Dweve
Weaving Intelligence

ENGLISH SOURCE EDITION · JULY 2026

FOUNDATION AND DECISION GUIDE · F10

# Security architecture and assurance

Assets, actors, trust boundaries, identities, capabilities, data, models, tools, supply chain, detection, recovery, and a scoped security proof.

DECISION SCOPE EVIDENCE OWNERSHIP PROOF EXIT


A complete guide to a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance.

F10

CONTENTS

# From the first question to a defensible decision

This 53-page guide turns 17 owned topics into an explanation, evidence requirement, proof challenge, and closing decision record.

| PRIMARY READERS | READING DEPTH | DECISION THIS SUPPORTS |
| --- | --- | --- |
| Security architects and assurance teams | Deep technical | Determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment. |



**TARGET LENGTH**

12,000–17,000 words.. Page count varies with the number and depth of topics rather than forcing every subject into the same shell.

**SOURCE DISCIPLINE**

Dweve-specific facts are taken directly from the English website text audited 2026-07-20; web-navigation wording is humanised for print. Evaluation guidance is editorial, and scenarios and derived visuals are illustrative. Unsupported synthesis is replaced by canonical copy or omitted.

READ THIS FIRST · READER DECISION

# Who should use this guide, and what decision it supports

CISOs, security architects, SOC teams, product security, privacy engineering, risk, and technical assurance teams.

Determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment.

The worked scenario is a security team attacks one representative workflow across identity, source, model/tool, capability, human, target, evidence, managed or local infrastructure, update and recovery while preserving an investigation record. It is illustrative and deliberately bounded. Replace it with the organisation's actual workflow, systems, data, sources, policies, roles, infrastructure, commercial conditions and consequences while preserving the evidence discipline.

The guide's concrete output is a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance. Use the chapter explanation pages to align language, the evidence pages to create acceptance records, and any challenge pages to test whether the boundary survives missing assumptions and failure. End in a named decision, not an attractive summary.

PROTECT

**Identity + data + work**

Current guide lens; verify source and scope.

CONTAIN

**Models + tools + runtime**

Current guide lens; verify source and scope.

PROVE

**Events + packets + tests**

Current guide lens; verify source and scope.

RECOVER

**Keys + state + history**

Current guide lens; verify source and scope.

**PRIMARY RISK**

The highest-impact path may cross trusted prompts, poisoned sources, compromised dependencies, over-capable tools, human/admin bypass, external target replay, packet disclosure, support access or an update channel even when the core network perimeter is strong.

READ THIS FIRST · DISCIPLINE

# How to read claims, evidence, responsibility, and status

The website is the only factual source for this edition. The guide adds explanation, architecture and proof requirements without inventing current product, regulatory, assurance, commercial, or customer-result facts.

## Six evidence layers

A published description states a current proposition. Delivered software and documentation show what is available in a selected version. Customer configuration shows how it is used. An observed run shows one result. Independent assurance has scope, method and date. A written agreement creates commercial commitments and remedies.

## Responsibility stays explicit

The relevant owners include CISO and risk owner, product/application security, security architecture, SOC/incident response, identity and keys, privacy/data, model/tool and source owners, platform operations, evidence/records, supplier and independent assurance. The supplier can provide product, evidence and commitments; customer owners still decide purpose, source and policy authority, professional or domain judgement, acceptable risk, infrastructure, staffing, affected-person treatment and production reliance.

## Status and strong statements

Measured claims stay tied to workload, hardware and method. Research remains research. Alignment is not certification. “Open” is verified per component and licence. Pricing is re-audited at release. Roadmap and availability are not assumed. Conflicts or ambiguity become diligence items.

| LAYER | EVIDENCE | CANNOT ESTABLISH ALONE |
|---|---|---|
| **Website** | website copy + captured date | Delivery, fit, contract |
| **Delivery** | Manifest + docs + artefacts | Customer result |
| **Configuration** | Data/roles/policy/ posture | Operating outcome |
| **Run** | Raw output + target + packet | General performance |
| **Assurance** | Scoped test/report + findings | Universal security/compliance |
| **Agreement** | Rights + duties + remedy | Technical behaviour without test |

Keep the layers separate in every chapter and decision record.

**READING RULE**

Architecture describes intended controls; implementation and configuration realise them; tests and operational evidence show behaviour; independent reports have scope and date; certification or regulatory status must be stated exactly. Open or inspectable code does not remove vulnerability or customer hardening duties.

READ THIS FIRST · CANONICAL SPOTLIGHT

# The security model follows the work

Protect the complete lifecycle instead of drawing one perimeter around an undefined “AI system”.

The work begins with authenticated identity, purpose and scope; enters through a typed contract; resolves sources and context; selects models, agents and tools; executes under capabilities and resource limits; crosses human gates; may affect an external target; writes events and evidence; and remains accessible to operators, reviewers, support and export. Each transition is a trust boundary.

High-value assets include data and knowledge, policies and rules, models and prompts/programmes, tools and credentials, decisions and target capabilities, human authority, product state, keys and trust policy, packets and archives, packages/build systems, infrastructure and administrative paths. Threat actors include insiders, compromised sources/providers/dependencies, malicious users, tenants, plugins, partners and support.

The assurance package ties design to implementation and observation: architecture/data flows, identity and capability model, key lifecycle, SBOM/build provenance and signatures, test results, findings/remediation, incident process and sampled records, posture-specific controls, backup/restore, responsibility and residual-risk acceptance.

| AREA | CURRENT GUIDE RECORD | VERIFY / DECIDE |
|---|---|---|
| **Ingress** | Identity, purpose, schema, limits | Impersonation, cross-scope, malicious input |
| **Context** | Sources, data, model/tool selection | Injection, poisoning, substitution, leakage |
| **Execution** | Sandbox, capabilities, resources | Escape, denial, excess action |
| **Authority** | Human gate and administration | Bypass, rubber stamp, privilege conflict |
| **Effect / evidence** | Target reconciliation, packets, keys | Replay, forgery, disclosure, erasure |
| **Supply / operate** | Build, update, support, incident, restore | Compromise, persistence, dependency failure |

Current source-edition summary. Recheck publication-sensitive facts and delivered scope at the decision date.

**GUIDE-SPECIFIC PROOF**

Run access, injection, isolation, capability, resource, secret, model/tool substitution, target replay, packet tamper, key rotation, signed-update, no-egress, support/admin, incident and clean-room recovery tests; map every finding to owner, risk, remedy, retest and accepted residual state.

01

SECURITY ARCHITECTURE AND ASSURANCE

# Understand the decision and its language

Define the real question, current state, responsibilities, source claims, and the artefact this guide must produce.


The same evidence discipline applies to every chapter in the section.

CHAPTER 01 · EXPLAIN

# Scope and assurance language

architecture claim, control, test, evidence, alignment, and certification.

This chapter owns a specific part of the reader's decision: architecture claim, control, test, evidence, alignment, and certification.

In this guide, security is expressed across the complete work path: identity and purpose, source and context, models and agents, tool capabilities, human gates, target actions, evidence and keys, deployment and management planes, packages and updates, support, incident response, recovery and exit. Applied to "Scope and assurance language", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Architecture describes intended controls; implementation and configuration realise them; tests and operational evidence show behaviour; independent reports have scope and date; certification or regulatory status must be stated exactly. Open or inspectable code does not remove vulnerability or customer hardening duties. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

### Architecture claim

Define the exact proposition and current authoritative source.

LENS 2

### Control

Name the responsible person, system, dependency, and version.

LENS 3

### Test

Specify the evidence and how an independent reviewer checks it.

LENS 4

### Evidence

Record scope, failure, unresolved question, and safe next action.

**OWNED ARTEFACT**

This chapter contributes to a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance.

CHAPTER 01 · APPLY

# Turn “Scope and assurance language” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, architecture claim, control, test, evidence, alignment, and certification., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant source and version, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: a security team attacks one representative workflow across identity, source, model/tool, capability, human, target, evidence, managed or local infrastructure, update and recovery while preserving an investigation record. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Run access, injection, isolation, capability, resource, secret, model/tool substitution, target replay, packet tamper, key rotation, signed-update, no-egress, support/admin, incident and clean-room recovery tests; map every finding to owner, risk, remedy, retest and accepted residual state. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

| QUESTION | REQUIRED RECORD | OWNER / STOP |
|---|---|---|
| **What is claimed?** | Exact wording, scope, date, source | Guide decision owner |
| **What exists?** | Delivered manifest, interface, artefact | Technical/product owner |
| **What is configured?** | Customer data, roles, rules, posture | Customer control owner |
| **What happened?** | Raw run, target state, packet, metric | Domain/operations owner |
| **What is assured?** | Test/report scope, findings, limits | Independent challenger |
| **What is committed?** | Offer/agreement, remedy, exit term | Procurement/legal owner |

Evidence ladder for scope and assurance language; not every question needs every row, but no row may be implied without evidence.

**DECISION RECORD**

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment.

CHAPTER 01 · CHALLENGE

# Stress-test the boundary behind “Scope and assurance language”

A deep-dive topic earns an additional page for counterexamples, adversarial conditions, missing dependencies, role failure, recovery, and evidence limits.

The principal guide risk is: The highest-impact path may cross trusted prompts, poisoned sources, compromised dependencies, over-capable tools, human/admin bypass, external target replay, packet disclosure, support access or an update channel even when the core network perimeter is strong.

For this chapter, remove or alter the strongest assumption in “architecture claim, control, test, evidence, alignment, and certification.”. Change the source or version, revoke a person or right, replace a delivered dependency, interrupt the selected deployment, introduce incompatible state, withhold a target acknowledgement, tamper with evidence, or make the relied-on contract condition unavailable. Observe whether the boundary refuses, degrades, escalates, reconciles, recovers, or fails silently.

Run the fault from the perspective of CISO and risk owner, product/application security, security architecture, SOC/incident response, identity and keys, privacy/data, model/tool and source owners, platform operations, evidence/records, supplier and independent assurance. Each owner should see only the authorised information needed to act, understand the current state, and have an explicit safe next step. Restore normal operation, retrieve a pre-change record, and decide whether the chapter proposition still holds, now holds conditionally, requires remediation, or blocks reliance.

| STRESS LANE | INJECTION | EXPECTED EVIDENCE |
|---|---|---|
| **Authority** | Role absent, revoked, conflicted or bypassed | Wait/refuse/escalate + receipt |
| **Dependency** | Source, service, package, model or key unavailable | Degraded state + manifest + recovery |
| **Integrity** | Alter, delete, reorder, substitute | Verification failure + quarantine |
| **Operation** | Load, outage, clock, storage or update fault | Alert + safe state + restore |
| **Target** | Timeout, duplicate, rejection or conflicting state | Idempotency + reconciliation |
| **Exit** | Supplier access removed or contract ends | Export + local verification + continuity result |

Select applicable lanes and add topic-specific consequences before the run.

**STOP RULE**

If a critical assumption fails without a detectable safe state and accountable owner, stop the proof or production lane until the boundary is repaired.

CHAPTER 02 · EXPLAIN

# System assets and actors

data, knowledge, models, tools, identities, policies, keys, artefacts, traces, infrastructure, and administrators.

This chapter owns a specific part of the reader's decision: data, knowledge, models, tools, identities, policies, keys, artefacts, traces, infrastructure, and administrators.

In this guide, security is expressed across the complete work path: identity and purpose, source and context, models and agents, tool capabilities, human gates, target actions, evidence and keys, deployment and management planes, packages and updates, support, incident response, recovery and exit. Applied to "System assets and actors", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Architecture describes intended controls; implementation and configuration realise them; tests and operational evidence show behaviour; independent reports have scope and date; certification or regulatory status must be stated exactly. Open or inspectable code does not remove vulnerability or customer hardening duties. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

**Data**

Define the exact proposition and current authoritative source.

LENS 2

**Knowledge**

Name the responsible person, system, dependency, and version.

LENS 3

**Models**

Specify the evidence and how an independent reviewer checks it.

LENS 4

**Tools**

Record scope, failure, unresolved question, and safe next action.

**OWNED ARTEFACT**

This chapter contributes to a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance.

CHAPTER 02 · APPLY

# Turn “System assets and actors” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, data, knowledge, models, tools, identities, policies, keys, artefacts, traces, infrastructure, and administrators., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant delivered mechanism, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: a security team attacks one representative workflow across identity, source, model/tool, capability, human, target, evidence, managed or local infrastructure, update and recovery while preserving an investigation record. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Run access, injection, isolation, capability, resource, secret, model/tool substitution, target replay, packet tamper, key rotation, signed-update, no-egress, support/admin, incident and clean-room recovery tests; map every finding to owner, risk, remedy, retest and accepted residual state. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

| QUESTION | REQUIRED RECORD | OWNER / STOP |
|---|---|---|
| **What is claimed?** | Exact wording, scope, date, source | Guide decision owner |
| **What exists?** | Delivered manifest, interface, artefact | Technical/product owner |
| **What is configured?** | Customer data, roles, rules, posture | Customer control owner |
| **What happened?** | Raw run, target state, packet, metric | Domain/operations owner |
| **What is assured?** | Test/report scope, findings, limits | Independent challenger |
| **What is committed?** | Offer/agreement, remedy, exit term | Procurement/legal owner |

Evidence ladder for system assets and actors; not every question needs every row, but no row may be implied without evidence.

**DECISION RECORD**

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment.

CHAPTER 03 · EXPLAIN

# Trust-boundary map

user/workspace, API, product services, source systems, model/tool connections, evidence stores, management plane, and external exports.

This chapter owns a specific part of the reader's decision: user/workspace, API, product services, source systems, model/tool connections, evidence stores, management plane, and external exports.

In this guide, security is expressed across the complete work path: identity and purpose, source and context, models and agents, tool capabilities, human gates, target actions, evidence and keys, deployment and management planes, packages and updates, support, incident response, recovery and exit. Applied to "Trust-boundary map", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Architecture describes intended controls; implementation and configuration realise them; tests and operational evidence show behaviour; independent reports have scope and date; certification or regulatory status must be stated exactly. Open or inspectable code does not remove vulnerability or customer hardening duties. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

### User/workspace

Define the exact proposition and current authoritative source.

LENS 2

### API

Name the responsible person, system, dependency, and version.

LENS 3

### Product services

Specify the evidence and how an independent reviewer checks it.

LENS 4

### Source systems

Record scope, failure, unresolved question, and safe next action.

**OWNED ARTEFACT**

This chapter contributes to a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance.

CHAPTER 03 · APPLY

# Turn “Trust-boundary map” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, user/workspace, API, product services, source systems, model/tool connections, evidence stores, management plane, and external exports., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant customer configuration, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: a security team attacks one representative workflow across identity, source, model/tool, capability, human, target, evidence, managed or local infrastructure, update and recovery while preserving an investigation record. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Run access, injection, isolation, capability, resource, secret, model/tool substitution, target replay, packet tamper, key rotation, signed-update, no-egress, support/admin, incident and clean-room recovery tests; map every finding to owner, risk, remedy, retest and accepted residual state. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

| QUESTION | REQUIRED RECORD | OWNER / STOP |
|---|---|---|
| **What is claimed?** | Exact wording, scope, date, source | Guide decision owner |
| **What exists?** | Delivered manifest, interface, artefact | Technical/product owner |
| **What is configured?** | Customer data, roles, rules, posture | Customer control owner |
| **What happened?** | Raw run, target state, packet, metric | Domain/operations owner |
| **What is assured?** | Test/report scope, findings, limits | Independent challenger |
| **What is committed?** | Offer/agreement, remedy, exit term | Procurement/legal owner |

Evidence ladder for trust-boundary map; not every question needs every row, but no row may be implied without evidence.

**DECISION RECORD**

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment.

CHAPTER 03 · CHALLENGE

# Stress-test the boundary behind "Trust-boundary map"

A deep-dive topic earns an additional page for counterexamples, adversarial conditions, missing dependencies, role failure, recovery, and evidence limits.

The principal guide risk is: The highest-impact path may cross trusted prompts, poisoned sources, compromised dependencies, over-capable tools, human/admin bypass, external target replay, packet disclosure, support access or an update channel even when the core network perimeter is strong.

For this chapter, remove or alter the strongest assumption in "user/workspace, API, product services, source systems, model/tool connections, evidence stores, management plane, and external exports.". Change the source or version, revoke a person or right, replace a delivered dependency, interrupt the selected deployment, introduce incompatible state, withhold a target acknowledgement, tamper with evidence, or make the relied-on contract condition unavailable. Observe whether the boundary refuses, degrades, escalates, reconciles, recovers, or fails silently.

Run the fault from the perspective of CISO and risk owner, product/application security, security architecture, SOC/incident response, identity and keys, privacy/data, model/tool and source owners, platform operations, evidence/records, supplier and independent assurance. Each owner should see only the authorised information needed to act, understand the current state, and have an explicit safe next step. Restore normal operation, retrieve a pre-change record, and decide whether the chapter proposition still holds, now holds conditionally, requires remediation, or blocks reliance.

| STRESS LANE | INJECTION | EXPECTED EVIDENCE |
|---|---|---|
| **Authority** | Role absent, revoked, conflicted or bypassed | Wait/refuse/escala te + receipt |
| **Dependency** | Source, service, package, model or key unavailable | Degraded state + manifest + recovery |
| **Integrity** | Alter, delete, reorder, substitute | Verification failure + quarantine |
| **Operation** | Load, outage, clock, storage or update fault | Alert + safe state + restore |
| **Target** | Timeout, duplicate, rejection or conflicting state | Idempotency + reconciliation |
| **Exit** | Supplier access removed or contract ends | Export + local verification + continuity result |

Select applicable lanes and add topic-specific consequences before the run.

**STOP RULE**

If a critical assumption fails without a detectable safe state and accountable owner, stop the proof or production lane until the boundary is repaired.

CHAPTER 04 · EXPLAIN

# Identity and human authority

authentication boundary, roles, least privilege, approvals, overrides, emergency access, and audit.

This chapter owns a specific part of the reader's decision: authentication boundary, roles, least privilege, approvals, overrides, emergency access, and audit.

In this guide, security is expressed across the complete work path: identity and purpose, source and context, models and agents, tool capabilities, human gates, target actions, evidence and keys, deployment and management planes, packages and updates, support, incident response, recovery and exit. Applied to "Identity and human authority", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Architecture describes intended controls; implementation and configuration realise them; tests and operational evidence show behaviour; independent reports have scope and date; certification or regulatory status must be stated exactly. Open or inspectable code does not remove vulnerability or customer hardening duties. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

**Authentication boundary**

Define the exact proposition and current authoritative source.

LENS 2

**Roles**

Name the responsible person, system, dependency, and version.

LENS 3

**Least privilege**

Specify the evidence and how an independent reviewer checks it.

LENS 4

**Approvals**

Record scope, failure, unresolved question, and safe next action.

OWNED ARTEFACT

This chapter contributes to a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance.

CHAPTER 04 · APPLY

# Turn “Identity and human authority” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, authentication boundary, roles, least privilege, approvals, overrides, emergency access, and audit., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant observed operation, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: a security team attacks one representative workflow across identity, source, model/tool, capability, human, target, evidence, managed or local infrastructure, update and recovery while preserving an investigation record. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Run access, injection, isolation, capability, resource, secret, model/tool substitution, target replay, packet tamper, key rotation, signed-update, no-egress, support/admin, incident and clean-room recovery tests; map every finding to owner, risk, remedy, retest and accepted residual state. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

| QUESTION | REQUIRED RECORD | OWNER / STOP |
|---|---|---|
| **What is claimed?** | Exact wording, scope, date, source | Guide decision owner |
| **What exists?** | Delivered manifest, interface, artefact | Technical/product owner |
| **What is configured?** | Customer data, roles, rules, posture | Customer control owner |
| **What happened?** | Raw run, target state, packet, metric | Domain/operations owner |
| **What is assured?** | Test/report scope, findings, limits | Independent challenger |
| **What is committed?** | Offer/agreement, remedy, exit term | Procurement/legal owner |

Evidence ladder for identity and human authority; not every question needs every row, but no row may be implied without evidence.

**DECISION RECORD**

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment.

CHAPTER 05 · EXPLAIN

# Tool and action security

permission matrices, policy checkpoints, guardian/output filters, sandboxing, proposed actions, and execution records.

This chapter owns a specific part of the reader's decision: permission matrices, policy checkpoints, guardian/output filters, sandboxing, proposed actions, and execution records.

In this guide, security is expressed across the complete work path: identity and purpose, source and context, models and agents, tool capabilities, human gates, target actions, evidence and keys, deployment and management planes, packages and updates, support, incident response, recovery and exit. Applied to "Tool and action security", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Architecture describes intended controls; implementation and configuration realise them; tests and operational evidence show behaviour; independent reports have scope and date; certification or regulatory status must be stated exactly. Open or inspectable code does not remove vulnerability or customer hardening duties. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

**Permission matrices**

Define the exact proposition and current authoritative source.

LENS 2

**Policy checkpoints**

Name the responsible person, system, dependency, and version.

LENS 3

**Guardian/output filters**

Specify the evidence and how an independent reviewer checks it.

LENS 4

**Sandboxing**

Record scope, failure, unresolved question, and safe next action.

**OWNED ARTEFACT**

This chapter contributes to a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance.

CHAPTER 05 · APPLY

# Turn “Tool and action security” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, permission matrices, policy checkpoints, guardian/output filters, sandboxing, proposed actions, and execution records., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant contract and assurance, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: a security team attacks one representative workflow across identity, source, model/tool, capability, human, target, evidence, managed or local infrastructure, update and recovery while preserving an investigation record. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Run access, injection, isolation, capability, resource, secret, model/tool substitution, target replay, packet tamper, key rotation, signed-update, no-egress, support/admin, incident and clean-room recovery tests; map every finding to owner, risk, remedy, retest and accepted residual state. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

| QUESTION | REQUIRED RECORD | OWNER / STOP |
|---|---|---|
| **What is claimed?** | Exact wording, scope, date, source | Guide decision owner |
| **What exists?** | Delivered manifest, interface, artefact | Technical/product owner |
| **What is configured?** | Customer data, roles, rules, posture | Customer control owner |
| **What happened?** | Raw run, target state, packet, metric | Domain/operations owner |
| **What is assured?** | Test/report scope, findings, limits | Independent challenger |
| **What is committed?** | Offer/agreement, remedy, exit term | Procurement/legal owner |

Evidence ladder for tool and action security; not every question needs every row, but no row may be implied without evidence.

**DECISION RECORD**

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment.

CHAPTER 05 · CHALLENGE

# Stress-test the boundary behind "Tool and action security"

A deep-dive topic earns an additional page for counterexamples, adversarial conditions, missing dependencies, role failure, recovery, and evidence limits.

The principal guide risk is: The highest-impact path may cross trusted prompts, poisoned sources, compromised dependencies, over-capable tools, human/admin bypass, external target replay, packet disclosure, support access or an update channel even when the core network perimeter is strong.

For this chapter, remove or alter the strongest assumption in "permission matrices, policy checkpoints, guardian/output filters, sandboxing, proposed actions, and execution records.". Change the source or version, revoke a person or right, replace a delivered dependency, interrupt the selected deployment, introduce incompatible state, withhold a target acknowledgement, tamper with evidence, or make the relied-on contract condition unavailable. Observe whether the boundary refuses, degrades, escalates, reconciles, recovers, or fails silently.

Run the fault from the perspective of CISO and risk owner, product/application security, security architecture, SOC/incident response, identity and keys, privacy/data, model/tool and source owners, platform operations, evidence/records, supplier and independent assurance. Each owner should see only the authorised information needed to act, understand the current state, and have an explicit safe next step. Restore normal operation, retrieve a pre-change record, and decide whether the chapter proposition still holds, now holds conditionally, requires remediation, or blocks reliance.

| STRESS LANE | INJECTION | EXPECTED EVIDENCE |
|---|---|---|
| **Authority** | Role absent, revoked, conflicted or bypassed | Wait/refuse/escala te + receipt |
| **Dependency** | Source, service, package, model or key unavailable | Degraded state + manifest + recovery |
| **Integrity** | Alter, delete, reorder, substitute | Verification failure + quarantine |
| **Operation** | Load, outage, clock, storage or update fault | Alert + safe state + restore |
| **Target** | Timeout, duplicate, rejection or conflicting state | Idempotency + reconciliation |
| **Exit** | Supplier access removed or contract ends | Export + local verification + continuity result |

Select applicable lanes and add topic-specific consequences before the run.

**STOP RULE**

If a critical assumption fails without a detectable safe state and accountable owner, stop the proof or production lane until the boundary is repaired.

CHAPTER 06 · EXPLAIN

# Model and agent security

selection, isolation, prompt/context boundaries, model swapping, agent lifecycle, task identity, refusal, and recovery.

This chapter owns a specific part of the reader's decision: selection, isolation, prompt/context boundaries, model swapping, agent lifecycle, task identity, refusal, and recovery.

In this guide, security is expressed across the complete work path: identity and purpose, source and context, models and agents, tool capabilities, human gates, target actions, evidence and keys, deployment and management planes, packages and updates, support, incident response, recovery and exit. Applied to "Model and agent security", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Architecture describes intended controls; implementation and configuration realise them; tests and operational evidence show behaviour; independent reports have scope and date; certification or regulatory status must be stated exactly. Open or inspectable code does not remove vulnerability or customer hardening duties. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

**Selection**

Define the exact proposition and current authoritative source.

LENS 2

**Isolation**

Name the responsible person, system, dependency, and version.

LENS 3

**Prompt/context boundaries**

Specify the evidence and how an independent reviewer checks it.

LENS 4

**Model swapping**

Record scope, failure, unresolved question, and safe next action.

**OWNED ARTEFACT**

This chapter contributes to a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance.

CHAPTER 06 · APPLY

# Turn “Model and agent security” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, selection, isolation, prompt/context boundaries, model swapping, agent lifecycle, task identity, refusal, and recovery., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant source and version, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: a security team attacks one representative workflow across identity, source, model/tool, capability, human, target, evidence, managed or local infrastructure, update and recovery while preserving an investigation record. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Run access, injection, isolation, capability, resource, secret, model/tool substitution, target replay, packet tamper, key rotation, signed-update, no-egress, support/admin, incident and clean-room recovery tests; map every finding to owner, risk, remedy, retest and accepted residual state. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

| QUESTION | REQUIRED RECORD | OWNER / STOP |
|---|---|---|
| **What is claimed?** | Exact wording, scope, date, source | Guide decision owner |
| **What exists?** | Delivered manifest, interface, artefact | Technical/product owner |
| **What is configured?** | Customer data, roles, rules, posture | Customer control owner |
| **What happened?** | Raw run, target state, packet, metric | Domain/operations owner |
| **What is assured?** | Test/report scope, findings, limits | Independent challenger |
| **What is committed?** | Offer/agreement, remedy, exit term | Procurement/legal owner |

Evidence ladder for model and agent security; not every question needs every row, but no row may be implied without evidence.

**DECISION RECORD**

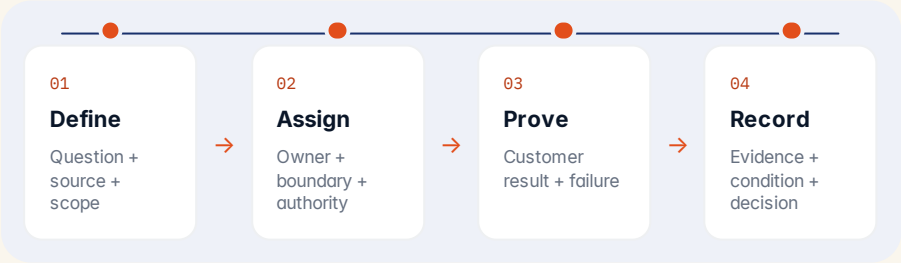
Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment.

02

SECURITY ARCHITECTURE AND ASSURANCE

# Define the mechanism, controls, and evidence

Turn architecture and policy language into versions, owners, boundaries, tests, records, failures, and explicit limitations.




The same evidence discipline applies to every chapter in the section.

CHAPTER 07 · EXPLAIN

# Data and privacy protection

residency, encryption boundary, key ownership, supported secret masking, access, retention, and erasure considerations.

This chapter owns a specific part of the reader's decision: residency, encryption boundary, key ownership, supported secret masking, access, retention, and erasure considerations.

In this guide, security is expressed across the complete work path: identity and purpose, source and context, models and agents, tool capabilities, human gates, target actions, evidence and keys, deployment and management planes, packages and updates, support, incident response, recovery and exit. Applied to "Data and privacy protection", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Architecture describes intended controls; implementation and configuration realise them; tests and operational evidence show behaviour; independent reports have scope and date; certification or regulatory status must be stated exactly. Open or inspectable code does not remove vulnerability or customer hardening duties. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

### Residency

Define the exact proposition and current authoritative source.

LENS 2

### Encryption boundary

Name the responsible person, system, dependency, and version.

LENS 3

### Key ownership

Specify the evidence and how an independent reviewer checks it.

LENS 4

### Supported secret masking

Record scope, failure, unresolved question, and safe next action.

**OWNED ARTEFACT**

This chapter contributes to a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance.

CHAPTER 07 · APPLY

# Turn “Data and privacy protection” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, residency, encryption boundary, key ownership, supported secret masking, access, retention, and erasure considerations., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant delivered mechanism, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: a security team attacks one representative workflow across identity, source, model/tool, capability, human, target, evidence, managed or local infrastructure, update and recovery while preserving an investigation record. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Run access, injection, isolation, capability, resource, secret, model/tool substitution, target replay, packet tamper, key rotation, signed-update, no-egress, support/admin, incident and clean-room recovery tests; map every finding to owner, risk, remedy, retest and accepted residual state. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

| QUESTION | REQUIRED RECORD | OWNER / STOP |
|---|---|---|
| **What is claimed?** | Exact wording, scope, date, source | Guide decision owner |
| **What exists?** | Delivered manifest, interface, artefact | Technical/product owner |
| **What is configured?** | Customer data, roles, rules, posture | Customer control owner |
| **What happened?** | Raw run, target state, packet, metric | Domain/operations owner |
| **What is assured?** | Test/report scope, findings, limits | Independent challenger |
| **What is committed?** | Offer/agreement, remedy, exit term | Procurement/legal owner |

Evidence ladder for data and privacy protection; not every question needs every row, but no row may be implied without evidence.

**DECISION RECORD**

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment.

CHAPTER 07 · CHALLENGE

# Stress-test the boundary behind "Data and privacy protection"

A deep-dive topic earns an additional page for counterexamples, adversarial conditions, missing dependencies, role failure, recovery, and evidence limits.

The principal guide risk is: The highest-impact path may cross trusted prompts, poisoned sources, compromised dependencies, over-capable tools, human/admin bypass, external target replay, packet disclosure, support access or an update channel even when the core network perimeter is strong.

For this chapter, remove or alter the strongest assumption in "residency, encryption boundary, key ownership, supported secret masking, access, retention, and erasure considerations.". Change the source or version, revoke a person or right, replace a delivered dependency, interrupt the selected deployment, introduce incompatible state, withhold a target acknowledgement, tamper with evidence, or make the relied-on contract condition unavailable. Observe whether the boundary refuses, degrades, escalates, reconciles, recovers, or fails silently.

Run the fault from the perspective of CISO and risk owner, product/application security, security architecture, SOC/incident response, identity and keys, privacy/data, model/tool and source owners, platform operations, evidence/records, supplier and independent assurance. Each owner should see only the authorised information needed to act, understand the current state, and have an explicit safe next step. Restore normal operation, retrieve a pre-change record, and decide whether the chapter proposition still holds, now holds conditionally, requires remediation, or blocks reliance.

| STRESS LANE | INJECTION | EXPECTED EVIDENCE |
|---|---|---|
| **Authority** | Role absent, revoked, conflicted or bypassed | Wait/refuse/escala te + receipt |
| **Dependency** | Source, service, package, model or key unavailable | Degraded state + manifest + recovery |
| **Integrity** | Alter, delete, reorder, substitute | Verification failure + quarantine |
| **Operation** | Load, outage, clock, storage or update fault | Alert + safe state + restore |
| **Target** | Timeout, duplicate, rejection or conflicting state | Idempotency + reconciliation |
| **Exit** | Supplier access removed or contract ends | Export + local verification + continuity result |

Select applicable lanes and add topic-specific consequences before the run.

**STOP RULE**

If a critical assumption fails without a detectable safe state and accountable owner, stop the proof or production lane until the boundary is repaired.

CHAPTER 08 · EXPLAIN

# Execution containment

capability-limited runtimes, WASM where supported, network/file boundaries, and sealed transcripts.

This chapter owns a specific part of the reader's decision: capability-limited runtimes, WASM where supported, network/file boundaries, and sealed transcripts.

In this guide, security is expressed across the complete work path: identity and purpose, source and context, models and agents, tool capabilities, human gates, target actions, evidence and keys, deployment and management planes, packages and updates, support, incident response, recovery and exit. Applied to "Execution containment", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Architecture describes intended controls; implementation and configuration realise them; tests and operational evidence show behaviour; independent reports have scope and date; certification or regulatory status must be stated exactly. Open or inspectable code does not remove vulnerability or customer hardening duties. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

**Capability-limited runtimes**

Define the exact proposition and current authoritative source.

LENS 2

**WASM where supported**

Name the responsible person, system, dependency, and version.

LENS 3

**Network/file boundaries**

Specify the evidence and how an independent reviewer checks it.

LENS 4

**And sealed transcripts**

Record scope, failure, unresolved question, and safe next action.

OWNED ARTEFACT

This chapter contributes to a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance.

CHAPTER 08 · APPLY

# Turn “Execution containment” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, capability-limited runtimes, WASM where supported, network/file boundaries, and sealed transcripts., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant customer configuration, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: a security team attacks one representative workflow across identity, source, model/tool, capability, human, target, evidence, managed or local infrastructure, update and recovery while preserving an investigation record. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Run access, injection, isolation, capability, resource, secret, model/tool substitution, target replay, packet tamper, key rotation, signed-update, no-egress, support/admin, incident and clean-room recovery tests; map every finding to owner, risk, remedy, retest and accepted residual state. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

| QUESTION | REQUIRED RECORD | OWNER / STOP |
|---|---|---|
| **What is claimed?** | Exact wording, scope, date, source | Guide decision owner |
| **What exists?** | Delivered manifest, interface, artefact | Technical/product owner |
| **What is configured?** | Customer data, roles, rules, posture | Customer control owner |
| **What happened?** | Raw run, target state, packet, metric | Domain/operations owner |
| **What is assured?** | Test/report scope, findings, limits | Independent challenger |
| **What is committed?** | Offer/agreement, remedy, exit term | Procurement/legal owner |

Evidence ladder for execution containment; not every question needs every row, but no row may be implied without evidence.

**DECISION RECORD**

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment.

CHAPTER 09 · EXPLAIN

# Integrity and non-repudiation

hashes, Merkle structures, signatures, event chains, artefact identity, and verifier independence.

This chapter owns a specific part of the reader's decision: hashes, Merkle structures, signatures, event chains, artefact identity, and verifier independence.

In this guide, security is expressed across the complete work path: identity and purpose, source and context, models and agents, tool capabilities, human gates, target actions, evidence and keys, deployment and management planes, packages and updates, support, incident response, recovery and exit. Applied to "Integrity and non-repudiation", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Architecture describes intended controls; implementation and configuration realise them; tests and operational evidence show behaviour; independent reports have scope and date; certification or regulatory status must be stated exactly. Open or inspectable code does not remove vulnerability or customer hardening duties. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

### Hashes

Define the exact proposition and current authoritative source.

LENS 2

### Merkle structures

Name the responsible person, system, dependency, and version.

LENS 3

### Signatures

Specify the evidence and how an independent reviewer checks it.

LENS 4

### Event chains

Record scope, failure, unresolved question, and safe next action.

**OWNED ARTEFACT**

This chapter contributes to a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance.

CHAPTER 09 · APPLY

# Turn “Integrity and non-repudiation” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, hashes, Merkle structures, signatures, event chains, artefact identity, and verifier independence., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant observed operation, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: a security team attacks one representative workflow across identity, source, model/tool, capability, human, target, evidence, managed or local infrastructure, update and recovery while preserving an investigation record. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Run access, injection, isolation, capability, resource, secret, model/tool substitution, target replay, packet tamper, key rotation, signed-update, no-egress, support/admin, incident and clean-room recovery tests; map every finding to owner, risk, remedy, retest and accepted residual state. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

| QUESTION | REQUIRED RECORD | OWNER / STOP |
|---|---|---|
| **What is claimed?** | Exact wording, scope, date, source | Guide decision owner |
| **What exists?** | Delivered manifest, interface, artefact | Technical/product owner |
| **What is configured?** | Customer data, roles, rules, posture | Customer control owner |
| **What happened?** | Raw run, target state, packet, metric | Domain/operations owner |
| **What is assured?** | Test/report scope, findings, limits | Independent challenger |
| **What is committed?** | Offer/agreement, remedy, exit term | Procurement/legal owner |

Evidence ladder for integrity and non-repudiation; not every question needs every row, but no row may be implied without evidence.

**DECISION RECORD**

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment.

CHAPTER 09 · CHALLENGE

# Stress-test the boundary behind “Integrity and non-repudiation”

A deep-dive topic earns an additional page for counterexamples, adversarial conditions, missing dependencies, role failure, recovery, and evidence limits.

The principal guide risk is: The highest-impact path may cross trusted prompts, poisoned sources, compromised dependencies, over-capable tools, human/admin bypass, external target replay, packet disclosure, support access or an update channel even when the core network perimeter is strong.

For this chapter, remove or alter the strongest assumption in “hashes, Merkle structures, signatures, event chains, artefact identity, and verifier independence.”. Change the source or version, revoke a person or right, replace a delivered dependency, interrupt the selected deployment, introduce incompatible state, withhold a target acknowledgement, tamper with evidence, or make the relied-on contract condition unavailable. Observe whether the boundary refuses, degrades, escalates, reconciles, recovers, or fails silently.

Run the fault from the perspective of CISO and risk owner, product/application security, security architecture, SOC/incident response, identity and keys, privacy/data, model/tool and source owners, platform operations, evidence/records, supplier and independent assurance. Each owner should see only the authorised information needed to act, understand the current state, and have an explicit safe next step. Restore normal operation, retrieve a pre-change record, and decide whether the chapter proposition still holds, now holds conditionally, requires remediation, or blocks reliance.

| STRESS LANE | INJECTION | EXPECTED EVIDENCE |
|---|---|---|
| **Authority** | Role absent, revoked, conflicted or bypassed | Wait/refuse/escalate + receipt |
| **Dependency** | Source, service, package, model or key unavailable | Degraded state + manifest + recovery |
| **Integrity** | Alter, delete, reorder, substitute | Verification failure + quarantine |
| **Operation** | Load, outage, clock, storage or update fault | Alert + safe state + restore |
| **Target** | Timeout, duplicate, rejection or conflicting state | Idempotency + reconciliation |
| **Exit** | Supplier access removed or contract ends | Export + local verification + continuity result |

Select applicable lanes and add topic-specific consequences before the run.

**STOP RULE**

If a critical assumption fails without a detectable safe state and accountable owner, stop the proof or production lane until the boundary is repaired.

CHAPTER 10 · EXPLAIN

# Supply-chain security

verify current source and release status, signing and delivery-media controls, dependencies, build provenance, and any SBOM or attestation evidence claimed for the selected delivery.

This chapter owns a specific part of the reader's decision: verify current source and release status, signing and delivery-media controls, dependencies, build provenance, and any SBOM or attestation evidence claimed for the selected delivery.

In this guide, security is expressed across the complete work path: identity and purpose, source and context, models and agents, tool capabilities, human gates, target actions, evidence and keys, deployment and management planes, packages and updates, support, incident response, recovery and exit. Applied to "Supply-chain security", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Architecture describes intended controls; implementation and configuration realise them; tests and operational evidence show behaviour; independent reports have scope and date; certification or regulatory status must be stated exactly. Open or inspectable code does not remove vulnerability or customer hardening duties. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

### Verify current source

Define the exact proposition and current authoritative source.

LENS 2

### Release status

Name the responsible person, system, dependency, and version.

LENS 3

### Signing

Specify the evidence and how an independent reviewer checks it.

LENS 4

### Delivery-media controls

Record scope, failure, unresolved question, and safe next action.

**OWNED ARTEFACT**

This chapter contributes to a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance.

CHAPTER 10 · APPLY

# Turn “Supply-chain security” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, verify current source and release status, signing and delivery-media controls, dependencies, build provenance, and any SBOM or attestation evidence claimed for the selected delivery., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant contract and assurance, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: a security team attacks one representative workflow across identity, source, model/tool, capability, human, target, evidence, managed or local infrastructure, update and recovery while preserving an investigation record. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Run access, injection, isolation, capability, resource, secret, model/tool substitution, target replay, packet tamper, key rotation, signed-update, no-egress, support/admin, incident and clean-room recovery tests; map every finding to owner, risk, remedy, retest and accepted residual state. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

| QUESTION | REQUIRED RECORD | OWNER / STOP |
|---|---|---|
| **What is claimed?** | Exact wording, scope, date, source | Guide decision owner |
| **What exists?** | Delivered manifest, interface, artefact | Technical/product owner |
| **What is configured?** | Customer data, roles, rules, posture | Customer control owner |
| **What happened?** | Raw run, target state, packet, metric | Domain/operations owner |
| **What is assured?** | Test/report scope, findings, limits | Independent challenger |
| **What is committed?** | Offer/agreement, remedy, exit term | Procurement/legal owner |

Evidence ladder for supply-chain security; not every question needs every row, but no row may be implied without evidence.

**DECISION RECORD**

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment.

CHAPTER 11 · EXPLAIN

# Detection and response

threat/incident records, investigation chronology, replay, affected artefacts, and recovery evidence.

This chapter owns a specific part of the reader's decision: threat/incident records, investigation chronology, replay, affected artefacts, and recovery evidence.

In this guide, security is expressed across the complete work path: identity and purpose, source and context, models and agents, tool capabilities, human gates, target actions, evidence and keys, deployment and management planes, packages and updates, support, incident response, recovery and exit. Applied to "Detection and response", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Architecture describes intended controls; implementation and configuration realise them; tests and operational evidence show behaviour; independent reports have scope and date; certification or regulatory status must be stated exactly. Open or inspectable code does not remove vulnerability or customer hardening duties. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

**Threat/incident records**

Define the exact proposition and current authoritative source.

LENS 2

**Investigation chronology**

Name the responsible person, system, dependency, and version.

LENS 3

**Replay**

Specify the evidence and how an independent reviewer checks it.

LENS 4

**Affected artefacts**

Record scope, failure, unresolved question, and safe next action.

OWNED ARTEFACT

This chapter contributes to a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance.

CHAPTER 11 · APPLY

# Turn “Detection and response” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, threat/incident records, investigation chronology, replay, affected artefacts, and recovery evidence., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant source and version, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: a security team attacks one representative workflow across identity, source, model/tool, capability, human, target, evidence, managed or local infrastructure, update and recovery while preserving an investigation record. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Run access, injection, isolation, capability, resource, secret, model/tool substitution, target replay, packet tamper, key rotation, signed-update, no-egress, support/admin, incident and clean-room recovery tests; map every finding to owner, risk, remedy, retest and accepted residual state. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

| QUESTION | REQUIRED RECORD | OWNER / STOP |
|---|---|---|
| **What is claimed?** | Exact wording, scope, date, source | Guide decision owner |
| **What exists?** | Delivered manifest, interface, artefact | Technical/product owner |
| **What is configured?** | Customer data, roles, rules, posture | Customer control owner |
| **What happened?** | Raw run, target state, packet, metric | Domain/operations owner |
| **What is assured?** | Test/report scope, findings, limits | Independent challenger |
| **What is committed?** | Offer/agreement, remedy, exit term | Procurement/legal owner |

Evidence ladder for detection and response; not every question needs every row, but no row may be implied without evidence.

**DECISION RECORD**

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment.

CHAPTER 11 · CHALLENGE

# Stress-test the boundary behind "Detection and response"

A deep-dive topic earns an additional page for counterexamples, adversarial conditions, missing dependencies, role failure, recovery, and evidence limits.

The principal guide risk is: The highest-impact path may cross trusted prompts, poisoned sources, compromised dependencies, over-capable tools, human/admin bypass, external target replay, packet disclosure, support access or an update channel even when the core network perimeter is strong.

For this chapter, remove or alter the strongest assumption in "threat/incident records, investigation chronology, replay, affected artefacts, and recovery evidence.". Change the source or version, revoke a person or right, replace a delivered dependency, interrupt the selected deployment, introduce incompatible state, withhold a target acknowledgement, tamper with evidence, or make the relied-on contract condition unavailable. Observe whether the boundary refuses, degrades, escalates, reconciles, recovers, or fails silently.

Run the fault from the perspective of CISO and risk owner, product/application security, security architecture, SOC/incident response, identity and keys, privacy/data, model/tool and source owners, platform operations, evidence/records, supplier and independent assurance. Each owner should see only the authorised information needed to act, understand the current state, and have an explicit safe next step. Restore normal operation, retrieve a pre-change record, and decide whether the chapter proposition still holds, now holds conditionally, requires remediation, or blocks reliance.

| STRESS LANE | INJECTION | EXPECTED EVIDENCE |
|---|---|---|
| **Authority** | Role absent, revoked, conflicted or bypassed | Wait/refuse/escalate + receipt |
| **Dependency** | Source, service, package, model or key unavailable | Degraded state + manifest + recovery |
| **Integrity** | Alter, delete, reorder, substitute | Verification failure + quarantine |
| **Operation** | Load, outage, clock, storage or update fault | Alert + safe state + restore |
| **Target** | Timeout, duplicate, rejection or conflicting state | Idempotency + reconciliation |
| **Exit** | Supplier access removed or contract ends | Export + local verification + continuity result |

Select applicable lanes and add topic-specific consequences before the run.

**STOP RULE**

If a critical assumption fails without a detectable safe state and accountable owner, stop the proof or production lane until the boundary is repaired.

CHAPTER 12 · EXPLAIN

# Deployment-specific threat models

managed Fabric, licensed customer operation, and air-gapped risks and responsibility transfer.

This chapter owns a specific part of the reader's decision: managed Fabric, licensed customer operation, and air-gapped risks and responsibility transfer.

In this guide, security is expressed across the complete work path: identity and purpose, source and context, models and agents, tool capabilities, human gates, target actions, evidence and keys, deployment and management planes, packages and updates, support, incident response, recovery and exit. Applied to "Deployment-specific threat models", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Architecture describes intended controls; implementation and configuration realise them; tests and operational evidence show behaviour; independent reports have scope and date; certification or regulatory status must be stated exactly. Open or inspectable code does not remove vulnerability or customer hardening duties. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

**Managed Fabric**

Define the exact proposition and current authoritative source.

LENS 2

**Licensed customer operation**

Name the responsible person, system, dependency, and version.

LENS 3

**And air-gapped risks**

Specify the evidence and how an independent reviewer checks it.

LENS 4

**Responsibility transfer**

Record scope, failure, unresolved question, and safe next action.

**OWNED ARTEFACT**

This chapter contributes to a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance.

CHAPTER 12 · APPLY

# Turn “Deployment-specific threat models” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, managed Fabric, licensed customer operation, and air-gapped risks and responsibility transfer., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant delivered mechanism, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: a security team attacks one representative workflow across identity, source, model/tool, capability, human, target, evidence, managed or local infrastructure, update and recovery while preserving an investigation record. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Run access, injection, isolation, capability, resource, secret, model/tool substitution, target replay, packet tamper, key rotation, signed-update, no-egress, support/admin, incident and clean-room recovery tests; map every finding to owner, risk, remedy, retest and accepted residual state. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

| QUESTION | REQUIRED RECORD | OWNER / STOP |
|---|---|---|
| **What is claimed?** | Exact wording, scope, date, source | Guide decision owner |
| **What exists?** | Delivered manifest, interface, artefact | Technical/product owner |
| **What is configured?** | Customer data, roles, rules, posture | Customer control owner |
| **What happened?** | Raw run, target state, packet, metric | Domain/operations owner |
| **What is assured?** | Test/report scope, findings, limits | Independent challenger |
| **What is committed?** | Offer/agreement, remedy, exit term | Procurement/legal owner |

Evidence ladder for deployment-specific threat models; not every question needs every row, but no row may be implied without evidence.

**DECISION RECORD**

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment.

03

SECURITY ARCHITECTURE AND ASSURANCE

# Prove adoption, operation, continuity, and exit

Use customer evidence, challenge, recovery, diligence, commercial terms, and a signed decision before expanding reliance.


The same evidence discipline applies to every chapter in the section.

CHAPTER 13 · EXPLAIN

# Availability and continuity

link loss, external-service loss, failover, offline operation, backups, restore, updates, and licence continuity.

This chapter owns a specific part of the reader's decision: link loss, external-service loss, failover, offline operation, backups, restore, updates, and licence continuity.

In this guide, security is expressed across the complete work path: identity and purpose, source and context, models and agents, tool capabilities, human gates, target actions, evidence and keys, deployment and management planes, packages and updates, support, incident response, recovery and exit. Applied to "Availability and continuity", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Architecture describes intended controls; implementation and configuration realise them; tests and operational evidence show behaviour; independent reports have scope and date; certification or regulatory status must be stated exactly. Open or inspectable code does not remove vulnerability or customer hardening duties. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

### Link loss

Define the exact proposition and current authoritative source.

LENS 2

### External-service loss

Name the responsible person, system, dependency, and version.

LENS 3

### Failover

Specify the evidence and how an independent reviewer checks it.

LENS 4

### Offline operation

Record scope, failure, unresolved question, and safe next action.

**OWNED ARTEFACT**

This chapter contributes to a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance.

CHAPTER 13 · APPLY

# Turn “Availability and continuity” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, link loss, external-service loss, failover, offline operation, backups, restore, updates, and licence continuity., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant customer configuration, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: a security team attacks one representative workflow across identity, source, model/tool, capability, human, target, evidence, managed or local infrastructure, update and recovery while preserving an investigation record. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Run access, injection, isolation, capability, resource, secret, model/tool substitution, target replay, packet tamper, key rotation, signed-update, no-egress, support/admin, incident and clean-room recovery tests; map every finding to owner, risk, remedy, retest and accepted residual state. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

| QUESTION | REQUIRED RECORD | OWNER / STOP |
|---|---|---|
| **What is claimed?** | Exact wording, scope, date, source | Guide decision owner |
| **What exists?** | Delivered manifest, interface, artefact | Technical/product owner |
| **What is configured?** | Customer data, roles, rules, posture | Customer control owner |
| **What happened?** | Raw run, target state, packet, metric | Domain/operations owner |
| **What is assured?** | Test/report scope, findings, limits | Independent challenger |
| **What is committed?** | Offer/agreement, remedy, exit term | Procurement/legal owner |

Evidence ladder for availability and continuity; not every question needs every row, but no row may be implied without evidence.

**DECISION RECORD**

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment.

CHAPTER 13 · CHALLENGE

# Stress-test the boundary behind "Availability and continuity"

A deep-dive topic earns an additional page for counterexamples, adversarial conditions, missing dependencies, role failure, recovery, and evidence limits.

The principal guide risk is: The highest-impact path may cross trusted prompts, poisoned sources, compromised dependencies, over-capable tools, human/admin bypass, external target replay, packet disclosure, support access or an update channel even when the core network perimeter is strong.

For this chapter, remove or alter the strongest assumption in "link loss, external-service loss, failover, offline operation, backups, restore, updates, and licence continuity.". Change the source or version, revoke a person or right, replace a delivered dependency, interrupt the selected deployment, introduce incompatible state, withhold a target acknowledgement, tamper with evidence, or make the relied-on contract condition unavailable. Observe whether the boundary refuses, degrades, escalates, reconciles, recovers, or fails silently.

Run the fault from the perspective of CISO and risk owner, product/application security, security architecture, SOC/incident response, identity and keys, privacy/data, model/tool and source owners, platform operations, evidence/records, supplier and independent assurance. Each owner should see only the authorised information needed to act, understand the current state, and have an explicit safe next step. Restore normal operation, retrieve a pre-change record, and decide whether the chapter proposition still holds, now holds conditionally, requires remediation, or blocks reliance.

| STRESS LANE | INJECTION | EXPECTED EVIDENCE |
|---|---|---|
| **Authority** | Role absent, revoked, conflicted or bypassed | Wait/refuse/escala te + receipt |
| **Dependency** | Source, service, package, model or key unavailable | Degraded state + manifest + recovery |
| **Integrity** | Alter, delete, reorder, substitute | Verification failure + quarantine |
| **Operation** | Load, outage, clock, storage or update fault | Alert + safe state + restore |
| **Target** | Timeout, duplicate, rejection or conflicting state | Idempotency + reconciliation |
| **Exit** | Supplier access removed or contract ends | Export + local verification + continuity result |

Select applicable lanes and add topic-specific consequences before the run.

**STOP RULE**

If a critical assumption fails without a detectable safe state and accountable owner, stop the proof or production lane until the boundary is repaired.

CHAPTER 14 · EXPLAIN

# Cryptography and future-facing claims

explain current website statements precisely and avoid implying unclaimed certification or universal quantum safety.

This chapter owns a specific part of the reader's decision: explain current website statements precisely and avoid implying unclaimed certification or universal quantum safety.

In this guide, security is expressed across the complete work path: identity and purpose, source and context, models and agents, tool capabilities, human gates, target actions, evidence and keys, deployment and management planes, packages and updates, support, incident response, recovery and exit. Applied to "Cryptography and future-facing claims", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Architecture describes intended controls; implementation and configuration realise them; tests and operational evidence show behaviour; independent reports have scope and date; certification or regulatory status must be stated exactly. Open or inspectable code does not remove vulnerability or customer hardening duties. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

**Current website statements precisely**

Define the exact proposition and current authoritative source.

LENS 2

**Avoid implying unclaimed certification or universal quantum safety**

Name the responsible person, system, dependency, and version.

LENS 3

**Scope and limits**

Specify the evidence and how an independent reviewer checks it.

LENS 4

**Owner and evidence**

Record scope, failure, unresolved question, and safe next action.

**OWNED ARTEFACT**

This chapter contributes to a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance.

CHAPTER 14 · APPLY

# Turn “Cryptography and future-facing claims” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, explain current website statements precisely and avoid implying unclaimed certification or universal quantum safety., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant observed operation, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: a security team attacks one representative workflow across identity, source, model/tool, capability, human, target, evidence, managed or local infrastructure, update and recovery while preserving an investigation record. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Run access, injection, isolation, capability, resource, secret, model/tool substitution, target replay, packet tamper, key rotation, signed-update, no-egress, support/admin, incident and clean-room recovery tests; map every finding to owner, risk, remedy, retest and accepted residual state. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

| QUESTION | REQUIRED RECORD | OWNER / STOP |
|---|---|---|
| **What is claimed?** | Exact wording, scope, date, source | Guide decision owner |
| **What exists?** | Delivered manifest, interface, artefact | Technical/product owner |
| **What is configured?** | Customer data, roles, rules, posture | Customer control owner |
| **What happened?** | Raw run, target state, packet, metric | Domain/operations owner |
| **What is assured?** | Test/report scope, findings, limits | Independent challenger |
| **What is committed?** | Offer/agreement, remedy, exit term | Procurement/legal owner |

Evidence ladder for cryptography and future-facing claims; not every question needs every row, but no row may be implied without evidence.

**DECISION RECORD**

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment.

CHAPTER 15 · EXPLAIN

# Assurance evidence package

architecture, data-flow, keys, controls, tests, update path, incident process, and responsibility matrix to request.

This chapter owns a specific part of the reader's decision: architecture, data-flow, keys, controls, tests, update path, incident process, and responsibility matrix to request.

In this guide, security is expressed across the complete work path: identity and purpose, source and context, models and agents, tool capabilities, human gates, target actions, evidence and keys, deployment and management planes, packages and updates, support, incident response, recovery and exit. Applied to "Assurance evidence package", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Architecture describes intended controls; implementation and configuration realise them; tests and operational evidence show behaviour; independent reports have scope and date; certification or regulatory status must be stated exactly. Open or inspectable code does not remove vulnerability or customer hardening duties. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

**Architecture**

Define the exact proposition and current authoritative source.

LENS 2

**Data-flow**

Name the responsible person, system, dependency, and version.

LENS 3

**Keys**

Specify the evidence and how an independent reviewer checks it.

LENS 4

**Controls**

Record scope, failure, unresolved question, and safe next action.

**OWNED ARTEFACT**

This chapter contributes to a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance.

CHAPTER 15 · APPLY

# Turn "Assurance evidence package" into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, architecture, data-flow, keys, controls, tests, update path, incident process, and responsibility matrix to request., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant contract and assurance, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: a security team attacks one representative workflow across identity, source, model/tool, capability, human, target, evidence, managed or local infrastructure, update and recovery while preserving an investigation record. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Run access, injection, isolation, capability, resource, secret, model/tool substitution, target replay, packet tamper, key rotation, signed-update, no-egress, support/admin, incident and clean-room recovery tests; map every finding to owner, risk, remedy, retest and accepted residual state. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

| QUESTION | REQUIRED RECORD | OWNER / STOP |
|---|---|---|
| **What is claimed?** | Exact wording, scope, date, source | Guide decision owner |
| **What exists?** | Delivered manifest, interface, artefact | Technical/product owner |
| **What is configured?** | Customer data, roles, rules, posture | Customer control owner |
| **What happened?** | Raw run, target state, packet, metric | Domain/operations owner |
| **What is assured?** | Test/report scope, findings, limits | Independent challenger |
| **What is committed?** | Offer/agreement, remedy, exit term | Procurement/legal owner |

Evidence ladder for assurance evidence package; not every question needs every row, but no row may be implied without evidence.

**DECISION RECORD**

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment.

CHAPTER 15 · CHALLENGE

# Stress-test the boundary behind "Assurance evidence package"

A deep-dive topic earns an additional page for counterexamples, adversarial conditions, missing dependencies, role failure, recovery, and evidence limits.

The principal guide risk is: The highest-impact path may cross trusted prompts, poisoned sources, compromised dependencies, over-capable tools, human/admin bypass, external target replay, packet disclosure, support access or an update channel even when the core network perimeter is strong.

For this chapter, remove or alter the strongest assumption in "architecture, data-flow, keys, controls, tests, update path, incident process, and responsibility matrix to request.". Change the source or version, revoke a person or right, replace a delivered dependency, interrupt the selected deployment, introduce incompatible state, withhold a target acknowledgement, tamper with evidence, or make the relied-on contract condition unavailable. Observe whether the boundary refuses, degrades, escalates, reconciles, recovers, or fails silently.

Run the fault from the perspective of CISO and risk owner, product/application security, security architecture, SOC/incident response, identity and keys, privacy/data, model/tool and source owners, platform operations, evidence/records, supplier and independent assurance. Each owner should see only the authorised information needed to act, understand the current state, and have an explicit safe next step. Restore normal operation, retrieve a pre-change record, and decide whether the chapter proposition still holds, now holds conditionally, requires remediation, or blocks reliance.

| STRESS LANE | INJECTION | EXPECTED EVIDENCE |
|---|---|---|
| **Authority** | Role absent, revoked, conflicted or bypassed | Wait/refuse/escala te + receipt |
| **Dependency** | Source, service, package, model or key unavailable | Degraded state + manifest + recovery |
| **Integrity** | Alter, delete, reorder, substitute | Verification failure + quarantine |
| **Operation** | Load, outage, clock, storage or update fault | Alert + safe state + restore |
| **Target** | Timeout, duplicate, rejection or conflicting state | Idempotency + reconciliation |
| **Exit** | Supplier access removed or contract ends | Export + local verification + continuity result |

Select applicable lanes and add topic-specific consequences before the run.

**STOP RULE**

If a critical assumption fails without a detectable safe state and accountable owner, stop the proof or production lane until the boundary is repaired.

CHAPTER 16 · EXPLAIN

# Residual risk and customer duties

source quality, identity administration, policy ownership, infrastructure hardening, retention, lawful basis, and human review.

This chapter owns a specific part of the reader's decision: source quality, identity administration, policy ownership, infrastructure hardening, retention, lawful basis, and human review.

In this guide, security is expressed across the complete work path: identity and purpose, source and context, models and agents, tool capabilities, human gates, target actions, evidence and keys, deployment and management planes, packages and updates, support, incident response, recovery and exit. Applied to "Residual risk and customer duties", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Architecture describes intended controls; implementation and configuration realise them; tests and operational evidence show behaviour; independent reports have scope and date; certification or regulatory status must be stated exactly. Open or inspectable code does not remove vulnerability or customer hardening duties. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

**Source quality**

Define the exact proposition and current authoritative source.

LENS 2

**Identity administration**

Name the responsible person, system, dependency, and version.

LENS 3

**Policy ownership**

Specify the evidence and how an independent reviewer checks it.

LENS 4

**Infrastructure hardening**

Record scope, failure, unresolved question, and safe next action.

**OWNED ARTEFACT**

This chapter contributes to a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance.

CHAPTER 16 · APPLY

# Turn “Residual risk and customer duties” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, source quality, identity administration, policy ownership, infrastructure hardening, retention, lawful basis, and human review., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant source and version, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: a security team attacks one representative workflow across identity, source, model/tool, capability, human, target, evidence, managed or local infrastructure, update and recovery while preserving an investigation record. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Run access, injection, isolation, capability, resource, secret, model/tool substitution, target replay, packet tamper, key rotation, signed-update, no-egress, support/admin, incident and clean-room recovery tests; map every finding to owner, risk, remedy, retest and accepted residual state. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

| QUESTION | REQUIRED RECORD | OWNER / STOP |
|---|---|---|
| **What is claimed?** | Exact wording, scope, date, source | Guide decision owner |
| **What exists?** | Delivered manifest, interface, artefact | Technical/product owner |
| **What is configured?** | Customer data, roles, rules, posture | Customer control owner |
| **What happened?** | Raw run, target state, packet, metric | Domain/operations owner |
| **What is assured?** | Test/report scope, findings, limits | Independent challenger |
| **What is committed?** | Offer/agreement, remedy, exit term | Procurement/legal owner |

Evidence ladder for residual risk and customer duties; not every question needs every row, but no row may be implied without evidence.

**DECISION RECORD**

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment.

CHAPTER 17 · EXPLAIN

# Security proof plan

concrete adversarial, isolation, permission, tamper, replay, update, and incident tests for a proof environment.

This chapter owns a specific part of the reader's decision: concrete adversarial, isolation, permission, tamper, replay, update, and incident tests for a proof environment.

In this guide, security is expressed across the complete work path: identity and purpose, source and context, models and agents, tool capabilities, human gates, target actions, evidence and keys, deployment and management planes, packages and updates, support, incident response, recovery and exit. Applied to "Security proof plan", the practical task is to name the current state, the desired state, the owner of the transition, the material input and dependency, the evidence a reviewer can inspect, and the condition that would make the claim false or the design unsuitable.

The governing boundary is this: Architecture describes intended controls; implementation and configuration realise them; tests and operational evidence show behaviour; independent reports have scope and date; certification or regulatory status must be stated exactly. Open or inspectable code does not remove vulnerability or customer hardening duties. Keep website publication, delivered software, customer configuration, operating result, independent assurance and written contract as separate evidence layers. A strong statement at one layer must not silently become a guarantee at another.

LENS 1

**Concrete adversarial**

Define the exact proposition and current authoritative source.

LENS 2

**Isolation**

Name the responsible person, system, dependency, and version.

LENS 3

**Permission**

Specify the evidence and how an independent reviewer checks it.

LENS 4

**Tamper**

Record scope, failure, unresolved question, and safe next action.

**OWNED ARTEFACT**

This chapter contributes to a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance.

CHAPTER 17 · APPLY

# Turn “Security proof plan” into decision evidence

Move from explanation to a record an accountable owner can use, challenge, update, and carry into proof, procurement, operation, or exit.

Begin with the chapter proposition, concrete adversarial, isolation, permission, tamper, replay, update, and incident tests for a proof environment., and write it as one or more falsifiable questions. For each question, define scope and time, authoritative website source or customer source, delivered artefact, relevant delivered mechanism, named owner, comparison or acceptance rule, raw evidence, known limitation, and decision consequence.

Use the guide-wide scenario: a security team attacks one representative workflow across identity, source, model/tool, capability, human, target, evidence, managed or local infrastructure, update and recovery while preserving an investigation record. Walk it end to end and ask what this chapter changes at request, source, model/tool/rule, human authority, target, evidence, deployment, operation, commercial or exit state. Do not accept a screenshot or summary when the underlying manifest, result, packet, record or contractual term can be inspected.

The proof discipline for this guide is: Run access, injection, isolation, capability, resource, secret, model/tool substitution, target replay, packet tamper, key rotation, signed-update, no-egress, support/admin, incident and clean-room recovery tests; map every finding to owner, risk, remedy, retest and accepted residual state. Preserve negative and unverifiable results. A gap can be remediated, contracted, accepted by the competent owner, or block the decision; it must not disappear behind roadmap language or an average score.

| QUESTION | REQUIRED RECORD | OWNER / STOP |
|---|---|---|
| **What is claimed?** | Exact wording, scope, date, source | Guide decision owner |
| **What exists?** | Delivered manifest, interface, artefact | Technical/product owner |
| **What is configured?** | Customer data, roles, rules, posture | Customer control owner |
| **What happened?** | Raw run, target state, packet, metric | Domain/operations owner |
| **What is assured?** | Test/report scope, findings, limits | Independent challenger |
| **What is committed?** | Offer/agreement, remedy, exit term | Procurement/legal owner |

Evidence ladder for security proof plan; not every question needs every row, but no row may be implied without evidence.

**DECISION RECORD**

Record pass, conditional pass, gap, fail, or unverifiable; link evidence, owner, remediation, date, residual risk, and effect on determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment.

CHAPTER 17 · CHALLENGE

# Stress-test the boundary behind "Security proof plan"

A deep-dive topic earns an additional page for counterexamples, adversarial conditions, missing dependencies, role failure, recovery, and evidence limits.

The principal guide risk is: The highest-impact path may cross trusted prompts, poisoned sources, compromised dependencies, over-capable tools, human/admin bypass, external target replay, packet disclosure, support access or an update channel even when the core network perimeter is strong.

For this chapter, remove or alter the strongest assumption in "concrete adversarial, isolation, permission, tamper, replay, update, and incident tests for a proof environment.". Change the source or version, revoke a person or right, replace a delivered dependency, interrupt the selected deployment, introduce incompatible state, withhold a target acknowledgement, tamper with evidence, or make the relied-on contract condition unavailable. Observe whether the boundary refuses, degrades, escalates, reconciles, recovers, or fails silently.

Run the fault from the perspective of CISO and risk owner, product/application security, security architecture, SOC/incident response, identity and keys, privacy/data, model/tool and source owners, platform operations, evidence/records, supplier and independent assurance. Each owner should see only the authorised information needed to act, understand the current state, and have an explicit safe next step. Restore normal operation, retrieve a pre-change record, and decide whether the chapter proposition still holds, now holds conditionally, requires remediation, or blocks reliance.

| STRESS LANE | INJECTION | EXPECTED EVIDENCE |
|---|---|---|
| **Authority** | Role absent, revoked, conflicted or bypassed | Wait/refuse/escala te + receipt |
| **Dependency** | Source, service, package, model or key unavailable | Degraded state + manifest + recovery |
| **Integrity** | Alter, delete, reorder, substitute | Verification failure + quarantine |
| **Operation** | Load, outage, clock, storage or update fault | Alert + safe state + restore |
| **Target** | Timeout, duplicate, rejection or conflicting state | Idempotency + reconciliation |
| **Exit** | Supplier access removed or contract ends | Export + local verification + continuity result |

Select applicable lanes and add topic-specific consequences before the run.

**STOP RULE**

If a critical assumption fails without a detectable safe state and accountable owner, stop the proof or production lane until the boundary is repaired.

FIND A SUBJECT

# Subject index

Page references point to the substantive explanation, worked example, control, boundary, or decision record, not merely to a passing label.

SUBJECT INDEX CONTINUED

# Subject index · K–V

Terms are grouped alphabetically. Consecutive page references are collapsed into ranges.

SUBJECT INDEX CONTINUED

# Subject index · W–W

Terms are grouped alphabetically. Consecutive page references are collapsed into ranges.

CLOSING DECISION RECORD

# Close with one evidence-backed, reversible decision

The guide is complete when its reader can decide whether to proceed, narrow, remediate, contract a condition, hold, reject, or exit, and another person can understand why.

Assemble a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance. State the question, scope and exclusions, scenario or workflow, current baseline, canonical source snapshot, delivered manifest, customer configuration, corpus and method, results and failures, owner decisions, risks, operational and commercial conditions, evidence links, remedies, stop/exit and review triggers.

Run the final discipline: Run access, injection, isolation, capability, resource, secret, model/tool substitution, target replay, packet tamper, key rotation, signed-update, no-egress, support/admin, incident and clean-room recovery tests; map every finding to owner, risk, remedy, retest and accepted residual state.

Have CISO and risk owner, product/application security, security architecture, SOC/incident response, identity and keys, privacy/data, model/tool and source owners, platform operations, evidence/records, supplier and independent assurance sign only the boundaries they own. Record pass, conditional pass, gap, fail or unverifiable per proposition. Preserve dissent and raw evidence. A brochure can organise the decision; it cannot make the evidence true or take responsibility from the people who rely on it.

| DECISION RECORD | security-architecture-assurance-decision | |
|---|---|---|
| QUESTION | **determine whether the platform's security boundaries and evidence are suitable for deeper diligence, threat modelling, and a scoped deployment.** | BOUNDED |
| SOURCES | **website copy + delivered material** | CAPTURED |
| PROOF | **corpus + failures + recovery** | RUN |
| OWNERS | **domain + technical + assurance** | SIGNED |
| CONDITIONS | **gaps + remedies + contract** | DATED |
| DECISION | **go / narrow / hold / stop / exit** | RECORDED |

The closing record remains useful after the presentation, release, team, supplier, or contract changes.

**GUIDE OUTPUT**

Produce a threat model and assurance package with assets, actors, boundaries, abuse cases, controls, evidence, residual risk, responsibilities, tests, findings, remedies, incident paths and acceptance.