Secure code review evidence (SAMPLE)

Fernbrook Systems

Period	Q1 2025, 1 January 2025 to 31 March 2025
Produced	31 August 2026 by the sample report
Export reference	sample-py-q1
Named detail	Open: named per-person detail is included for the roles entitled to it.
Plan	Team

NOT EVIDENCE. This is a sample document produced from invented data so that a buyer can read one before paying. Every name, score and date in it was made up, and it is not recorded against any organisation.

What this is

Each exercise is a staged merge request against a working codebase: a branch, its commits and the diff a reviewer would actually open, written to look like ordinary feature work.

Findings are planted in that diff by hand, each with a CWE identifier, the exact lines, and a written explanation.

Grading runs on the server against a prepared answer that is never sent to the browser. The reviewer submits one finding at a time, the lines they flagged and the weakness they think it is, and each is marked as it is submitted. The score is worked out on the server, so a submission cannot be forged.

A weakness is found when the reviewer's selection overlaps the lines it was planted on and the weakness they name is the answer's own, an alternative the answer accepts or one in the same family. Each weakness is worth 10 points, so a challenge with three weaknesses planted in it is out of 30.

Every wrong try costs 1 point: a try on lines with nothing wrong in them, or on the right lines naming the wrong weakness. The score is the points for the weaknesses found less what the wrong tries cost, and never below nought. A weakness never found scores nothing. In an assessment, a weakness closes after 3 tries on its lines that name the wrong weakness, and scores nothing from then on.

A challenge is passed at 80 per cent of its full marks, rounded up to whole points: 24 of 30 with three weaknesses. The score recorded for it is the percentage of its full marks it reached.

Assessment work is kept separate from training in the database. Everything in this report comes from assessments an administrator set. Practice a seat holder does on their own account never reaches it.

The grading policy is published in full and does not change per customer. It is at justappsec.com/docs/how-a-review-works.

What this record does not establish

What this record establishes: that the account named against each result was signed in, submitted those findings in the window shown, and was marked on our server against a prepared answer the browser never receives. The exercise is never published, so it cannot be practised beforehand. Hints are off unless an administrator switched them on, which is stated on each row. No assessment shows a walkthrough or a video, whatever its switches say.

What it does not establish: who was at the keyboard. The sitting is not invigilated. Nothing here shows whether the named person worked alone, whether someone was helping, or whether the diff was pasted into another tool. We only see what the product offered and what was submitted to it.

SAMPLE

Read a result as evidence of work done under this account, in this window, under the stated settings. Where a role needs more than that, run the sitting under your own supervision and record that alongside this document.

Who is in this document: everybody who was assigned an assessment in the period and held a seat when they sat it. That is not necessarily everybody who writes code here. This record cannot say how many engineers the organisation has or how the people assigned were chosen. An auditor who needs that should ask for it.

Summary

Assessments	People assessed	Assignments	Submitted
0	**0**	**0**	**0**
In progress	Not started	Not completed in window	Reviews recorded
0	**0**	**0**	**0**

Team catch rate	Team median score
Not published	**Not published**

An aggregate is only published when at least 10 people are in it, and a median when at least 20 are, because with fewer than that you could work out who is who.

SAMPLE

Coverage by CWE category

No assessed work was submitted in this period.

Coverage by language

No assessed work was submitted in this period.

Track records

No assessed submissions were recorded in this period, so there is nothing per person to report.

No retake was granted or sat in this period. Every figure above is a first sitting.

Assessment records

No assessment was set or sat in this period.

Seat integrity

Seats purchased today	Assigned now	Held a seat in period	Assignments and releases
50	**48**	**44**	**1 and 0**

A seat is assigned to a named individual and can move to someone else once a calendar month. 50 seats are paid for as things stand today, and 44 different people held one during this period, counted from the seat assignments and releases in the audit log. A number well above the number purchased means seats were passed between people, which changes what the results above cover.

Where this fits

This is one layer of an evidence pack, not the whole of one. It shows that named people were measured against planted weaknesses, on dated exercises, under stated settings, and how they did. It does not cover your policies, your pipeline controls, your penetration testing or your vulnerability management, and it is not a certification.

PCI DSS v4.0.1, 6.2.2 and 6.2.3.1
Software development personnel are to be trained at least annually in secure design and coding for the languages and tools they use, and bespoke code is to be reviewed before release by someone other than its author, knowledgeable in secure coding. The graded records here double as the training record 6.2.2 asks for, and evidence that reviewers can do what 6.2.3.1 requires of them.

DORA (EU) 2022/2554, Article 13(6), and the ICT risk management RTS, Article 16(3)
Financial entities are to run ICT security awareness programmes and digital operational resilience training for staff, and to apply secure coding practices with a proportionate review of the code they produce. This report evidences the second half of that: that the people writing the code were measured against real weaknesses.

ISO/IEC 27001:2022, clause 7.2 and Annex A 8.28
Clause 7.2 asks for competence to be determined, achieved and evidenced, and A.8.28 asks for secure coding principles to be applied. A dated, per-person record of what a seat holder caught and missed is the kind of evidence of competence clause 7.2 asks for.

EU AI Act (EU) 2024/1689, Article 4
Providers and deployers are to take measures to ensure a sufficient level of AI literacy among staff dealing with AI systems. Where the exercises cover AI-assisted coding and AI-specific weaknesses, this record speaks to that obligation.

SAMPLE

Integrity of this file

NOT EVIDENCE. This is a sample document produced from invented data so that a buyer can read one before paying. Every name, score and date in it was made up, and it is not recorded against any organisation.

The underlying rows are exported alongside this document as a CSV, so the figures above can be recomputed.