Forwards or Backwards

https://forwardorbackwards.com



CISSP Domain 6: Security Assessment and Testing

Watch on YouTube: https://www.youtube.com/watch?v=8PldAhXZL34

A full tour of Domain 6 of the CISSP exam - Security Assessment and Testing - worth 12% of the exam under ISC2's exam outline effective 15 April 2024.

This video works through every sub-domain of Domain 6 in order: designing an assessment, test and audit strategy by choosing between internal, external and third-party reviews (6.1); the testing techniques themselves (6.2) - vulnerability assessment, the CVE Program (since September 1999) and CVSS severity scoring (v3.1, June 2019; v4.0, November 2023), penetration testing with red, blue and purple teams and black/grey/white box scope, log reviews, synthetic transactions and benchmark comparisons, code review and misuse-case testing, test coverage, interface testing, breach and attack simulation, and compliance checks; collecting the ongoing evidence a security

programme produces (6.3) - account management reviews, KPIs and KRIs, backup verification and DR/BC exercises, and training and phishing-simulation data; analysing test output into a risk-prioritised report, formal time-limited exceptions, and responsible vulnerability disclosure (6.4); and internal, external and third-party audits, and the SOC 1/2/3 and Type I/II report family read against the AICPA's Trust Services Criteria (6.5).

It closes with a worked scenario following one logistics company's year through the whole assessment-test-audit cycle, how Domain 6 feeds the other seven CISSP domains, a full recap, and sixteen exam-style questions with explained answers.

Every figure is on screen with its source and date.

Educational documentary. Not financial or investment advice.

Chapters

1. Welcome and learning objectives
2. Domain 6 in the CBK
3. Designing an assessment, test and audit strategy
4. Vulnerability assessment, CVE and CVSS
5. Penetration testing and team roles
6. Log reviews, synthetic transactions and benchmarks
7. Code review, misuse-case testing and test coverage
8. Interface testing, BAS and compliance checks
9. Security process data: accounts and management review
10. KPIs, KRIs and measuring the security programme
11. Backup verification, DR and business continuity data
12. Training and awareness data
13. Analysing test output and reporting
14. Remediation and exception handling
15. Ethical disclosure
16. Internal audits
17. External and third-party audits
18. SOC reports and audit standards
19. Putting the cycle together: a worked scenario
20. How Domain 6 links to the other seven domains
21. Domain 6 recap
22. Review: exam-style questions, part one
23. Review: exam-style questions, part two
24. Review: exam-style questions, part three

1. Welcome and learning objectives



Welcome to this training module on Domain 6 of the CISSP exam: Security Assessment and Testing.

By the end of this module you will be able to explain how an organisation designs an assessment and audit strategy, what the main testing techniques actually do, and how test results turn into reports, fixes and formal audits. You will also be able to work through the kind of scenario question the exam asks about each of those steps.

Under ISC2's exam outline, Domain 6 is worth twelve per cent of the exam. That outline took effect on 15 April 2024. The exam itself runs as a Computerized Adaptive Testing exam, between a hundred and twenty-five and a hundred and seventy-five questions long. It is taken over four hours. The pass mark is seven hundred out of a thousand.

Twelve per cent is a solid single-figure share of the exam, and this domain earns it by covering the part of security work that proves every other domain's claims are actually true. A policy that nobody tests is a guess dressed up as a control.

This module follows Domain 6's five sub-domains in order. First, designing an assessment, test and audit strategy. Second, the control-testing techniques themselves: vulnerability scanning, penetration testing, log review, code review, interface testing and more. Third, collecting the ongoing data a security programme produces. Fourth, turning test output into a report and acting on it. Fifth, running or supporting internal, external and third-party audits.

Each chapter defines its own terms, works through at least one worked example, and closes with key points and an exam tip: how ISC2 tends to test the idea, and the trap candidates fall into. Near the end, a worked scenario ties every sub-domain together, then a full recap, then exam-style review questions with explained answers.

Let's begin.

2. Domain 6 in the CBK



Domain 6 has not always existed in its current shape. For most of the CISSP's history, the material it now holds was scattered across other parts of the syllabus.

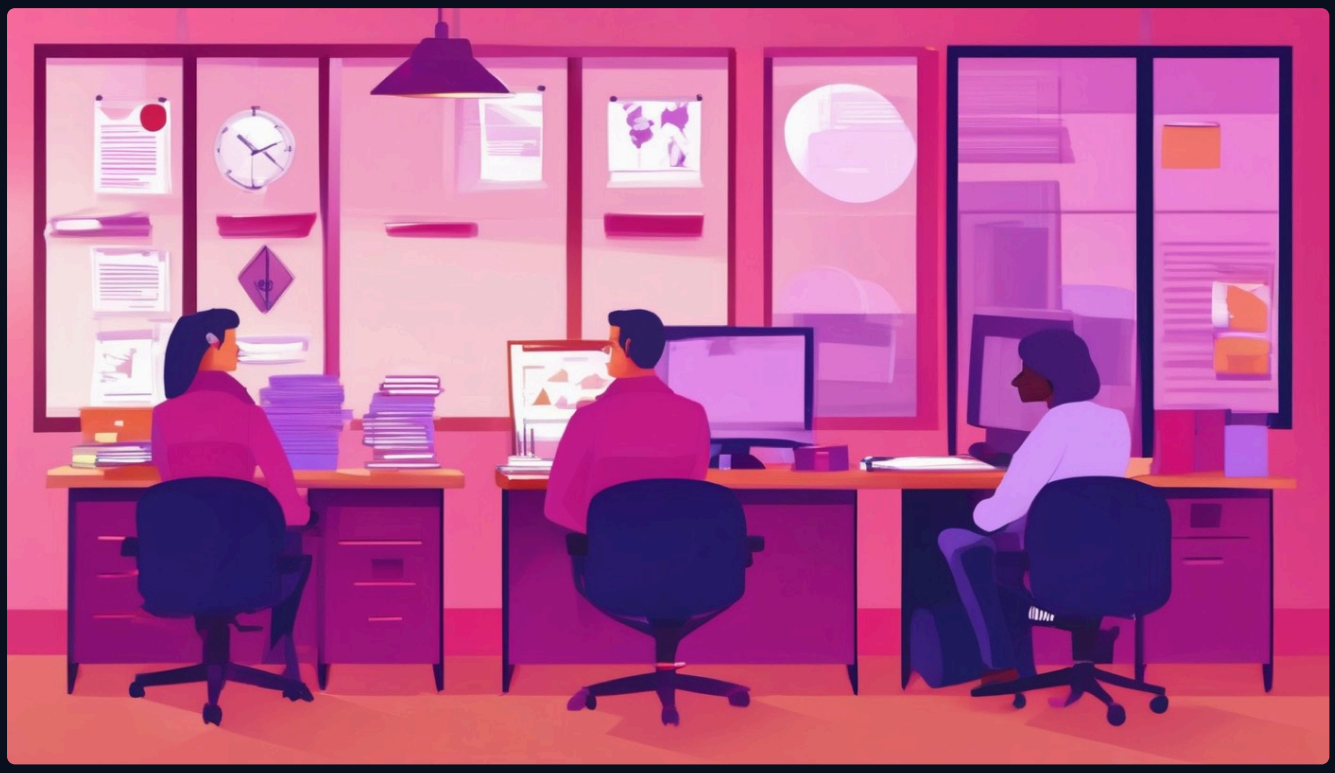
On 15 April 2015, ISC2 restructured the Common Body of Knowledge from ten domains down to eight. Testing and audit material that used to sit inside domains such as Operations Security, and Information Security Governance and Risk Management, was gathered into one new domain: Security Assessment and Testing. The eight-domain structure from that restructure is still the one the exam uses today, refreshed most recently by the outline effective 15 April 2024.

Gathering that material into one place was not just tidying. **An assessment, a test and an audit are three different activities that practitioners had always talked about together**, because each one checks whether a control that was designed somewhere else actually works. Putting them in one domain matched how the work is actually organised in most security teams.

Domain 6 sits between the domain that builds controls and the domain that watches them run day to day. Security architecture and engineering designs a control. Domain 6 proves whether that control, once built, actually does what it was designed to do. Security operations then watches it continuously, after Domain 6's one-time or periodic checks are over.

Exam tip: ISC2 sometimes tests whether a candidate can place a named activity in the right domain. A question might describe scanning, testing, auditing or reviewing logs for evidence. Whenever a question describes checking that a control actually works, the answer is almost always Domain 6, even if the control itself belongs to a different domain's design.

3. Designing an assessment, test and audit strategy



Before any scan runs or any auditor walks in, someone has to decide what gets checked, how often, and by whom. That planning step is sub-domain 6.1.

There are three broad types of review to choose between. An **internal** assessment is run by the organisation's own staff, using its own tools and schedule. An **external** assessment is run by an outside party the organisation hires directly, often for independence or specialist skill the organisation lacks in-house. A **third-party** assessment is run by, or for, someone else entirely, such as a regulator, a customer or an insurer. That outside party needs assurance about the organisation, rather than assurance the organisation sought for itself.

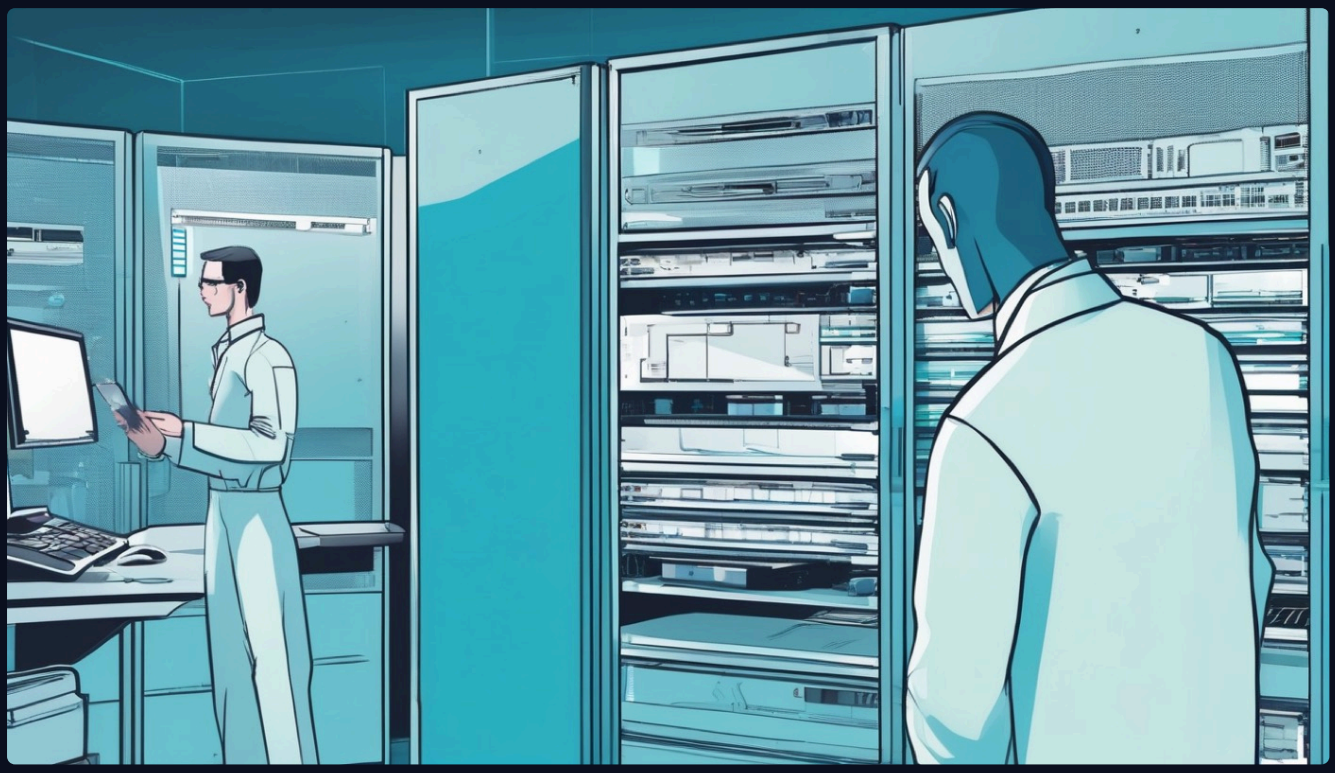
Here is a worked example. A security manager at a mid-sized payments company learns that a new regulatory requirement demands proof that customer data is protected, delivered to the regulator by an independent party rather than the company's own staff. Running the check internally would be cheaper and faster, but it would not satisfy the requirement, because the whole point is a view nobody inside the company could be accused of shaping.

The manager chooses an external assessment for the regulator's deliverable, while still running internal vulnerability scans every month to catch problems between the external reviews. That mix — internal activity for day-to-day assurance, external activity where independence is required — is the normal shape of a mature assessment programme, not a one-off choice.

Exam tip: when a scenario asks which type of assessment fits a situation, look for the word independence. A requirement for an outside, unbiased view points to external or third-party; a

requirement for frequent, low-cost, repeatable checking points to internal.

4. Vulnerability assessment, CVE and CVSS



Sub-domain 6.2 covers the actual techniques used to test controls, starting with the most common one: the vulnerability scan.

A **vulnerability assessment** is an automated scan of systems and networks for known weaknesses, compared against a list of problems other researchers have already found and catalogued. It does not try to break in; it only checks for flaws that are already documented. Those flaws are catalogued under the **CVE Program** – Common Vulnerabilities and Exposures – which has assigned a standard identifier to each publicly known flaw continuously since September 1999.

That continuity nearly broke. In April 2025, the US government contract funding MITRE to run the CVE Program was due to lapse on the sixteenth of that month, which would have stopped new identifiers being issued with no replacement system ready. CISA stepped in at the last moment and extended the contract by eleven months, keeping the catalogue running without a gap.

Once a flaw has a CVE identifier, it still needs a severity score, and that is what **CVSS**, the Common Vulnerability Scoring System, provides. CVSS scores a flaw from zero to ten using base, temporal and environmental metric groups, and the result sorts flaws into bands: critical, high, medium and low. Version 3.1 was published in June 2019 and is still widely used. Version 4.0 was published in November 2023, and is the current standard and the one this module uses when a score is mentioned.

Here is how those bands line up on the scale. A score from nine up to ten is critical, from seven up to nine is high, from four up to seven is medium, and from zero point one up to four is low. A zero score

means no vulnerability was found at all. Reading a real scan report means reading both halves: the CVE identifier says what the flaw is, and the CVSS score says how urgently it needs fixing.

Exam tip: ISC2 tests the difference between CVE and CVSS directly. CVE identifies; CVSS scores. A question describing a unique identifier for a publicly known flaw is pointing at CVE. A question describing a numeric severity rating from zero to ten is pointing at CVSS.

5. Penetration testing and team roles



A vulnerability scan tells you a door might be unlocked. **Penetration testing** goes further: a tester actively tries to open it, and see what is on the other side, under a controlled, agreed scope.

Pentests are organised around team colours. A **red team** attacks, playing the role of a real adversary. A **blue team** defends, trying to detect and stop the red team's actions. A **purple team** is not a third group so much as a way of running the exercise: red and blue share findings with each other in real time, so the organisation learns faster than it would from a report delivered only at the end.

Before any of that starts, the scope has to be nailed down in a document called the **rules of engagement**. It states exactly which systems may be touched, which techniques are off-limits, who to call if something breaks, and the start and end times of the exercise. Skipping this step is how a pentest turns into an unplanned outage.

The amount of inside knowledge a tester is given also has a name. **Black box** testing gives the tester no inside information at all, matching an outside attacker with nothing but public information. **White box** testing gives the tester full access to source code, architecture diagrams and credentials, matching an insider or a very advanced attacker. **Grey box** testing sits in between, with partial knowledge, which is where most real-world pentests actually land.

Here is a worked example. A retailer commissions a grey-box pentest of its online checkout system, giving the testers a standard customer account but no source code. The rules of engagement explicitly exclude the payment-processing partner's own servers, which belong to someone else entirely and are out of scope for this exercise.

Exam tip: ISC2 frequently tests what the rules of engagement actually are for. The correct answer is almost always that they define scope and limits before testing starts; a scenario where testers discover an unexpected system mid-test and keep going without updating the agreement is a process failure, not good initiative.

6. Log reviews, synthetic transactions and benchmarks



Scans and pentests are periodic snapshots. Three other techniques check controls continuously, between those snapshots, using evidence a system is already producing or can be made to produce on demand.

A **log review** is the systematic reading of records a system already keeps — logins, file access, configuration changes — looking for evidence a control is or is not working. A **synthetic transaction** is a scripted, fake transaction run deliberately through a live system, purely to check that the system still behaves correctly, the way a bank might run a test payment every hour to confirm its payment pipeline is still processing normally.

A **benchmark** comparison checks a system's actual configuration against a published standard for how it should be configured, such as the widely used CIS Benchmarks for operating systems and applications. Where a scan asks whether there is a known flaw here, a benchmark asks whether this system is set up the way good practice says it should be — which can catch a weak setting long before anyone publishes a flaw against it.

Here is a worked example. An online retailer runs a synthetic transaction every fifteen minutes that attempts a small test purchase through its own checkout, confirming the payment pipeline is actually working, not merely that the server is running and answering pings. One morning the synthetic transaction starts failing while every other monitoring dashboard stays green, because the server is up but a backend queue has silently stopped processing orders.

That gap between a server that answers and a business function that actually works is exactly why synthetic transactions exist: an uptime check watches a server, but the business watches whether a customer can actually complete an order.

Exam tip: ISC2 likes to test the difference between a technique that checks configuration against a standard and one that checks a live process end to end. If the scenario mentions a published standard, that is a benchmark; if it mentions running a fake transaction through production, that is synthetic testing.

7. Code review, misuse-case testing and test coverage



Some flaws are never found by scanning a running system, because they live in source code that has not shipped yet, or in logic a scanner has no way to exercise.

Code review examines source code before or alongside testing. A **static** code review uses automated tools to scan the code itself without running it, catching patterns known to be dangerous. A **manual** code review has a person read the code, which is slower but catches logic errors a pattern-matching tool cannot see, such as a check that looks correct but runs in the wrong order.

Testing a system's intended functions is only half the job. **Misuse-case testing** deliberately designs test cases around how a system could be misused, rather than how it is meant to be used. What happens if a quantity field is given a negative number, or if a password-reset form is called thousands of times a second? Where an ordinary use case asks what a legitimate user should be able to do, a misuse case asks what a hostile or careless one might try instead.

Test coverage measures how much of an application's actual logic was exercised by a given set of tests, usually as a percentage of code paths executed at least once. A high coverage percentage is reassuring, but it is not proof of correctness: a test can touch a line of code and still fail to check whether that line behaved correctly, so coverage measures reach, not quality.

Here is a worked example. A development team reports ninety per cent test coverage on a new feature, and treats that figure as a sign the feature is safe to ship. A later misuse-case test, which the automated coverage tool had never been asked to generate, feeds the feature a deliberately malformed input the normal test suite never tried, and the feature crashes.

Exam tip: ISC2 tests coverage as a measurement of extent, not of quality. A question asking what high test coverage proves should be answered carefully: it shows how much code ran during testing, not that the code is free of security flaws.

8. Interface testing, BAS and compliance checks



A system is rarely attacked through its core logic directly. It is attacked through the doors it exposes to the outside: its interfaces.

Interface testing checks each of a system's exposed surfaces in turn. User interface testing checks what a human can do through a screen or form. Network interface testing checks what is reachable over the network itself, port by port. API interface testing checks the programmatic interfaces other systems call directly, which often skip the careful validation a human-facing screen applies, and so need testing in their own right.

A newer technique automates a whole attack, rather than one interface at a time. **Breach and attack simulation**, or BAS, uses software to run that attack safely and repeatedly against the organisation's own infrastructure. Gartner describes BAS tools this way: "continually and consistently simulate the full attack cycle, including insider threats, lateral movement and data exfiltration, against enterprise infrastructure."

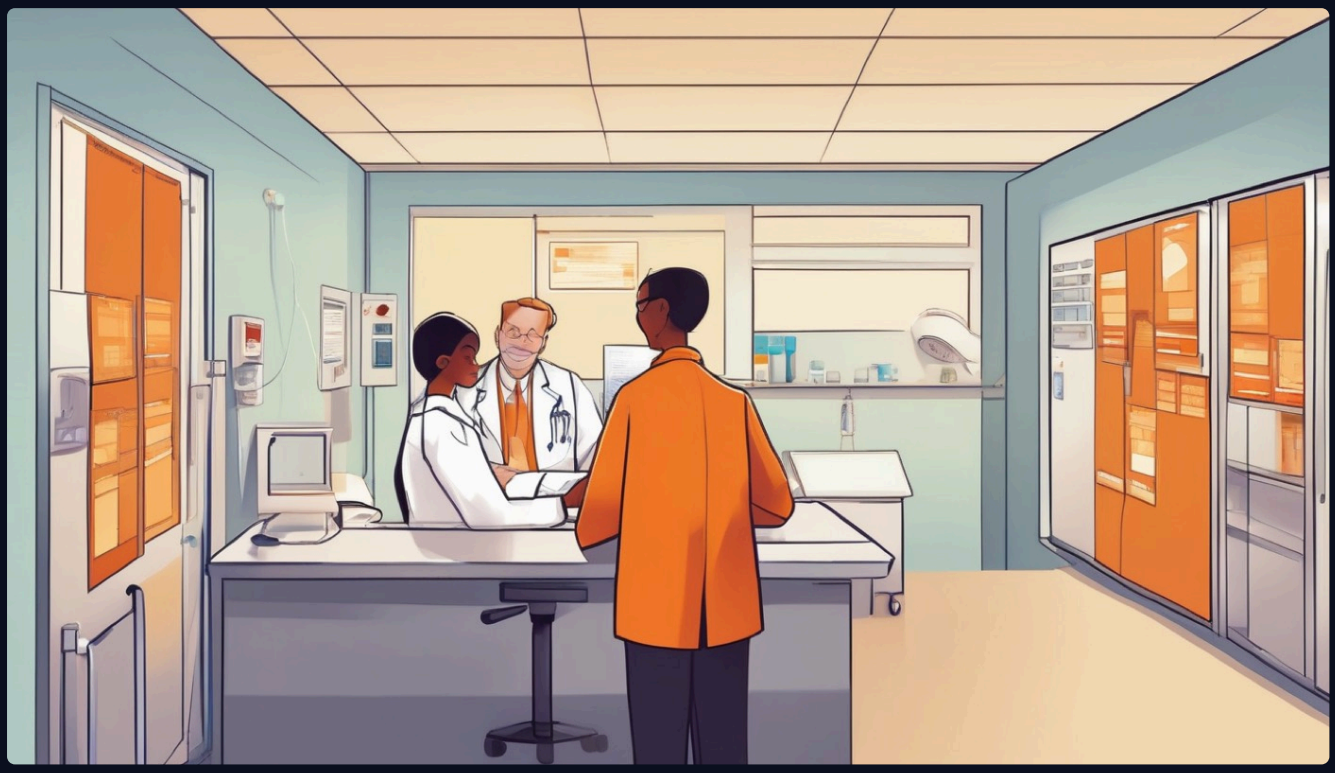
Where a single pentest is a snapshot taken once or twice a year, BAS runs the same kind of attack continuously, flagging the moment a defence that worked last month stops working after a change. The two are complementary rather than competing: a pentest brings a skilled human's judgement to a scoped target, and BAS brings repeatable, automated coverage between those human-led exercises.

The last technique in this sub-domain checks a system against rules rather than against attackers. A **compliance check** compares a control against a specific regulatory or contractual requirement, such

as a payment-card standard or a data-protection law, and reports where the system falls short of that named requirement rather than where it is simply vulnerable.

Exam tip: ISC2 tests BAS as continuous and automated, against a pentest's periodic and human-led nature. If a question describes a tool that runs the same attack chain every night without a person driving it, that is BAS, not a pentest.

9. Security process data: accounts and management review



Sub-domain 6.3 moves away from one-off tests, towards the ongoing data a security programme produces simply by operating day after day.

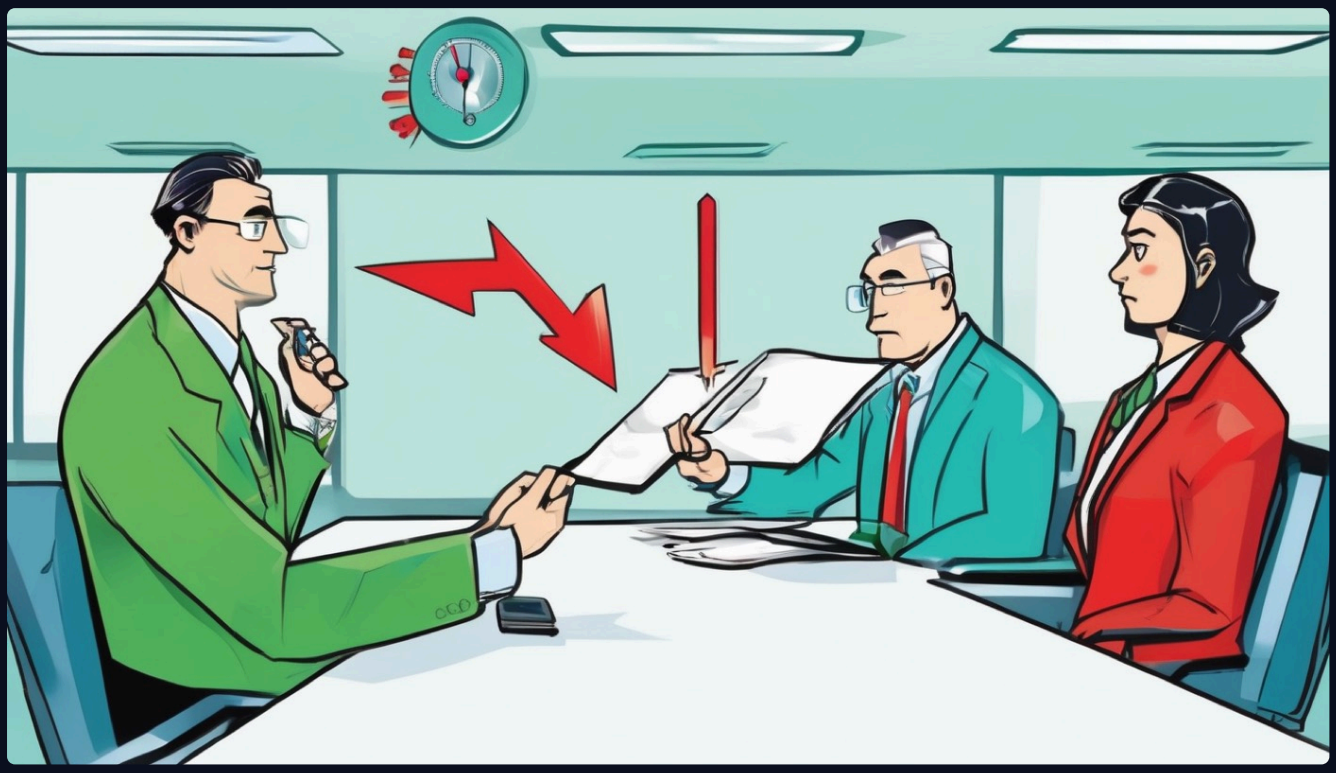
An **account management review** periodically checks every account against who should actually hold it: is the person still employed, still in the same role, still needing the same access. **Management review and approval** adds a second layer. A manager, not just a system, formally signs off that an account's access is still appropriate. That creates a record that a human looked and agreed, not just that a report was generated and filed.

Here is a worked example. A quarterly account review at a hospital finds an account belonging to a nurse who transferred to an administrative role eight months earlier, still holding access to the clinical records system she no longer needs for her new job.

Nobody had done anything malicious with the account, and that is exactly the point: a stale permission does not need to be misused to be a finding. The review closes the access, and the hospital's manager signs off the closure, creating the evidence trail that the gap was found and fixed, not merely noticed and left.

Exam tip: ISC2 frequently tests account reviews as ongoing evidence rather than a one-time check. A scenario describing access removed only when an employee leaves, with nothing checked in between, is describing a gap in the review process, even if no breach has happened yet.

10. KPIs, KRIs and measuring the security programme



Individual findings tell you about one system. To tell whether the whole security programme is getting better or worse, an organisation needs trend data, and that is what this chapter's two measures provide.

A **key performance indicator**, or KPI, measures how well a process is running — for example, the percentage of known vulnerabilities fixed within the time the organisation's own policy allows. A **key risk indicator**, or KRI, signals rising exposure to a risk before it becomes a loss. For example: a rising count of vulnerabilities that have sat past their fix deadline. The two are related but answer different questions: a KPI asks how well are we doing our job, and a KRI asks how much danger are we exposed to right now.

Here is a worked example. A security programme's patch-completion KPI looks healthy month after month, holding near its target. Underneath that steady KPI, a KRI tells a different story. The number of critical vulnerabilities still unpatched after their deadline has quietly climbed for three straight months. The team has been closing a growing pile of easy, low-risk patches, while a small number of harder, high-risk ones keep rolling over.

Leadership reading only the KPI would see a healthy-looking number and move on. Reading the KRI alongside it shows the real risk is rising even while the headline performance figure looks fine, which is exactly why a mature programme tracks both rather than either alone.

Exam tip: ISC2 tests the distinction by asking which measure to use for a given purpose. A question about whether a process is meeting its own targets points to a KPI; a question about whether danger

is building up, regardless of whether targets are being met, points to a KRI.

11. Backup verification, DR and business continuity data



A backup that has never been tested is a belief, not a control. This chapter is about the evidence that turns that belief into something provable.

Backup verification means actually restoring data from a backup and checking it works, not merely confirming that a backup job completed without an error message. A completed job proves data was written somewhere; a successful restore proves the data can be read back and used. **Disaster recovery**, or DR, is the process of restoring systems after a disruption. **Business continuity**, or BC, is the broader plan for keeping the organisation's critical functions running at all, with or without those systems.

Here is a worked example. A finance team's nightly backup job has reported success every night for a year. During a scheduled DR exercise, the team tries to restore from one of those backups and discovers the backup file itself is corrupted, and has been for months, with every "successful" job quietly backing up the same broken file.

Nobody had tried an actual restore until the exercise forced one. That is the whole argument for verification: the only backup an organisation can trust is one it has already proven it can read back, and DR and BC exercises are where that proof gets produced on a schedule rather than during a real emergency.

Exam tip: ISC2 tests backup verification as restoring and checking, not completing and logging. A scenario describing a backup process that only checks for error messages, with no restore ever attempted, is describing an unverified backup, however clean its logs look.

12. Training and awareness data



Technical controls protect systems. This chapter's data measures the part of the control environment that is a person, not a system, and that data is just as much evidence as a scan report.

Training completion data tracks who has actually finished required security training and when, giving the organisation a measurable view of how much of its workforce has had the instruction a control relies on. **Phishing-simulation data** goes further than completion: it sends realistic but fake phishing emails to staff and records who clicked, who reported it, and how quickly, measuring behaviour rather than attendance.

Here is a worked example. A company's training-completion rate sits at ninety-eight per cent, which looks like strong evidence the workforce understands phishing risk. Its most recent phishing simulation shows a click rate that has barely moved from the simulation run before the training rolled out, even though almost everyone sat through the course.

Finishing a course is not the same as changing behaviour under real pressure, and that gap is exactly why both figures get tracked rather than just one. A security programme that only measures completion can look successful while the actual human control it is paying for has not improved at all.

Exam tip: ISC2 tests training metrics as behaviour evidence, not attendance evidence. A question treating a high completion rate alone as proof that security awareness has improved is describing the weaker of the two measures this chapter covers.

13. Analysing test output and reporting



Every technique covered so far produces raw output: scan results, pentest notes, review findings, process data. Sub-domain 6.4 begins with turning that pile into something a decision-maker can act on.

Analysing test output means sorting raw findings by actual risk, not by how many of them a tool happened to generate or how technically interesting each one is. A reported finding carries little weight until it is matched to what it would cost the organisation if it were exploited, and how likely that is given the system it sits on. **A report written for a non-technical audience leads with that risk, in the business's own terms, and keeps the technical detail available but secondary.**

Here is a worked example. A vulnerability scan returns four hundred findings for one business unit, most of them low-severity and already scheduled for a routine patch cycle. Three of the four hundred are a critical flaw on an internet-facing server that handles customer payments, and that server's owner is not in the room when the raw list is first reviewed.

A report that lists all four hundred findings in the order the scanner produced them buries the three that matter under the three hundred and ninety-seven that do not. The analyst instead writes a one-page summary naming the payment server's three critical findings first, with the remainder grouped and summarised beneath, so the reader who approves budget sees the real risk in the first sentence.

Exam tip: ISC2 tests report quality as fitness for its audience, not completeness. A question praising a report purely for including every single finding, with no prioritisation, is describing a report that technically informs nobody.

14. Remediation and exception handling



A finding does not fix itself once it is reported. This chapter covers what happens to it next, and the two outcomes every finding eventually reaches.

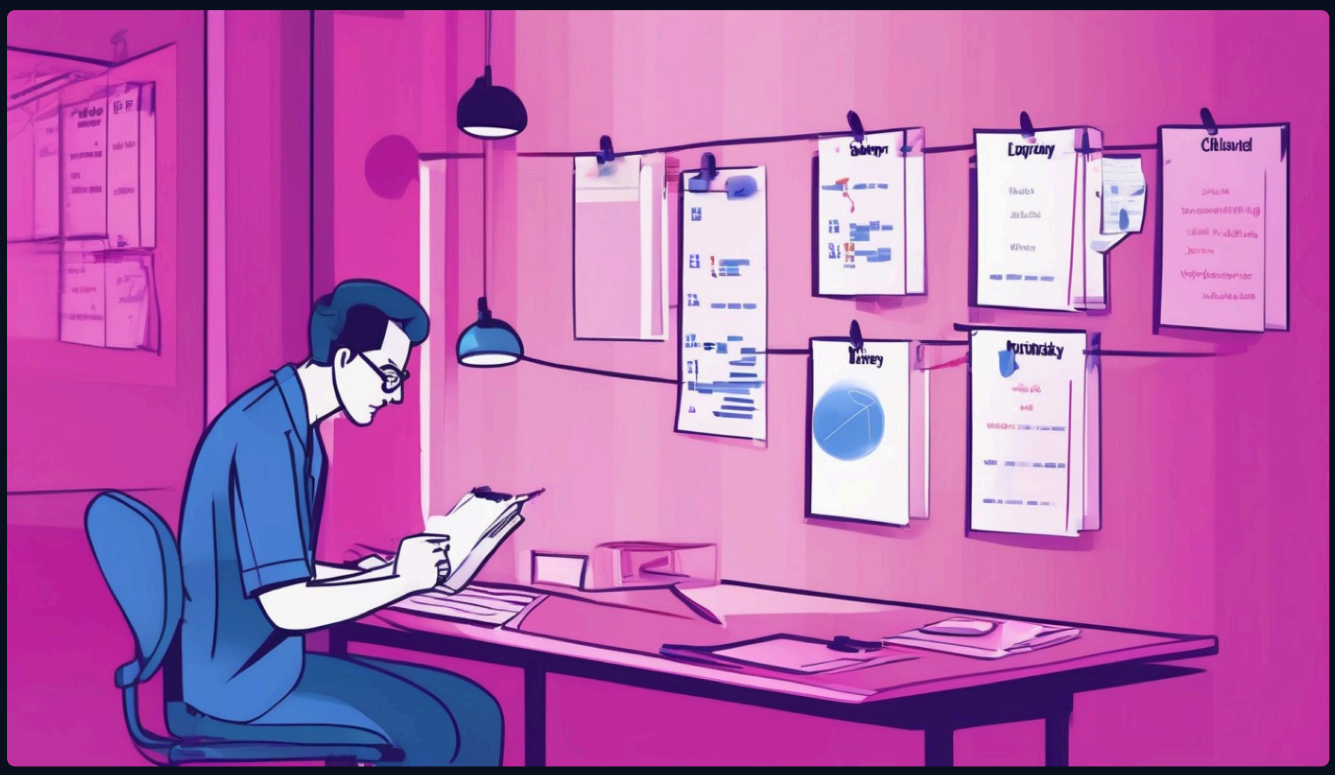
Remediation is fixing the underlying problem a finding describes, prioritised by risk rather than by the order findings happened to arrive in. Not every finding gets remediated on the timeline its severity would suggest. When it does not, the organisation needs a formal, honest alternative to simply ignoring it: an **exception**, sometimes called a risk acceptance. An exception is a documented, time-limited, signed-off decision that a specific risk will be tolerated rather than fixed, for a stated reason, until a stated expiry date.

Here is a worked example. A manufacturing plant finds a critical vulnerability on a control system that cannot be patched without a planned shutdown, and that shutdown cannot happen for another four months without halting production. The security team cannot remediate the flaw immediately, so a risk owner formally accepts the risk, in writing, with compensating controls in the meantime and a fixed date the exception expires.

That expiry date is the detail that separates an exception from simply looking away. Without it, "temporary" risk acceptances have a well-earned reputation for quietly becoming permanent, because nothing forces anyone to revisit a decision that never comes back onto anyone's calendar.

Exam tip: ISC2 tests exceptions as formal, time-limited and signed-off, never informal or open-ended. A scenario describing a team that simply decided not to fix something, with no documented owner and no expiry, is describing a missing control, not a valid exception.

15. Ethical disclosure



Sometimes the flaw a test finds is not in the organisation's own system at all, but in a vendor's product, or in software the whole industry relies on. This chapter is about what happens then.

Ethical disclosure, also called responsible or coordinated disclosure, is the process of telling an affected vendor about a vulnerability privately first, giving them a fair chance to fix it before any public detail is released. Going public immediately, with no warning, is sometimes called full disclosure, and it forces a fast fix at the cost of leaving users exposed the moment the detail is known. Many researchers instead work to a fixed disclosure window, commonly around ninety days, which gives the vendor a known deadline rather than an open-ended wait.

Here is a worked example. A researcher at a security consultancy finds a serious flaw in a widely used piece of network hardware during an unrelated engagement for a client. The consultancy privately notifies the hardware vendor, agrees a ninety-day window for a fix, and publishes full technical detail only after that window closes, once the vendor has shipped a patch.

That timing matters on both sides. Disclosing too early, before a fix exists, hands attackers a working flaw and no defence. Disclosing too late, or never, leaves users exposed for longer than necessary while the vendor has no real deadline forcing action.

Exam tip: ISC2 tests ethical disclosure as a balance between urgency and responsibility. It is never a quiet report-it-and-walk-away, and never an immediate publish-everything move. The correct answer almost always involves notifying the vendor first, agreeing a timeline, and only then considering any public detail.

16. Internal audits



Sub-domain 6.5 moves from testing controls to formally auditing them, and the first of three audit types starts inside the organisation itself.

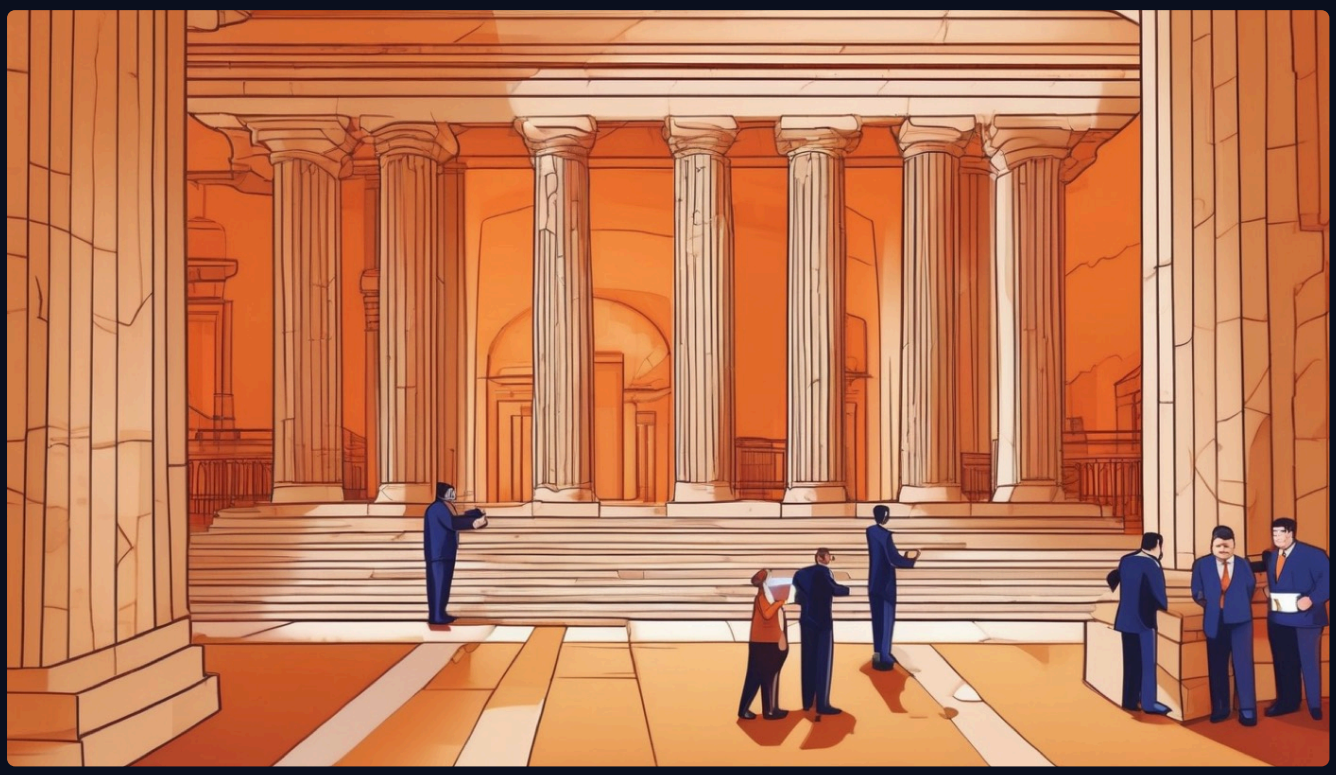
An **internal audit** is run by staff employed by the organisation being audited. **Independence** still has to be real, even so: internal auditors must report to a function separate from, and senior to, whatever they are auditing. That reporting line usually runs all the way up to the board's own audit committee, rather than to the manager of the team being checked. Without that separation, an internal audit is just a self-review wearing an audit's name.

Here is a worked example. An internal audit team reviewing a company's change-management process finds that emergency changes to production systems are frequently approved after the fact rather than before, which breaks the control's whole purpose of reviewing a change before it goes live.

Because the audit team reports to the audit committee rather than to the IT director whose process they just criticised, the finding goes into a formal report the committee itself reviews, rather than being quietly smoothed over by the person whose process failed. That reporting line is what makes the finding mean something.

Exam tip: ISC2 tests internal-audit independence as a reporting-line question. A scenario where the people auditing a function report to the manager of that same function is describing broken independence, regardless of how skilled or honest the auditors personally are.

17. External and third-party audits



Independence inside an organisation only goes so far. Some situations call for a reviewer who is not an employee at all.

An **external audit** is commissioned by the organisation itself, but carried out by an outside firm, usually to satisfy a legal, regulatory or investor requirement for a genuinely independent opinion. A **third-party audit** goes a step further: it is commissioned by, or for the benefit of, someone outside the organisation entirely – most often a customer who wants assurance about a supplier before trusting that supplier with their own data.

Here is a worked example. A software company's biggest customer, a bank, will not renew its contract without evidence of a current, independent security audit covering the systems that process the bank's data. The software company commissions an external audit from a recognised audit firm specifically to satisfy that customer's requirement, separate from any internal review it was already running.

The audit firm's report becomes something the software company hands directly to the bank, which is exactly why a third-party audit's value depends on the auditor's independence from the company being audited, not just from the customer requesting it.

Exam tip: ISC2 tests the difference by asking who the audit is FOR. An audit commissioned by the organisation about itself, run by outsiders, is external. An audit demanded by, or run for, someone else entirely is third-party – even when the same audit firm carries out both.

18. SOC reports and audit standards



Third-party assurance over outsourced services has its own well-known report family, and this chapter is about reading one correctly.

SOC 1 reports on controls relevant to a service organisation's effect on its customers' financial reporting. **SOC 2** reports on security-relevant controls more broadly, written against the AICPA's Trust Services Criteria. **SOC 3** is a shorter, public-facing summary of a SOC 2 report, written for a general audience rather than for a customer's own auditors. Within SOC 2, a **Type I** report covers whether controls were suitably designed at a single point in time; a **Type II** report covers whether those controls actually operated effectively over a period, typically six to twelve months.

The AICPA's Trust Services Criteria, the standard a SOC 2 auditor works against, cover five categories: security, availability, processing integrity, confidentiality and privacy. Security is mandatory in every SOC 2 report; the other four are included only where they are relevant to the service being reported on. The criteria were set out in 2017 and had their points of focus revised in 2022, and remain the current standard.

Here is a worked example. A customer evaluating a cloud vendor is handed a SOC 2 Type I report and treats it as full assurance the vendor's controls work. A Type I report only confirms the controls were well designed on the day they were checked. It says nothing about whether they kept operating correctly the following month — only a Type II report, covering a period rather than a moment, can show that.

Exam tip: ISC2 tests Type I against Type II as a point-in-time snapshot against sustained operation over a period. A scenario needing assurance that controls kept working month after month needs Type II; Type I alone is not enough evidence for that claim.

19. Putting the cycle together: a worked scenario



Every sub-domain in this module has been covered on its own. This chapter follows one organisation's year, showing how all five connect into a single running cycle.

A mid-sized logistics company begins its year by designing its assessment strategy under 6.1: monthly internal vulnerability scans, an annual external pentest, and a third-party SOC 2 audit demanded by its largest customer. Under 6.2, the monthly scans turn up a set of CVE-identified flaws, CVSS-scored for urgency. The annual pentest's red team finds a separate, more serious path into the warehouse-management system that no automated scan had flagged.

Under 6.3, the company's account reviews catch two stale accounts, its KRI for overdue critical patches ticks upward for a month, and a DR exercise proves a backup restore works cleanly. Under 6.4, an analyst writes a single prioritised report covering the scan findings and the pentest's warehouse-system path, remediates the warehouse flaw within the week, and formally accepts a lower-risk exception on an older system scheduled for retirement in three months.

Under 6.5, the external pentest firm's report and the internal remediation records both become evidence the third-party SOC 2 Type II auditor reviews months later, covering the whole period rather than a single day. The audit passes, specifically because the company could show its controls had been tested, its findings tracked, and its one exception formally signed off with an expiry date rather than quietly forgotten.

That is the whole of Domain 6 in one running year: a strategy decided in advance, tests that produce evidence, and data collected continuously. A report that drives action, and an audit that checks the

whole cycle actually happened the way it was supposed to.

20. How Domain 6 links to the other seven domains



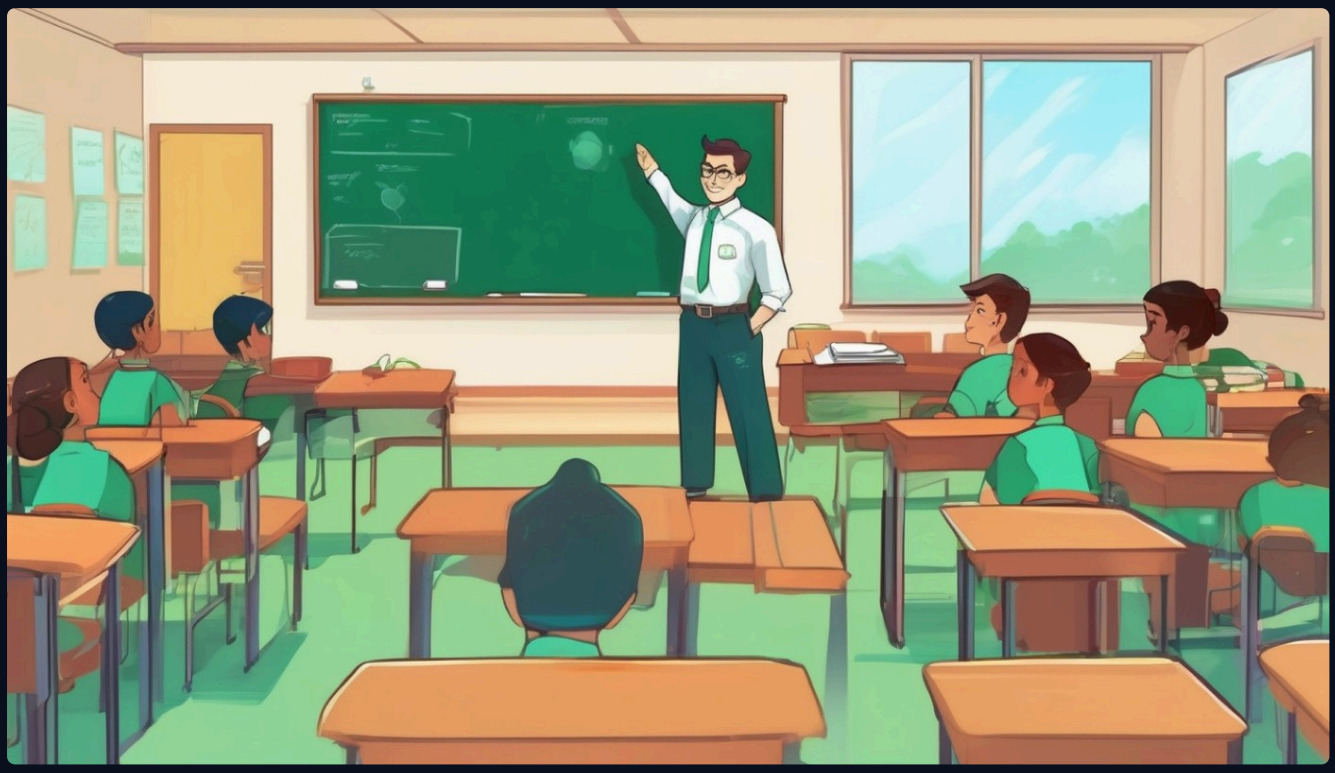
Domain 6 does not operate alone. Its output is the raw material several other domains depend on to do their own jobs.

Domain 1, Security and Risk Management, uses Domain 6's findings and KRIs as the evidence a risk register is built from; a risk nobody has tested for is a risk nobody can actually size. Domain 5, Identity and Access Management, depends on Domain 6's account management reviews to prove access controls are working as designed, not just configured correctly on paper. Domain 7, Security Operations, picks up continuous monitoring once Domain 6's periodic tests are done, watching the same controls day to day between one scan and the next. Domain 8, Software Development Security, relies on Domain 6's code review and misuse-case testing to catch flaws before software ever ships.

None of those other domains can prove its own claims without Domain 6's evidence. A risk register, an access review, an operations dashboard and a secure development process are all, underneath, statements of belief until something from this domain actually tests them.

That is why this module sits in the middle of the eight-domain series rather than at either end: it is the domain every other domain eventually needs something from.

21. Domain 6 recap



Before the review questions, here is everything this module has covered, sub-domain by sub-domain.

Sub-domain 6.1 is designing an assessment, test and audit strategy. It chooses between internal, external and third-party reviews, based on what independence a situation actually requires. Sub-domain 6.2 is the testing techniques themselves: vulnerability assessment and CVSS-scored CVE flaws, and penetration testing with red, blue and purple teams. And rules of engagement setting black, grey and white box scope.

Sub-domain 6.2 also covers log reviews, synthetic transactions and benchmark comparisons. It covers code review, misuse-case testing and test coverage. And it covers interface testing, breach and attack simulation, and compliance checks.

Sub-domain 6.3 is collecting the ongoing evidence a security programme produces: account management reviews, KPIs and KRIs tracked side by side, and verified backups. Disaster recovery and business continuity exercises, and training and phishing-simulation data measured as behaviour, not attendance.

Sub-domain 6.4 is analysing test output into a risk-prioritised report. It is remediating by risk, formally time-limiting any exception, and disclosing a found vulnerability responsibly on an agreed timeline.

Sub-domain 6.5 is internal audits with a genuinely independent reporting line, external and third-party audits for outside assurance, and the SOC report family. SOC 1, 2 and 3, and Type I against

Type II, are read against the AICPA's Trust Services Criteria. Every one of those pieces feeds the other seven CISSP domains, which is why this domain earns its twelve per cent of the exam.

That is the whole of Domain 6. Now, the review.

22. Review: exam-style questions, part one



Here are sixteen exam-style questions on Domain 6, in three parts. Each is followed by a short pause to think, then the answer and why the other options are wrong.

Question 1. Under the ISC2 2024 CISSP exam outline, what is Domain 6's weighting? A) Eight per cent. B) Twelve per cent. C) Sixteen per cent. D) Twenty per cent.

Answer: B. Domain 6 is twelve per cent of the exam. The outline took effect on 15 April 2024. None of the other figures correspond to Domain 6 under that outline.

Question 2. Which identifies a publicly known vulnerability, and which scores its severity? A) CVSS identifies; CVE scores. B) CVE identifies; CVSS scores. C) Both identify; neither scores. D) Both score; neither identifies.

Answer: B. CVE assigns the identifier; CVSS supplies the zero-to-ten severity score, which rules out A, C and D.

Question 3. A tester is given a standard customer account but no source code. What kind of test is this? A) Black box. B) White box. C) Grey box. D) Blue box.

Answer: C, grey box. Partial knowledge is grey box; no knowledge would be black box, full knowledge white box, and "blue box" is not one of the three categories.

Question 4. What does a synthetic transaction primarily test? A) Whether a server responds to a network ping. B) Whether a scripted business transaction actually completes correctly. C) Whether a

password meets complexity rules. D) Whether a backup job logged success.

Answer: B. A synthetic transaction runs a fake but realistic transaction through a live system to check the business function itself works, which is a stronger check than a simple ping, ruling out A; C and D describe unrelated checks.

Question 5. What does high test coverage actually prove? A) The code has no security flaws. B) How much of the code's logic was exercised by the tests. C) The code passed a misuse-case test. D) The application is compliant with a named regulation.

Answer: B. Coverage measures how much code ran during testing, not whether it ran correctly or securely, which rules out A, C and D.

Question 6. What best describes breach and attack simulation (BAS)? A) A once-a-year human-led pentest. B) Continuous, automated simulation of a full attack cycle against live infrastructure. C) A static code-scanning tool. D) A manual log review process.

Answer: B. BAS runs automated attack simulations continuously and repeatedly, unlike a periodic human-led pentest, ruling out A; C and D describe entirely different techniques.

23. Review: exam-style questions, part two



The second part of the review continues straight on, same format as before: think it through, then hear which answer is right and which are not.

Question 7. An internal account review finds an account with access its holder no longer needs, though nothing was misused. What should happen? A) Nothing, since no harm occurred. B) The access should be closed and the closure signed off. C) The account should be deleted immediately with no review. D) The finding should wait until the next annual review.

Answer: B. A stale permission is a finding on its own regardless of misuse, and closing it with sign-off creates the evidence trail a review is meant to produce, which rules out A, C and D.

Question 8. A KPI shows steady patch performance while a related KRI climbs for three straight months. What does this show? A) The KPI is wrong and should be ignored. B) Risk can rise even while a performance target looks healthy. C) KRIs are only useful when KPIs are failing. D) The two measures always move together.

Answer: B. That is exactly why a mature programme tracks both: a steady KPI can hide rising risk a KRI reveals, which rules out A, C and D.

Question 9. What does a successful backup verification actually require? A) A backup job reporting no errors. B) A completed backup with no restore attempted. C) A full restore of the data, checked to confirm it works. D) A written policy stating backups are taken daily.

Answer: C. Only an actual restore proves the data can be read back and used. A clean job log alone, as in A, B and D, proves nothing about whether the backup actually works.

Question 10. What does a phishing-simulation click rate measure that a training-completion rate does not? A) How many staff attended a course. B) Actual behaviour under a realistic phishing attempt. C) Whether training was delivered on time. D) The cost of the training programme.

Answer: B. Completion measures attendance; a simulation measures what staff actually do when tested, which is the behaviour the training was meant to change, ruling out A, C and D.

Question 11. What makes a risk acceptance a valid exception rather than simply ignoring a finding? A) A manager mentioning it informally. B) A documented, signed-off decision with a stated expiry date. C) Leaving the finding open with no further action. D) Deleting the finding from the report.

Answer: B. Formality, sign-off and a fixed expiry are what separate a genuine exception from a risk quietly left unmanaged, ruling out A, C and D.

24. Review: exam-style questions, part three



The third and final part of the review closes it out, same format again: weigh the options, then find out what the right call is and why.

Question 12. What is the defining feature of ethical, coordinated disclosure? A) Publishing full technical detail immediately. B) Notifying the vendor privately first, with an agreed timeline before any public detail. C) Never disclosing a flaw to anyone. D) Only disclosing flaws found by the vendor's own staff.

Answer: B. Coordinated disclosure gives the vendor a fair, time-limited chance to fix the flaw before any public release, which rules out A, C and D as either reckless or unworkable.

Question 13. Why does internal-audit independence require a specific reporting line? A) Auditors must report to the manager of the function they are auditing. B) Auditors must report to a function separate from, and senior to, what they audit. C) Independence only matters for external auditors. D) Reporting lines are irrelevant as long as auditors are skilled.

Answer: B. Reporting to the very function being audited, as in A, defeats independence regardless of skill, which also rules out D. C is false too, because internal auditors need independence of their own.

Question 14. A SOC 2 report's auditor is hired directly by a company's biggest customer to assure the customer about that company. What type of audit is this? A) Internal. B) External. C) Third-party. D) Self-assessment.

Answer: C, third-party. An audit commissioned by, or for, someone outside the organisation being audited is third-party, not external. External is commissioned by the organisation about itself; A and D describe unrelated situations.

Question 15. What does a SOC 2 Type II report prove that a Type I report does not? A) That controls were well designed on one specific day. B) That controls actually operated effectively over a period, typically six to twelve months. C) That the organisation has no vulnerabilities. D) Nothing; the two types are identical.

Answer: B. Type I is a design snapshot at a point in time. Type II covers sustained operation over a period, which is the stronger assurance — ruling out A, C and D.

Question 16. Which domain's work most directly depends on Domain 6's account management reviews to prove its controls are actually working? A) Domain 4, Communication and Network Security. B) Domain 5, Identity and Access Management. C) Domain 6 itself, with no external dependency. D) Domain 8, Software Development Security.

Answer: B. Account management reviews are Domain 6's direct evidence that Domain 5's access controls are working in practice, not just configured correctly on paper, which rules out A, C and D.

That closes the review, and this module. Domain 6 is the part of the CISSP CBK that proves every other domain's claims are true — and now you have been through every sub-domain of it, with a worked example and an exam-style question for each.

Sources and credits

Primary sources

- ISC2, 'CISSP Certification Exam Outline', effective 15 April 2024 (isc2.org) - Domain 6 weighted 12%; the five sub-domain structure (6.1-6.5); exam format (CAT, 125-175 items, 4 hours, pass mark 700/1000). Re-checked by web search 11 October 2026.
- ISC2's own CBK-update material, reported by a Training Camp press release, 2015, cross-checked against cm-alliance.com, 'Comparison of the old and new CISSP syllabus', April 2016 - the 2015 restructure from ten domains to eight, gathering testing and audit material into the new Domain 6.
- MITRE/CVE.org, CVE Program overview - continuous operation since September 1999.
- The Register, 'CVE program gets last-minute funding from CISA - and maybe a new home', 16 April 2025, corroborated by Help Net Security and CSOonline, same date - the averted funding lapse and eleven-month extension. Re-checked 11 October 2026.
- FIRST.org, CVSS v3.1 Specification Document, June 2019, and CVSS v4.0 Specification Document, November 2023 - the current scoring standard and its severity bands. Re-checked by web search 11 October 2026 against Mend.io, Red Canary and BleepingComputer coverage; no v4.1 found.
- NIST Special Publication 800-115, 'Technical Guide to Information Security Testing and Assessment', September 2008 - still the current, active guide, no revision issued. Re-checked 11 October 2026.
- AICPA, Trust Services Criteria (TSP Section 100), 2017 as revised 2022; Description Criteria updated July 2025 - the SOC 2 report family and its five categories. Re-checked by web search 11 October 2026 against Secureframe's 2025 summary.
- Gartner's definition of breach and attack simulation, as quoted by Picus Security and Rapid7 vendor glossaries, re-checked 11 October 2026.

Not regulated financial advice.