used correctly, since both are plausible readings of the unpinned-dependency fact.
- *Duplicates.* The two weaknesses must rest on different decisions. “Old library” and “no patching process” argued from the same pin count once.

- (b) **(2 marks)** For each weakness from part (a), recommend one appropriate mitigation and state briefly how it reduces that weakness.

Marking guidance: For each of the two weaknesses: **0.5** for an appropriate mitigation and **0.5** for correctly explaining how it reduces that weakness. The 2026 stem asks for an *appropriate* mitigation rather than one from the course list, so accept any technically sound measure; the taught list is given below for consistency, not as a restriction.

1/1 — known-vulnerable library. “Upgrade to a fixed version and add an SCA or OSV scan to the build that fails on known CVEs (0.5). This removes the vulnerable code from the release and catches the same class of problem before the next one ships (0.5).”

1/1 — unpinned dependencies. “Pin versions in a lockfile and promote updates only after review and scanning (0.5). Builds become reproducible, and an upstream compromise can no longer reach the app without someone approving it (0.5).”

1/1 — no inventory. “Generate an SBOM for every build and check it against OSV or the NVD (0.5). It gives the inventory needed to answer ‘are we affected’ when a CVE lands, which is what cuts response time (0.5).”

0.5/1. “Update the library.” Appropriate mitigation (0.5); no explanation of how it reduces the weakness (0).

0.5/1. “Use a scanner.” Relevant direction (0.5) but nothing about what it detects or when it runs, so the explanation mark is not earned.

Trap: “Pin the image library” preserves the vulnerable version. Award 0.5 for the mitigation at most, and 0 for the explanation where the student claims pinning fixes the CVE. Pinning is the right answer for the *unpinned* weakness, not for this one.

Follow-through limit: if part (a) named the debug-log page, mark (b) only on the genuinely supply-chain weaknesses the student gave. Follow-through covers reasonable earlier answers, not out-of-scope ones.

Taught mitigation list (reference): vendor assessment and management; regular security audits; software composition analysis (SCA) tools; keeping dependencies up to date; enhanced code review; least privilege; SBOM generation checked against CVE, CVSS and OSV.

- (c) **(3 marks)** Choose which issue to address first: the vulnerable image library or the publicly accessible debug-log page. Compare the two issues in terms of (i) their probable security impact and (ii) how easily each could be exploited. Use scenario information and justify your choice of priority. Either issue may be placed first if the comparison supports the decision.

Marking guidance: Award **priority (1):** 1 for an explicit choice; 0.5 where the preference is clear but not stated as a decision. Award **impact comparison (1):** 1 where the probable security impact of *both* issues is characterised and the comparison is tied to the priority; 0.5 where only one issue is characterised or the comparison is not connected to the choice; 0 for an unsupported assertion. Award **ease of exploitation (1)** on the same basis. **Either issue may be placed first.**

DREAD categories are not required by the 2026 stem. Plain language earns full marks. Where a student uses Damage potential and Exploitability, mark the substance, not the labels, and do not require numerical scores.

Indicative content (marker reference).

- *Debug-log page.* Impact: exposure of internal server logs, which in a banking app may carry account identifiers, session tokens or transaction detail; bounded by what the logs actually

contain, so a student should state an assumption. Ease: very high, a web address anyone can open without signing in, needing only a browser.

- *Image library CVE*. Impact: remote code execution, so potentially control of the host rather than exposure of data, which is a larger class of harm. Ease: lower, since the attacker must identify the vulnerable version, obtain or write an exploit and reach the vulnerable code path, though a public CVE means exploit code may already exist.

3/3 — page first. “Fix the debug-log page first. On ease of exploitation the two are not close: the page needs a browser and no credentials, while the CVE needs the version identified and an exploit that reaches the vulnerable code. On impact, RCE is worse in principle, but assuming the logs carry account identifiers or session data, the page is leaking now to anyone who finds it, and a certain smaller harm outranks a harder-to-reach larger one.”

3/3 — library first. “Fix the library first. On impact, remote code execution can mean control of the server and all the data it holds, against exposure of whatever happens to be written to the logs. On ease, the page is easier to reach, but the CVE is public, so exploit code is likely available and the gap is narrower than it looks. The larger impact decides it.”

2/3. Clear choice (1); both issues compared on ease (1); impact asserted for one issue only (0.5), rounded to component values.

1/3. “The debug page is worse because it is more critical and easier.” Priority 1; neither dimension developed for both issues (0 and 0).

0.5/3. “The library is higher risk” with no comparison; priority only implied.

Notes.

- Accept reasonable stated assumptions about what the logs contain, where the library runs, or how exposed the app is. Do not accept an unstated assumption that the app has a huge user base, for either issue.
- An answer where the two dimensions point in opposite directions and the student explains the trade-off is the strongest form of this answer. Reward it; do not require the dimensions to agree.
- This item is independent; follow-through from (a)–(b) has no effect, since both candidate issues are named in the stem.

SECTION D – INTEGRATED CASE STUDY

(15 marks; suggested 15 minutes)

D1 – Student marketplace

(15 marks)

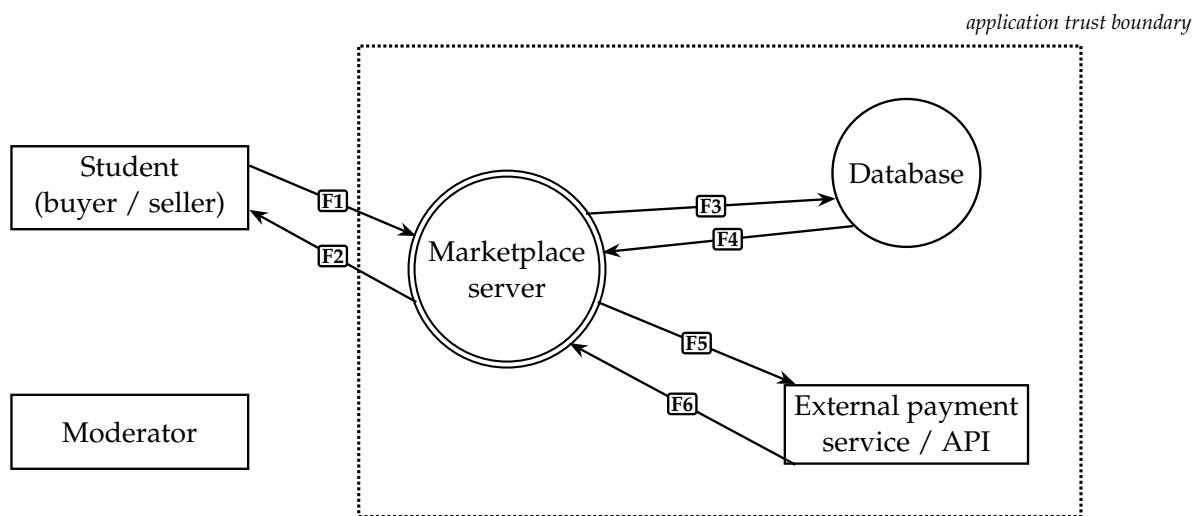
A university student marketplace is available on the web and through a mobile app. A typical student journey is:

- A student signs in. Password reset uses security questions such as “What is your pet’s name?”
- A buyer can browse listings and message a seller.
- After messaging a seller, the buyer can rate that seller even if no purchase occurred.
- A purchase is processed through an external payment service/ API using an SDK that has not been updated for more than a year.

Other design details are:

- Policy permits moderators to read only messages reported for abuse, but the system allows any moderator to read any message and does not log that access.
- After login, the server trusts an is_admin value supplied by the client when deciding whether to allow administrator actions.
- A central database stores user details, messages, ratings and transaction records.

Figure D1. Use the DFD notation in the legend and consider only the five components shown. The diagram contains three modelling errors.



Flow key: F1 login/listing/message data; F2 pages/results; F3 users/messages/ratings; F4 stored records; F5 payment request; F6 payment result.

Legend: rectangle = interactor; circle = process; double circle = multi-process; two parallel lines = data store; one-way arrow = data flow; dotted line = trust boundary.

Figure D1 – Proposed data-flow diagram for the student marketplace

- (a) **(3 marks)** Complete the table by identifying each of the three errors in Figure D1 and explaining how it could cause a threat to be missed.

Marking guidance: The 2026 paper supplies a three-row table with columns *Modelling error* and *How it could cause a threat to be missed*. For each row award **0.5** for accurately identifying the error and **0.5** for a plausible missed-threat consequence. An empty row scores 0. Do not require the model wording, and do not penalise a student who writes the error and its consequence across both columns.

1/1 — Error 1: the external payment service/API is drawn inside the application trust boundary. A trust boundary marks where the level of trust changes, and crossing one is where validation, authentication and authorisation belong. Placing the third party inside says the application trusts payment responses as though it produced them itself. The F5/F6 flows are then never questioned, so a spoofed or tampered payment result, disclosure of order data to an outside party, and the risk carried by the year-old SDK all drop out of the analysis.

1/1 — Error 2: the database is drawn as a process (circle) rather than a data store. Element-specific STRIDE applies a different question set to each element type, and the data-store row exists to force the data-at-rest questions. Treating the store as a process means those questions are never asked of it: tampering with stored ratings or transaction records, disclosure of the central message store, and repudiation where access to stored records is not recorded. It also breaks the rule that a data store must have at least one input and one output flow, which is how the modeller checks the store has been represented at all.

1/1 — Error 3: the Moderator interactor has no data flow. An element with no flows is never analysed, and every element must connect to at least one other. The moderator is the actor in the one policy conflict the case states: any moderator can read any message although policy allows only reported ones, and the access is not logged. With no flow there is nothing on which to ask whether authorisation is checked, so the information-disclosure and repudiation threats on that path are omitted entirely.

0.5/1. “The payment service should be outside the boundary.” Error identified; no consequence.

0.5/1. “The database is the wrong shape.” Error identified; no consequence. Accept “should be two parallel lines” as identification.

0.5/1. “The moderator is not connected to anything.” Identification 0.5; no missed-threat consequence.

Do not accept as errors (the figure caption rules these out): omitted SMS, email or notification services; unlabelled flows, since all six are labelled; “arrows should be two-way”; “the moderator should be inside the trust boundary”, since a moderator is a human user and is correctly modelled as an external interactor.

Do not double count. “The database is a circle” and “there is no data store in the diagram” are the same error. Credit one row and award 0 for the other.

Ceilings and extras. A student who finds two errors and explains both fully scores 2/3. Where a student writes more than three items, mark the three best rows; do not deduct for the extras.

- (b) **(6 marks)** Identify three threats, each from a different STRIDE category. For each, name the category and explain how a specific scenario detail enables the threat.

Marking guidance: Mark each of three threats out of 2: **category (1)** – 1 for the correct STRIDE label, or 0.5 where the label is absent or imprecise but the security effect is unambiguous; **scenario detail and explanation (1)** – 1 where a specific case detail is named *and* the student explains how it enables the threat, or 0.5 where the detail is present but the enabling step is left implicit.

Different categories are a stated requirement. Where two threats fall in the same category, mark the stronger out of 2 and award 0 for the duplicate. Do not apply an overall cap and do not split

marks between them. Judge the category by the effect described where the label is missing or wrong.

2/2 — Spoofing. “Password reset relies on security questions such as the pet’s name. On a student marketplace that answer is usually on a public social profile, so an attacker resets the victim’s password and signs in as them.”

2/2 — Tampering. “A buyer can rate a seller after messaging them even if no purchase occurred, so a rival opens a message thread and posts a run of one-star ratings. The stored reputation data no longer reflects real transactions.”

2/2 — Information disclosure. “Policy allows moderators to read only reported messages, but the system lets any moderator read any message, so a moderator can read private negotiations between students who have never been reported.”

2/2 — Repudiation. “Moderator message access is not logged, so a moderator who reads messages outside policy can deny having done so and the platform has no record to show otherwise.”

2/2 — Elevation of privilege. “The server trusts an `is_admin` value supplied by the client, so an ordinary user can set it to true in the request and carry out administrator actions.”

1.5/2. “Information disclosure — moderators can see too much.” Category 1; detail 0.5, since the policy-versus-system gap is not identified.

1/2. “A normal user could become an admin.” Category 0.5 (effect unambiguous, unlabelled); detail 0.5 where `is_admin` is not named.

0/2. “Denial of service — there is one central database.” No mechanism.

Adjudications.

- *Moderator facts.* Reading messages outside policy is disclosure; the absence of a record of that reading is repudiation. Both are available, and both may be credited where the student separates them. Stated as one undifferentiated point, they count once.
- *Label flexibility on `is_admin`.* A student who calls the client-supplied flag “tampering” has a defensible reading of the flow. Award the category mark where the described effect is administrator access being obtained. Do not then credit the same fact again as elevation of privilege; one fact, one threat.
- *The stale SDK.* “The SDK is old so there might be a vulnerability” is not yet a STRIDE threat. Award up to 2 where the student names a category and a security effect, for example tampering with payment data through a known SDK flaw.
- *Unsupported claims.* Do not infer that a shared login or a normal server-side role flag is insecure. Do not equate the absence of end-to-end encryption with the absence of transport or storage encryption. Do not credit “there is no MFA” or “traffic is unencrypted”; neither is stated.

- (c) **(4 marks)** Choose two threats from part (b) and rank them first and second. Compare them using the DREAD categories Damage potential and Exploitability, then explain how those comparisons support your ranking.

Marking guidance: Four components, 1 mark each.

- **Ranking (1):** a clear first and second; 0.5 where the preferred threat is identifiable but second place is only implied.
- **Damage potential (1):** the category applied comparatively to *both* threats; 0.5 where it is applied correctly to one threat only.
- **Exploitability (1):** on the same basis.
- **Link to the ranking (1):** the student explains how the two comparisons produce the order chosen; 0.5 where the link is asserted but not explained; 0 where the ranking and the comparisons sit side by side with nothing joining them.

The 2026 stem fixes the two categories and states that complete numerical scores are not needed. Do not require numbers, and do not credit a substitute category in place of Damage potential or Exploitability.

4/4. “First the `is_admin` elevation of privilege, second the moderator message disclosure. *Damage potential:* the admin flag reaches account deletion and permission changes across the platform, which is the complete-destruction end of the scale, against individual users’ message content for the disclosure threat. *Exploitability:* the flag is a client-supplied value that can be changed in the request with a browser and no special knowledge, while the disclosure threat needs a moderator account first. *Link:* both comparisons favour the same threat, and the one that is simultaneously more damaging and easier to reach has to be fixed first.”

4/4 — comparisons in tension. “First the security-question account takeover, second the rating manipulation. *Damage potential:* takeover compromises a whole individual account, including messages and transaction history, against distorted reputation data for one seller. *Exploitability:* both are easy, but rating manipulation only needs a message thread, so it is marginally easier. *Link:* exploitability slightly favours the ratings threat and damage strongly favours takeover, so the larger harm decides the order.”

3/4. Ranking (1), both categories compared across both threats (1 + 1), no explanation of how the comparisons produce the order (0).

2/4. Ranking (1); Damage potential compared across both (1); Exploitability asserted for one threat only (0.5) with no link (0), rounded to component values.

1/4. “Elevation of privilege is worst, then disclosure. It is more damaging and easier to exploit.” Ranking 1; neither category developed for both threats; no link.

Notes.

- An answer whose two comparisons point in opposite directions and which explains the trade-off is the strongest form. Reward it; the categories are not required to agree.
- Follow-through from (b): mark the comparison on the student’s own two threats, including a weak but reasonable one. Do not re-penalise a category error already deducted in (b).
- Do not require the student to say “10” or “5”. Anchor language, or plain language that matches the anchors, earns the mark.

- (d) **(2 marks)** The team is considering two separate changes: A – an SMS one-time code at login; B – a one-time password-reset link sent to the user’s university email. For each change, identify one new security risk or dependency it introduces.

Marking guidance: Award **1 for A** and **1 for B**. For each: 1 for a specific new risk or dependency that the change introduces; 0.5 for a relevant but generic concern that does not identify what is new; 0 for a risk that existed before the change.

1 — A, dependency. “Login now depends on the student holding a working number on a mobile network the university does not control; if delivery fails or the number changes, nobody can sign in.”

1 — A, new risk. “The phone number becomes an authentication asset, so SIM swap is now worth attempting: convince the carrier to move the number and the codes follow.” Also accept SMS interception, codes readable from a lock-screen preview, number recycling, real-time relay phishing where the attacker replays the code within its validity window, and brute-forcing a six-digit code where attempt limits are unspecified.

1 — B, dependency. “The university email account becomes a dependency of the marketplace account: whoever controls the mailbox controls the marketplace login, so a phished university sign-in now yields both.”

1 — B, new risk. “Mailboxes are tied to enrolment, so graduates lose access or addresses get reassigned and the reset path points at an address the user no longer controls.” Also accept a forwarding rule planted in the mailbox, and the attacker shifting target from the marketplace to the university mail system.

0.5. “SMS is not secure.” / “Email accounts can be hacked.” Relevant, but nothing new is identified.

0.5. “Some students might not have a phone.” Award 1 where the student frames it as the design now depending on a device and network outside the university’s control.

0. “The password could still be guessed” (pre-existing). “The link could be reused” (the stem specifies one-time).

Note: this is the clearest test of second-order thinking in the paper. “It is more secure” scores 0 even where the security claim is true, because no new risk or dependency is named. Mark A and B independently; a strong A does not carry a weak B.

Marker quick reference

Element-specific STRIDE (taught table). Used mainly for D1(a) error 2 and D1(b).

	S	T	R	I	D	E
Data flow		X		X	X	
Data store		X	X	X	X	
Process	X	X	X	X	X	X
Multi-process	X	X	X	X	X	X
Interactor	X					

Secure-design principles (course framework). Used for C2(a). P1 minimise attack surface; P2 defence in depth; P3 do not reinvent the wheel; P4 economy of mechanism; P5 separation of duty; P6 open design; P7 secure the weakest link; P8 least privilege; P9 fail securely; P10 psychological acceptability; P11 validate inputs; P12 secure data at rest; P13 avoid bypass attacks.

Supply-chain weakness categories (taught list). Used for C3(a). Compromised third-party libraries; malicious code injection; exploiting trusted update channels; insider threats in development; misconfiguration of third-party libraries.

Items where a duplicate scores zero. C1(a) two different STRIDE categories; C2(a) two different principles on different design details; D1(b) three different STRIDE categories.

Items where follow-through applies. C1(b)(iii)–(iv) from (ii); C3(b) from (a), for supply-chain weaknesses only; D1(c) from (b). C3(c) and D1(d) are independent.

Per-script tally and verification

A /5	B /3	C1 /9	C2 /9	C3 /9	D1 /15	Total /50

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

* * * * *