# POST-QUANTUM PENTEST

# POST-QUANTUM PENTEST

## Ethical Hacking and Resilience Testing for the Quantum Transition

**Prof. Dr. Šarūnas Grigaliūnas**

*Kaunas University of Technology*

*Cyber Security Centre of Excellence*

Educator handbook and Wiley-style authoring manuscript

Copyright © 2026 Šarūnas Grigaliūnas. All rights reserved.

This is a working educational manuscript prepared for editorial review. It has not been published or endorsed by John Wiley & Sons, Inc.

All laboratory targets are local and intentionally designed for authorised study. Third-party trademarks belong to their respective owners. No certification endorsement is implied.

*To learners and defenders who turn uncertainty into resilient systems.*

# Contents

Preface xix

Acknowledgments xxi

Acronyms xxiii

**1 The Quantum Threat Reframed for Security Testers 1**

**1.1 From future event to present exposure** . . . . . 2

**1.2 Confidentiality and authenticity have different clocks** . . . . . 3

**1.3 The Qacker adversary model** . . . . . 4

**1.4 A tester's first quantum-safe questions** . . . . . 5

**1.5 Current practice and project connections** . . . . . 6

**1.6 Instructor lens** . . . . . 7

**1.7 Hands-on laboratory** . . . . . 7

**Review and discussion** . . . . . 7

**2 Standards, Evidence, and the Emerging Discipline 9**

**2.1 The standards stream** . . . . . 9

**2.2 Migration and discovery guidance** . . . . . 11

**2.3 Protocol engineering and hybrid deployment** . . . . . 13

**2.4 Implementation research and the evidence gap** . . . . . 14

**2.5 Current practice and project connections** . . . . . 15

**2.6 Instructor lens** . . . . . 16

**2.7 Hands-on laboratory** . . . . . 16

**Review and discussion** . . . . . 16

**3 PAREX, PAREK, and the Testing Architecture** **19**

**3.1 Prepare and authorise** . . . . . 19

**3.2 Asset-discover and normalise** . . . . . 22

**3.3 Risk-evaluate and evidence-test** . . . . . 23

**3.4 Execute transition and retest** . . . . . 23

**3.5 Current practice and project connections** . . . . . 24

**3.6 Instructor lens** . . . . . 25

**3.7 Hands-on laboratory** . . . . . 25

**Review and discussion** . . . . . 25

**4 Reconnaissance and Cryptographic Asset Discovery** **27**

**4.1 Start passive and stay in scope** . . . . . 27

**4.2 Discover cryptography in Transit, Use, and Rest** . . . . . 29

**4.3 Build a cryptographic bill of materials** . . . . . 30

**4.4 Validate coverage and ownership** . . . . . 31

**4.5 Current practice and project connections** . . . . . 32

**4.6 Instructor lens** . . . . . 32

**4.7 Hands-on laboratory** . . . . . 33

**Review and discussion** . . . . . 33

**5 Laboratory Toolchain and Safe Experiment Design** **35**

**5.1 Design for isolation and repeatability** . . . . . 35

**5.2 Fixtures before live targets** . . . . . 36
**5.3 Evidence capture and deterministic outputs** . . . . . 38
**5.4 Assessment rubrics for practical work** . . . . . 38
**5.5 Current practice and project connections** . . . . . 39
**5.6 Instructor lens** . . . . . 40
**5.7 Hands-on laboratory** . . . . . 40
**Review and discussion** . . . . . 40

**6 Testing Data in Transit** **41**
**6.1 Classical baseline before quantum analysis** . . . . . 41
**6.2 TLS, SSH, VPN, and API evidence** . . . . . 42
**6.3 Hybrid negotiation and downgrade paths** . . . . . 44
**6.4 Performance, middleboxes, and observability** . . . . . 44
**6.5 Current practice and project connections** . . . . . 45
**6.6 Instructor lens** . . . . . 46
**6.7 Hands-on laboratory** . . . . . 46
**Review and discussion** . . . . . 46

**7 Testing Data in Use** **49**
**7.1 Source, dependency, and runtime discovery** . . . . . 49
**7.2 Signing, tokens, and build pipelines** . . . . . 50
**7.3 Cryptographic agility as a testable property** . . . . . 51
**7.4 Editing and preserving assessment models** . . . . . 52
**7.5 Current practice and project connections** . . . . . 53

**7.6 Instructor lens** . . . . . 54

**7.7 Hands-on laboratory** . . . . . 54

**Review and discussion** . . . . . 54

**8 Testing Data at Rest** **55**

**8.1 Separate bulk encryption from key protection** . . . . . 55

**8.2 Archives, backups, and rewrapping** . . . . . 56

**8.3 Secrets managers and password vaults** . . . . . 57

**8.4 Retention, recovery, and deletion evidence** . . . . . 58

**8.5 Current practice and project connections** . . . . . 59

**8.6 Instructor lens** . . . . . 59

**8.7 Hands-on laboratory** . . . . . 60

**Review and discussion** . . . . . 60

**9 Implementation Security, Side Channels, and Fault Thinking** **61**

**9.1 Algorithm security is not implementation security** . . . . . 61

**9.2 Timing, cache, power, and error channels** . . . . . 62

**9.3 Randomness and failure handling** . . . . . 64

**9.4 Triage before specialist laboratory work** . . . . . 64

**9.5 Current practice and project connections** . . . . . 65

**9.6 Instructor lens** . . . . . 65

**9.7 Hands-on laboratory** . . . . . 66

**Review and discussion** . . . . . 66

**10 PKI, Certificates, and Hybrid Identity** **67**

**10.1 Map the trust system, not only certificates** . . . . . 67
**10.2 Long-term validation and archival authenticity** . . . . . 68
**10.3 Hybrid and composite transition choices** . . . . . 70
**10.4 Firmware, document, and software signing** . . . . . 70
**10.5 Current practice and project connections** . . . . . 71
**10.6 Instructor lens** . . . . . 71
**10.7 Hands-on laboratory** . . . . . 72
**Review and discussion** . . . . . 72

11 QKD, Quantum Communication, and Hybrid Assurance **73**
**11.1 Keep QKD and PQC claims distinct** . . . . . 73
**11.2 Trusted nodes, endpoints, and key management** . . . . . 74
**11.3 The Lithuanian and European QCI context** . . . . . 76
**11.4 Assurance for hybrid infrastructures** . . . . . 76
**11.5 Current practice and project connections** . . . . . 77
**11.6 Instructor lens** . . . . . 78
**11.7 Hands-on laboratory** . . . . . 78
**Review and discussion** . . . . . 78

12 Reporting, Scoring, and Executive Communication **79**
**12.1 Observed fact, inference, and forecast** . . . . . 79
**12.2 QARS and Quantum-Safe Index context** . . . . . 80
**12.3 From finding to decision** . . . . . 81
**12.4 Retest criteria and evidence retention** . . . . . 82

**12.5 Current practice and project connections** . . . . . 83
**12.6 Instructor lens** . . . . . 83
**12.7 Hands-on laboratory** . . . . . 84
**Review and discussion** . . . . . 84

13 Practical CTF and Training Scenarios **85**
**13.1 Teach decisions, not tool theatre** . . . . . 85
**13.2 Challenge patterns and evidence flags** . . . . . 86
**13.3 Team roles and safe competition** . . . . . 87
**13.4 Debriefing and transfer to the workplace** . . . . . 88
**13.5 Current practice and project connections** . . . . . 88
**13.6 Instructor lens** . . . . . 89
**13.7 Hands-on laboratory** . . . . . 89
**Review and discussion** . . . . . 89

14 Ethics, Legal Boundaries, and Professional Responsibility **91**
**14.1 Authorisation, minimisation, and accuracy** . . . . . 91
**14.2 Avoid quantum fear, uncertainty, and doubt** . . . . . 92
**14.3 Responsible implementation-security work** . . . . . 94
**14.4 Certification alignment and professional limits** . . . . . 94
**14.5 Current practice and project connections** . . . . . 94
**14.6 Instructor lens** . . . . . 95
**14.7 Hands-on laboratory** . . . . . 95
**Review and discussion** . . . . . 96

15 From Book to Field Programme 97

15.1 A fourteen-week educator pathway . . . . . 97

15.2 Tools that preserve models and secrets . . . . . 99

15.3 Lithuanian projects and public infrastructure . . . . . 100

15.4 Service patterns, research, and continuous improvement . . . . 101

15.5 Current practice and project connections . . . . . 101

15.6 Instructor lens . . . . . 102

15.7 Hands-on laboratory . . . . . 102

Review and discussion . . . . . 103

A PAREX Field Checklist 105

A.1 Minimum engagement record . . . . . 106

A.2 Minimum CBOM fields . . . . . 106

B Sample Findings and Evidence Language 107

B.1 Finding quality pattern . . . . . 108

C Glossary and Teaching Vocabulary 109

D Educator Rubrics and Certification Crosswalk 111

D.1 Suggested assessment weighting . . . . . 111

D.2 Certification crosswalk . . . . . 111

# List of Figures


**1.1** **The migration timing window: data lifetime plus migration time can exceed the time available before a cryptographically relevant quantum capability.** . . . . . 4

**2.1** **Standards-to-evidence chain used in this handbook.** . . . . . 13

**3.1** **PAREX operating cycle for repeatable post-quantum pentesting.** . . . . . 23

**4.1** **Transit, Use, and Rest discovery surfaces and their evidence artefacts.** . . . . . 31

**5.1** **Safe laboratory topology used by the companion exercises.** . . 37

**6.1** **Hybrid key establishment combines independent classical and post-quantum contributions before deriving traffic keys.** 43

**7.1** **A CBOM links code, libraries, runtime services, keys, data, and owners.** . . . . . 51

**8.1** **Layered protection of a stored secret: password derivation, KEM wrapping, symmetric data key, and encrypted payload.** 57

**9.1** **Implementation assurance model: inputs and faults can create observable leakage around an otherwise sound algorithm.** 63

**10.1 Signature lifecycle and the evidence required to preserve authenticity through algorithm transition.** . . . . . 69

**11.1 Layered QKD and PQC assurance: optical key delivery does not remove endpoint, authentication, or governance obligations.** . . . . . 75

**12.1 Evidence pipeline from RQmod source through normalised findings, CBOM, risk scoring, and decision-ready reporting.** 81

**13.1 Capstone laboratory evidence graph and team workflow.** . . 87

**14.1 Certification and educator alignment across engagement, discovery, testing, analysis, and reporting.** . . . . . 93

**15.1 Educator learning loop from concept and laboratory evidence to assessment, reflection, and field transfer.** . . . . . 100

# List of Tables

**D.1 Suggested course assessment weighting.** . . . . . 112

**D.2 High-level certification crosswalk; always verify the current official blueprint.** . . . . . 112

# Preface

Post-quantum security is no longer only an algorithm-selection problem. It is a migration, evidence, implementation, interoperability, and governance problem. This handbook is written for educators, students, penetration testers, security analysts, architects, and programme owners who need to make that transition observable and testable.

The book uses a strict ethical boundary. Every exercise is designed for local containers, static fixtures, or local Python simulations. Readers must not scan, probe, modify, fault-inject, or test systems without explicit written authorisation. The recurring discipline is to separate what was observed from what was inferred and what remains uncertain.

The material can supplement preparation for CEH, ISECOM OPST/OPSA, and CompTIA PenTest+, but it is independent teaching material. Certification owners update their objectives; students should always consult the current official blueprint.

Prof. Dr. Šarūnas Grigaliūnas

Kaunas, Lithuania

August 2026

# Acknowledgments

The author acknowledges the public work of NIST, CISA, NSA, the European Commission, IETF, ISECOM, OWASP, the Open Quantum Safe community, and the educators and practitioners building a responsible quantum-safe transition.

Š. G.

# Acronyms

| | |
|---|---|
| AEAD | Authenticated Encryption with Associated Data |
| CBOM | Cryptographic Bill of Materials |
| CRQC | Cryptographically Relevant Quantum Computer |
| HNDL | Harvest Now, Decrypt Later |
| KEM | Key-Encapsulation Mechanism |
| ML-DSA | Module-Lattice-Based Digital Signature Algorithm |
| ML-KEM | Module-Lattice-Based Key-Encapsulation Mechanism |
| PAREK | Post-quantum inventory, assessment, roadmap, execution, and key governance lifecycle |
| PAREX | Prepare, Asset-discover, Risk-evaluate, Evidence-test, eXecute transition |
| PQC | Post-Quantum Cryptography |
| QARS | Quantum-Adjusted Risk Score |

| | |
|---|---|
| QCI | Quantum Communication Infrastructure |
| QKD | Quantum Key Distribution |
| QSI | Quantum-Safe Index |
| SLH-DSA | Stateless Hash-Based Digital Signature Algorithm |

# Chapter 1

# The Quantum Threat Reframed for Security Testers

**Chapter overview:** This chapter develops the post-quantum pentesting knowledge and evidence practices needed to explain why quantum risk is observable before a cryptographically relevant quantum computer exists.

**Keywords:** post-quantum cryptography, penetration testing, cryptographic agility, evidence, education

After reading this chapter, you should be able to:

- Explain why quantum risk is observable before a cryptographically relevant quantum computer exists.
- Distinguish harvest-now-decrypt-later confidentiality risk from long-term authenticity risk.
- Identify public-key dependencies that ordinary vulnerability scanners commonly miss.
- Translate a quantum threat statement into a testable security claim.

## 1.1. From future event to present exposure

For many years quantum computing appeared in security presentations as a distant disruption: important, fascinating, but rarely connected to the everyday work of a penetration tester. That framing is now inadequate. A tester does not need a cryptographically relevant quantum computer on the desk to observe quantum risk. The risk appears when an organization depends on RSA, Diffie-Hellman, ECDH, ECDSA or DSA for data that must remain confidential or authentic beyond the useful life of those algorithms. It appears when TLS certificates, VPN handshakes, firmware signing flows, S/MIME deployments, SSH keys, blockchain identities, device update chains and HSM-backed trust anchors are treated as permanently safe even though their mathematical foundations are not quantum-resistant.

The core threat model is simple to state but difficult to operationalize. Shor-type attacks threaten widely deployed public-key cryptography once sufficiently large and reliable quantum machines exist. Grover-type search affects symmetric-key and hash security margins differently, generally requiring key-size and parameter adjustments rather than a complete redesign of symmetric cryptography. The public-key problem is more disruptive because it is embedded into identity, key establishment, certificates, signatures, code signing, device provisioning, secure boot, transport protocols and regulatory assurance. A penetration tester therefore has to look for places where public-key cryptography is not visible to normal vulnerability scanners.

## 1.2. Confidentiality and authenticity have different clocks

The most urgent operational case is harvest-now-decrypt-later. An adversary can record encrypted traffic or exfiltrate encrypted archives now and wait for future cryptanalytic capability. The breach is not visible at the time the data is harvested because the attacker does not need to decrypt it immediately. This changes the meaning of confidentiality testing. A normal pentest might report that a TLS endpoint currently negotiates a strong classical suite and has no known downgrade flaw. A post-quantum pentest adds a time horizon: if the data protected by that connection must remain secret for ten, twenty or thirty years, then the endpoint's current classical key exchange is an exposure even if no classical vulnerability exists today.

The same logic applies to authenticity, but with a different timeline. A recorded signature does not necessarily become invalid because a future quantum computer exists. However, systems that rely on long-term signature verification, certificate chains, legal non-repudiation, firmware provenance or archival integrity need a strategy for time-stamping, re-signing, hybrid attestations and post-quantum verification. The tester must distinguish confidentiality risk from authenticity risk. Confidentiality can fail retroactively against harvested ciphertext. Authenticity can fail prospectively or legally if the organization cannot demonstrate that signatures were generated and validated in a trustworthy period.

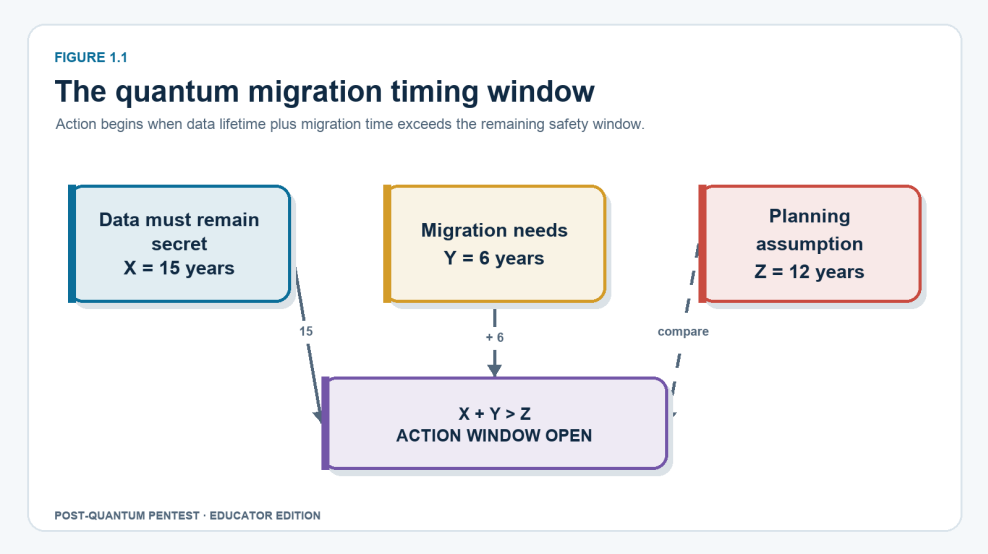


**Figure 1.1:** The migration timing window: data lifetime plus migration time can exceed the time available before a cryptographically relevant quantum capability.

## 1.3. The Qacker adversary model

Post-quantum testing also changes the definition of an asset. In classical testing, assets are hosts, applications, credentials, data stores, endpoints and identities. In PQC testing, cryptographic primitives become first-class assets. A single RSA public key in a forgotten integration can be more strategically important than a server banner. A certificate template, a TLS library, a Java keystore, a cloud KMS configuration, a mobile application signing certificate or an SSH trust policy may decide whether migration succeeds. This is why CBOM thinking is essential. The tester is not only asking what software exists, but which algorithms, parameters, keys, protocols, libraries and dependencies protect which data and business processes.

The publication landscape confirms this shift. NIST's FIPS 203 defines ML-KEM for key establishment and describes the relationship of ML-KEM security

to the Module Learning with Errors problem while specifying ML-KEM-512, ML-KEM-768 and ML-KEM-1024 [National Institute of Standards and Technology, 2024a]. FIPS 204 defines ML-DSA for post-quantum digital signatures [National Institute of Standards and Technology, 2024b]. FIPS 205 defines SLH-DSA as a stateless hash-based signature standard based on SPHINCS+ [National Institute of Standards and Technology, 2024c]. NIST's broader PQC project page states that organizations should begin applying these standards now, because migration requires identifying vulnerable algorithm use and planning replacement [National Institute of Standards and Technology, 2026]. The tester's job is to turn this statement into evidence.

## 1.4. A tester's first quantum-safe questions

This book uses the term Qacker for a quantum-era adversary model. A Qacker does not need to be a lone attacker with a quantum machine. The realistic Qacker is an intelligence service, large criminal consortium, well-funded industrial actor, or supply-chain adversary that combines classical intrusion tradecraft with long-term cryptanalytic patience. Qackers harvest encrypted material, map cryptographic dependencies, exploit migration mistakes, abuse hybrid downgrade paths, target certificates and identity anchors, and search for implementation weaknesses in new PQC code. They are dangerous not because all systems collapse at once, but because the transition itself creates new seams.

A post-quantum pentest therefore has three simultaneous missions. First, it

must find quantum-vulnerable cryptography in systems that matter. Second, it must assess whether planned or deployed PQC is correctly implemented, interoperable and governable. Third, it must translate both findings into risk language that boards and regulators can act on. This combination is what differentiates PQC pentesting from academic cryptanalysis. The tester does not prove ML-KEM secure or insecure. The tester evaluates whether an organization has deployed, configured, monitored and governed cryptography in a way that is resilient under a credible quantum transition timeline.

## 1.5. Current practice and project connections

A useful planning test is Mosca's timing inequality: if the time for which data must remain protected plus the time required to migrate is greater than the time before a relevant quantum capability, the organisation has already entered its action window. The variables are uncertain, but uncertainty is not permission to set them to zero. The practical task is to record assumptions, owners, and a review date.

For students, the key intellectual move is from prediction to verification. A tester does not need to assert when a quantum computer will arrive. The tester can verify that an archive requires twenty years of confidentiality, that migration has no funded owner, and that the current key-wrapping method is quantum-vulnerable. Those are present facts with present governance consequences.

## 1.6. Instructor lens

Teach this chapter by asking learners to separate observed evidence, interpretation, and forecast. Require every claim to name its asset, owner, security property, time horizon, and evidence source. A suggested 90-minute pattern is 25 minutes of concepts, 35 minutes of guided evidence work, 20 minutes of team interpretation, and a 10-minute debrief.

**Certification alignment.** CEH: ethical hacking foundations and reconnaissance; OPST/OPSA: scope and operational security metrics; PenTest+: engagement management.

## 1.7. Hands-on laboratory

Complete companion lab01-scope-and-ethics in the labs directory. Begin with its rules of engagement, preserve the generated evidence, answer the interpretation questions, and compare the result with the educator solution only after submitting a first analysis. The laboratory is local-only by design.

## Review and discussion

1. Why can a correctly configured classical TLS endpoint still represent a high quantum-era risk?
2. Give one example in which authenticity outlives confidentiality and one in which the reverse is true.
3. Write a one-sentence claim that a tester could verify without making a prediction about the date of a quantum computer.

4. State one limitation of the chapter's laboratory and propose a safe next test.

# Chapter 2

# Standards, Evidence, and the Emerging Discipline

**Chapter overview:** This chapter develops the post-quantum pentesting knowledge and evidence practices needed to summarise the roles of FIPS 203, FIPS 204, FIPS 205, and NIST SP 800-227.

**Keywords:** post-quantum cryptography, penetration testing, cryptographic agility, evidence, education

After reading this chapter, you should be able to:

- Summarise the roles of FIPS 203, FIPS 204, FIPS 205, and NIST SP 800-227.
- Separate standards conformance from implementation and deployment assurance.
- Use primary sources to define a defensible target state for a pentest.
- Explain why post-quantum pentesting joins migration, protocol, implementation, and governance evidence.

## 2.1. The standards stream

There are few publications that explicitly use the term post-quantum pentesting. That absence is itself important. The discipline is emerging at the intersection of four streams: standards and migration guidance, protocol en-

gineering, implementation-security research, and automated discovery or risk-scoring literature. A useful book on PQC pentesting must synthesise those streams rather than wait for a mature subfield to name itself. The standards define what must eventually replace vulnerable cryptography. Migration guidance defines how organizations should find, prioritise and replace cryptographic dependencies. Protocol drafts define how hybrid deployments are likely to look in real networks. Implementation-security publications warn that formally selected algorithms can still fail through leakage, faults, randomness, parameter mistakes or unsafe integration.

The standards stream begins with NIST's first three final PQC FIPS documents. FIPS 203 establishes ML-KEM as the main KEM standard for general key establishment [National Institute of Standards and Technology, 2024a]. FIPS 204 establishes ML-DSA as a digital-signature standard [National Institute of Standards and Technology, 2024b]. FIPS 205 establishes SLH-DSA as a stateless hash-based digital-signature standard [National Institute of Standards and Technology, 2024c]. NIST has also selected HQC as a backup KEM candidate with different mathematical foundations from ML-KEM, while emphasizing that organizations should continue migrating to the 2024 standards [National Institute of Standards and Technology, 2025]. For a pentester, this means the target state is no longer vague. The assessor can test whether a roadmap, procurement specification or proof of concept actually aligns with named standards and their intended uses.

The migration stream is equally important. NIST IR 8547 frames the transition away from quantum-vulnerable algorithms and identifies the need for timelines that inform agencies, industry and standards organizations [Moody et al., 2024]. The NCCoE Migration to PQC project emphasizes two workstreams:

cryptographic discovery and interoperability testing [National Cybersecurity Center of Excellence, 2026]. This is almost a direct description of the first two practical phases of post-quantum pentesting. Discovery produces inventories of where cryptography is used. Interoperability testing checks whether new PQC or hybrid mechanisms operate in controlled environments before production rollout. A pentest adds adversarial and evidence discipline to these activities: it asks what is missed, what can be downgraded, what assumptions fail, and what audit proof is credible.

## 2.2. Migration and discovery guidance

The protocol-engineering stream shows how hybrid systems will behave. The IETF hybrid TLS design draft defines hybrid key exchange as simultaneous use of multiple key exchange algorithms with combined outputs so that security remains if at least one component remains unbroken [Stebila et al., 2025]. The ECDHE-MLKEM TLS draft defines concrete TLS 1.3 groups such as X25519MLKEM768, SecP256r1MLKEM768 and SecP384r1MLKEM1024 [Kwiatkowski et al., 2026]. These are not merely standards-committee details. They become pentest artefacts. A tester can inspect ClientHello and ServerHello messages, verify group negotiation, detect unsupported fallbacks, measure handshake size effects, check policy decisions, and observe whether middleboxes or monitoring tools break when key-share payloads grow.

The implementation-security stream warns against naive optimism. The survey Algorithmic Security is Insufficient argues that PQC algorithms can be vulnerable to side-channel and active implementation attacks even if their mathematical design is sound [Cintas Canto et al., 2023]. Work on attacks

against Kyber-style implementations illustrates the practical importance of power analysis, leakage models, countermeasures and hardware behaviour [Wang et al., 2024]. These publications should not be read as reasons to avoid PQC. They should be read as reasons to test deployments with the same discipline that mature cryptographic engineering already demands. A deployment that simply says "Kyber inside" or "ML-KEM enabled" is not automatically resilient. The tester must ask how keys are generated, whether decapsulation failures leak information, whether constant-time claims are credible, whether randomness is robust, and whether devices are exposed to physical or local attackers.

The automated tooling stream is rapidly developing. Recent work on quantum-safe code auditing proposes static analysis and contextual enrichment to find quantum-vulnerable primitives in codebases and prioritize findings [Shaw, 2026]. Other work such as QERS explores resilience scoring across constrained environments and protocols [Rassekhnia, 2026]. Performance and deployment studies evaluate how PQC behaves on hardware classes, cloud stacks, IoT devices and networks. These publications suggest that future PQC pentesting will be heavily automated but not fully automatic. Tools can find evidence; testers must validate, contextualize and report it. ResilQ is positioned exactly in this gap: it can ingest cryptographic evidence, produce QARS-based scoring, generate CBOMs and make the result usable for governance.

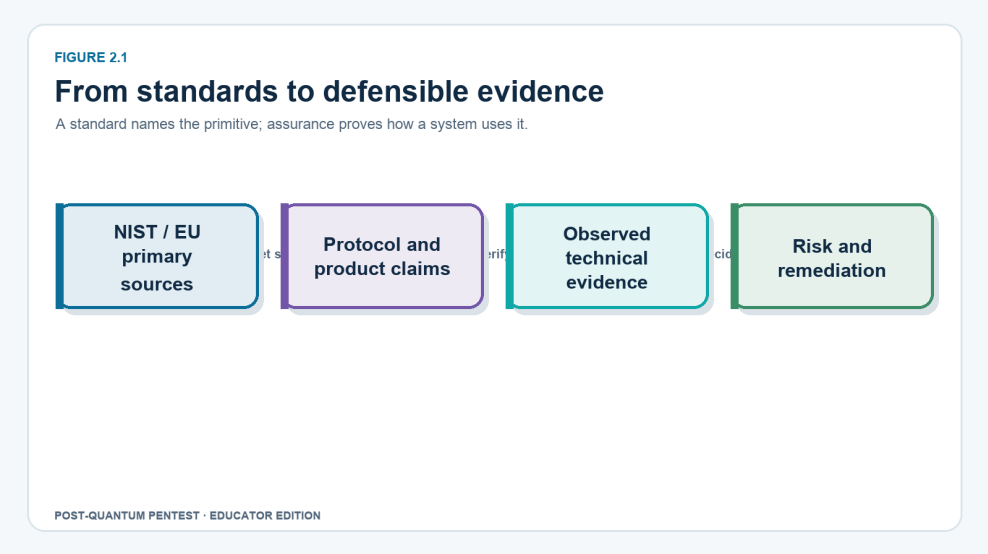


**Figure 2.1:** Standards-to-evidence chain used in this handbook.

## 2.3. Protocol engineering and hybrid deployment

The author's own QARS paper contributes a crucial missing bridge. QARS formalizes a quantum-adjusted risk score using timeline, sensitivity and exposure dimensions, embedded within the PAREK framework [Grigaliunas and Bruzgiene, 2025]. This matters for pentesting because a report that merely lists RSA and ECDSA occurrences is not enough. One RSA-2048 certificate on a public marketing site and another RSA-2048 key protecting twenty-year health records should not receive the same priority. QARS-style scoring makes the pentest context-aware. It asks how long the data must remain protected, how sensitive it is, how exposed it is, and how hard it will be to migrate.

The key gap across the literature is the absence of a unified practitioner playbook. Standards say what algorithms to use. Guidance says migration

is necessary. Protocol drafts show hybrid constructions. Research shows implementation pitfalls. Tools propose discovery and scoring methods. But the security-testing profession still needs a workflow that combines legal authorization, reconnaissance, cryptographic inventory, protocol validation, implementation assurance, migration risk scoring, evidence handling and executive reporting. That workflow is PAREX. In this book, PAREX is not a theoretical replacement for standards. It is the operational pentest wrapper around them.

A second gap is that current publications often separate confidentiality, authenticity, availability and governance. PQC pentesting must integrate them. A TLS migration may improve long-term confidentiality while increasing handshake size, causing performance problems, triggering middlebox failure, or breaking monitoring assumptions. A signature migration may improve authenticity while increasing firmware update size, causing constrained-device failures. A rushed hybrid certificate scheme may preserve compatibility but create validation ambiguity. Therefore the post-quantum pentest report must be multidisciplinary: cryptographic, architectural, operational, regulatory and managerial.

## 2.4. Implementation research and the evidence gap

The conclusion of the publication analysis is direct. "Post-quantum pentest" is an emerging label for a necessary practice. Its evidence base already exists, but it is scattered. The discipline should be built around standards-grounded targets, migration-oriented discovery, hybrid protocol testing, implementation

security, automated CBOM evidence, and risk scoring. ResilQ and PAREX provide the connective methodology. The tester becomes the person who can say not only "you use RSA here," but "this RSA exposure protects long-lived regulated data, appears in this dependency chain, lacks migration ownership, has no compensating hybrid control, and should be remediated before lower-risk assets." That is the value proposition.

## 2.5. Current practice and project connections

NIST published FIPS 203 (ML-KEM), FIPS 204 (ML-DSA), and FIPS 205 (SLH-DSA) on 13 August 2024. NIST selected HQC in March 2025 as a future backup KEM based on different mathematics, while stating that migration to the 2024 standards should continue. NIST SP 800-227 became final in September 2025 and provides secure-use guidance for KEMs. Assessors should record document status and date; a draft, a selected algorithm, and a final standard are not equivalent evidence [National Institute of Standards and Technology, 2026, Alagic et al., 2025].

The EU coordinated implementation roadmap published in June 2025 asks Member States to begin the transition in a synchronised way. The CISA, NSA, and NIST quantum-readiness material similarly emphasises roadmaps, inventories, risk assessment, and vendor engagement. These sources support the testing sequence used here: discover, contextualise, validate, and govern [European Commission and NIS Cooperation Group, 2025, CISA and NSA and NIST, 2023].

## 2.6. Instructor lens

Teach this chapter by asking learners to separate observed evidence, interpretation, and forecast. Require every claim to name its asset, owner, security property, time horizon, and evidence source. A suggested 90-minute pattern is 25 minutes of concepts, 35 minutes of guided evidence work, 20 minutes of team interpretation, and a 10-minute debrief.

**Certification alignment.** CEH: vulnerability analysis; OPSA: evidence-led analysis; PenTest+: vulnerability discovery and analysis.

## 2.7. Hands-on laboratory

Complete companion lab02-crypto-inventory in the labs directory. Begin with its rules of engagement, preserve the generated evidence, answer the interpretation questions, and compare the result with the educator solution only after submitting a first analysis. The laboratory is local-only by design.

## Review and discussion

1. What does a FIPS algorithm standard prove, and what does it not prove about a product?
2. Why is HQC relevant even though ML-KEM remains the recommended general-purpose KEM?
3. Which source should an assessor cite for secure KEM usage guidance after September 2025?
4. State one limitation of the chapter's laboratory and propose a safe next

test.

# Chapter 3

# PAREX, PAREK, and the Testing Architecture

**Chapter overview:** This chapter develops the post-quantum pentesting knowledge and evidence practices needed to apply the PAREX cycle to an authorised post-quantum assessment.

**Keywords:** post-quantum cryptography, penetration testing, cryptographic agility, evidence, education

After reading this chapter, you should be able to:

- Apply the PAREX cycle to an authorised post-quantum assessment.
- Relate PAREX testing activities to the PAREK migration lifecycle.
- Define evidence requirements before choosing tools.
- Create a retestable finding whose risk score remains linked to assets and data.

## 3.1. Prepare and authorise

PAREX is the book's operational methodology for post-quantum penetration testing. It extends the PAREK framework used in QARS research into a field-ready testing cycle. PAREK gives the strategic governance structure: post-quantum asset and algorithm inventory, risk assessment, road mapping, execution and key governance [Grigaliunas and Bruzgiene, 2025]. PAREX turns

that structure into authorized assessment activity: Prepare, Asset-discover, Risk-evaluate, Evidence-test and eXecute continuous transition. The method is intentionally simple enough to remember during a client engagement but detailed enough to support repeatable evidence and tooling.

Prepare means defining the legal, technical and business scope before any cryptographic probing begins. The tester identifies owners, systems, data classes, test windows, permitted scanning ranges, source-code access, cloud accounts, HSM boundaries, certificate authorities, vendor dependencies, and rules for handling cryptographic secrets. This phase also defines the quantum risk question. Is the client worried about long-term confidentiality, regulatory readiness, customer assurance, supply-chain exposure, firmware authenticity, cloud migration, critical infrastructure resilience or executive visibility? The answer determines what evidence matters. Without a precise risk question, PQC pentesting becomes a generic scan that produces noise.

Asset-discover means building the cryptographic view of the environment. The tester enumerates TLS endpoints, SSH services, VPN gateways, PKI assets, code repositories, containers, packages, mobile applications, device firmware, cloud KMS policies, signing pipelines, S/MIME or PGP use, database encryption, backup encryption, API gateways and identity providers. The output is a CBOM-style inventory. It should capture algorithm names, key sizes, certificate chains, protocol versions, library versions, dependency paths, data protected, business process, owner and lifecycle. ResilQ can represent this across Transit, Use and Rest layers. Transit covers network and protocol protection. Use covers application execution, signing, key handling and active service logic. Rest covers stored data, backups, archives, secrets and encrypted databases.

Risk-evaluate means scoring the evidence. A post-quantum finding is not

a normal CVSS item. RSA-2048 in one place and RSA-2048 in another place can represent radically different risk. QARS provides a better model because it includes timeline, sensitivity and exposure [Grigaliunas and Bruzgiene, 2025]. The timeline dimension asks whether the protected data or signature assurance must survive until a cryptographically relevant quantum computer may exist. The sensitivity dimension asks how damaging disclosure or forgery would be. The exposure dimension asks whether the cryptographic material is visible to adversaries, stored by adversaries, used in public protocols, embedded in products, or protected behind layered controls. ResilQ should produce scores that are explainable rather than magical. Every score must be traceable to evidence.

Evidence-test means validating the claim under controlled conditions. This is the phase that differentiates a pentest from a policy assessment. The tester verifies observed TLS groups, checks whether legacy fallbacks are still accepted, confirms whether certificates are classical or hybrid, runs OQS lab handshakes, measures handshake size and latency, attempts authorized downgrade scenarios, reviews source code for vulnerable primitive use, examines error handling, checks whether library versions actually support claimed algorithms, validates CBOM accuracy, and confirms that monitoring or WAF systems see what they need to see. For implementation security, evidence-testing may include code review for constant-time practices, randomness review, side-channel threat modelling or laboratory analysis of a test device. It does not mean uncontrolled attacks on production cryptography.

eXecute continuous transition means turning findings into a roadmap, control updates and exercises. A PQC pentest report should not end with "replace RSA." It should provide an ordered migration plan: what to retire, what to hy-

bridize, what to monitor, what to defer, what vendor evidence to request, what policy to change, what procurement language to adopt, and what tests should be repeated after deployment. This phase also includes tabletop exercises. For example: what happens if NIST changes a parameter recommendation, if a PQC library receives a critical advisory, if a vendor cannot supply a CBOM, if a middlebox breaks hybrid TLS, or if a sector regulator asks for evidence of quantum transition planning?

ResilQ is the engineering layer that makes PAREX repeatable. It should support RQmod scripts that declare audit blocks, evidence sources, constraints, backend profiles and output formats. A minimal ResilQ job might scan TLS endpoints, parse certificates, produce a CBOM, generate a QARS input file, execute a controlled OQS handshake, and create a management report. A richer job might include source-code scanning, dependency analysis, QKD/QMS integration checks, SpinQ demonstration output, and a DORA/NIS2-aligned control map. The platform should remain evidence-first. It should never hide uncertainty behind a shiny score.

## 3.2. Asset-discover and normalise

A small RQmod example illustrates the idea:

```
audit "banking_tls_pqc_readiness" {
scope transit {
targets = ["api.example.internal:443", "vpn.example.internal:443"]
collect = ["tls_groups", "cert_chain", "cipher_suites", "library_hint"]
}
scope use {
```

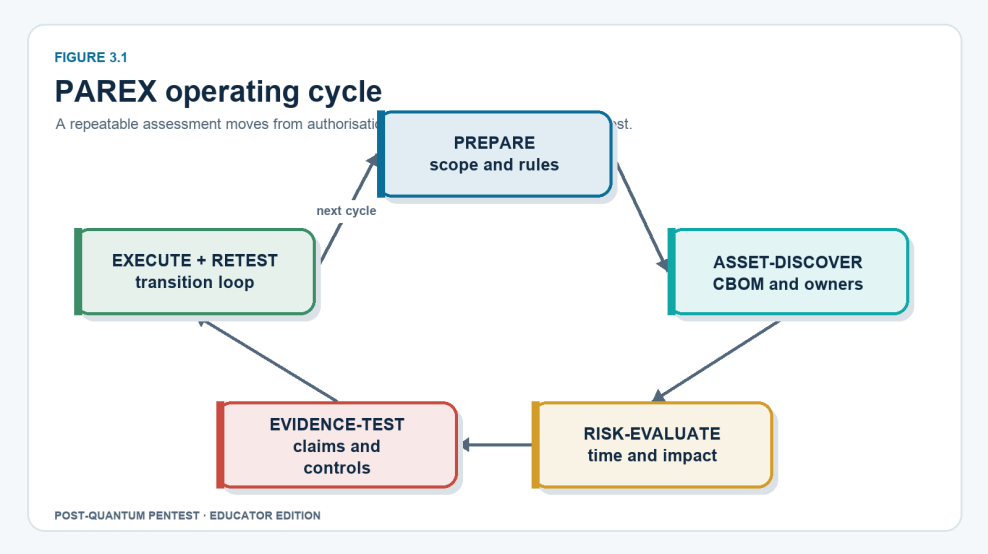


**Figure 3.1:** PAREX operating cycle for repeatable post-quantum pentesting.

## 3.3. Risk-evaluate and evidence-test

repositories = [”git@example:payments/api.git”]

detect = [”rsa”, ”ecdsa”, ”ecdh”, ”dh”, ”x509”, ”jwt_signing”]

}

scoring qars {

data_lifetime_years = 15

sensitivity = ”regulated_financial”

exposure = ”public_and_partner”

## 3.4. Execute transition and retest

}

outputs = [”cbom.json”, ”qars-input.json”, ”resilq-report.docx”]

}

The value of such a language is not elegance for its own sake. It makes the pentest reproducible. Another team can run the same job after remediation and compare the evidence. A board can see that the score changed because an algorithm disappeared or a data lifetime was reclassified, not because a consultant changed wording. A regulator can see the chain from discovery to risk to decision. This is precisely what post-quantum migration needs: not fear, but traceable governance.

## 3.5. Current practice and project connections

The PAREK lifecycle and Quantum-Safe Index provide governance and maturity views; PAREX is the adversarial-assurance wrapper used in this book. A maturity score is not a vulnerability. A vulnerability finding is not a migration programme. Keeping these artefacts linked but distinct prevents both audit theatre and tool-driven scope creep.

RQmod is a small declarative language for expressing authorised targets, Transit/Use/Rest layers, constraints, QARS-NI inputs, and evidence outputs. The companion laboratory compiles a deliberately limited subset into normalised JSON. It is a teaching implementation, not the complete ResilQ Studio compiler available through https://rqmod.com/.

## 3.6. Instructor lens

Teach this chapter by asking learners to separate observed evidence, interpretation, and forecast. Require every claim to name its asset, owner, security property, time horizon, and evidence source. A suggested 90-minute pattern is 25 minutes of concepts, 35 minutes of guided evidence work, 20 minutes of team interpretation, and a 10-minute debrief.

**Certification alignment.** CEH: methodology; OPST/OPSA: operational testing and analysis; PenTest+: engagement management and reporting.

## 3.7. Hands-on laboratory

Complete companion lab03-risk-window in the labs directory. Begin with its rules of engagement, preserve the generated evidence, answer the interpretation questions, and compare the result with the educator solution only after submitting a first analysis. The laboratory is local-only by design.

## Review and discussion

1. Which PAREX phase must define whether intrusive implementation tests are permitted?
2. How does a PAREK roadmap become testable through PAREX?
3. Name three evidence artefacts that should survive after a tool has been replaced.
4. State one limitation of the chapter's laboratory and propose a safe next test.

# Chapter 4

# Reconnaissance and Cryptographic Asset Discovery

**Chapter overview:** This chapter develops the post-quantum pentesting knowledge and evidence practices needed to design passive and active discovery steps that respect rules of engagement.

**Keywords:** post-quantum cryptography, penetration testing, cryptographic agility, evidence, education

After reading this chapter, you should be able to:

- Design passive and active discovery steps that respect rules of engagement.
- Locate cryptography in network services, repositories, signing systems, and stored data.
- Produce a minimum viable cryptographic bill of materials (CBOM).
- Identify blind spots and false confidence in automated discovery.

## 4.1. Start passive and stay in scope

Reconnaissance in a post-quantum pentest is broader and quieter than in a conventional network assessment. The tester is not primarily looking for open ports or vulnerable banners, although those still matter. The tester is looking

for cryptographic dependencies and algorithmic commitments. These commitments are often hidden inside configuration files, libraries, certificates, token issuers, package manifests, build scripts, firmware images, signing services, cloud policies and vendor appliances. A good PQC reconnaissance phase therefore combines network scanning, source-code review, dependency discovery, certificate analysis, architecture interviews and documentation review.

The first target is the external cryptographic perimeter. Public TLS endpoints reveal certificate algorithms, key sizes, signature chains, TLS versions, cipher suites and supported groups. The tester should record whether ECDHE or finite-field DH is used, whether RSA or ECDSA authenticates certificates, whether old TLS versions remain enabled, and whether the certificate chain relies on classical signatures throughout. This does not mean the endpoint is "broken" today. It means the endpoint's future confidentiality and authenticity posture must be understood. If the endpoint carries low-sensitivity content, it may be lower priority. If it carries health, finance, energy, government or legal data with long retention, it becomes high priority.

The second target is the internal cryptographic perimeter. Internal APIs, service meshes, Kubernetes ingress controllers, message brokers, database connections, privileged administration portals and east-west traffic often use certificates that are never inspected by external tools. A post-quantum test should ask whether internal traffic carries long-lived sensitive data, whether private CAs can migrate, whether certificate templates are algorithm-agile, and whether automation can rotate keys without outages. Internal PKI is frequently more difficult to migrate than public web certificates because it is entangled with device identity, service discovery, legacy applications and operational change windows.

The third target is source code and package dependencies. RSA and ECC can appear in obvious imports, but they also appear indirectly through libraries such as JWT frameworks, SAML toolkits, SSH libraries, TLS wrappers, payment SDKs, cloud libraries and embedded device SDKs. Static analysis should search for algorithm names, key generation calls, certificate handling, signature verification logic, hardcoded curves, PEM headers, keystore paths and custom crypto code. Automated tools can accelerate discovery, but human review is necessary to classify whether the usage protects confidentiality, authenticity, key agreement, identity or test-only code. A false positive in a unit test is less important than a hidden RSA key in a production token issuer.

The fourth target is cryptographic material at rest. Backups, archives, encrypted databases, object storage, document repositories, secrets vaults and offline media can carry the most severe harvest-now-decrypt-later consequences. The tester should map which encryption algorithms protect them, how keys are wrapped, whether public-key encryption is used for envelope keys, how long data must remain confidential, and whether adversaries could have copied ciphertext. At-rest exposures should be linked to data retention. A seven-day log archive and a thirty-year legal archive are not equivalent.

## 4.2. Discover cryptography in Transit, Use, and Rest

The fifth target is identity and signatures. Code signing, firmware signing, secure boot, package repositories, document signing, eID systems, S/MIME, PGP, container signing and transaction signatures require authenticity over

time. A tester should identify signing algorithms, certificate lifetimes, timestamping practices, revocation dependencies, re-signing policies and verification environments. Quantum transition for signatures is complicated because verifiers are often distributed across devices that cannot be updated quickly. A firmware update signed today may need to be verified by a field device ten years from now. The pentest should make such dependencies visible.

Reconnaissance should produce a structured CBOM, not just a narrative. A useful CBOM includes the component, algorithm, parameter, key length, protocol, library, version, data protected, owner, business criticality, exposure, data lifetime, certificate expiry, rotation mechanism, vendor dependency and evidence source. ResilQ should be able to import this CBOM and link it to QARS scoring. The CBOM is a living artefact. The pentest report should recommend how the client will keep it current after the engagement.

A sample command sequence for an authorized perimeter assessment might be:

# TLS and certificate discovery for explicitly authorized targets only

nmap –script ssl-enum-ciphers -p 443,8443 -oA pqc_tls_scan targets.txt

## 4.3. Build a cryptographic bill of materials

testssl.sh –wide –jsonfile tls_findings.json https://api.example.org

openssl s_client -connect api.example.org:443 -showcerts ¡/dev/null ¿ cert_chain.txt

# Extract public-key algorithm evidence from certificates

openssl x509 -in leaf.pem -noout -text — egrep ”Public Key Algorithm—Signature

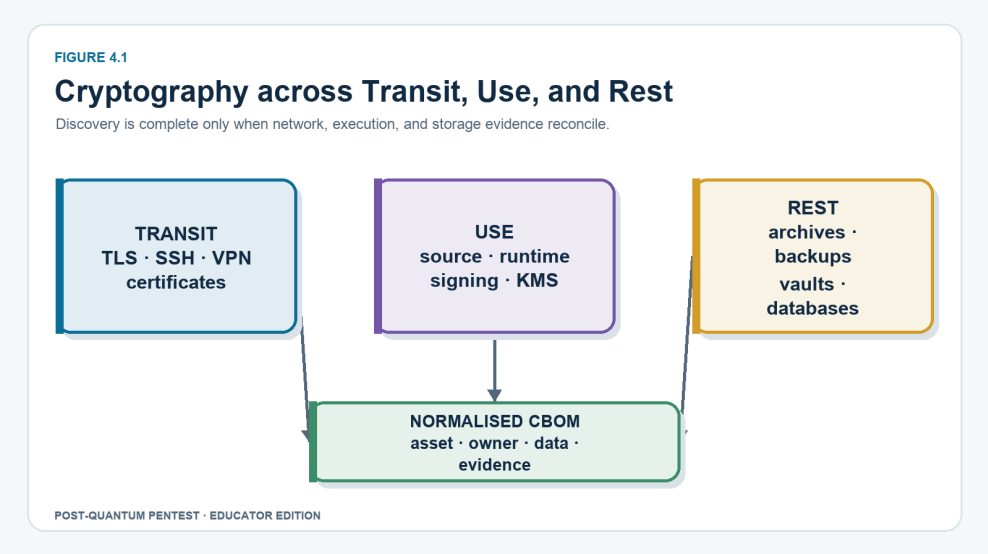


**Figure 4.1:** Transit, Use, and Rest discovery surfaces and their evidence artefacts.

Algorithm—Public-Key”

# Source-code discovery patterns, to be refined by language and repository

## 4.4. Validate coverage and ownership

rg -n ”RSA—ECDSA—ECDH—DSA—Diffie—X25519—secp256r1—P-256—P-384—crypto\.generateKeyPair—BEGIN RSA” .

Commands are only the beginning. The tester must validate context. A tool may report RSA in a certificate chain, but the report must explain what data the certificate protects, how long that data matters, whether the endpoint is public, whether there is a hybrid option, who owns the migration, and what operational constraints exist. The reconnaissance layer is successful when cryptographic dependencies become governable assets.

## 4.5. Current practice and project connections

Discovery quality is measured twice: by precision and by coverage. Precision asks whether a reported primitive really exists in the stated context. Coverage asks whether important contexts were omitted. A clean scan with unknown coverage is weak assurance. Sampling, owner interviews, configuration exports, source searches, packet evidence, and supplier attestations should be reconciled rather than treated as interchangeable.

The public resource https://www.pqc.lt/ turns Lithuanian migration guidance into actions around inventory, risk, roadmaps, procurement, and tools. It can be used as a local-language bridge for students before they work with the English primary standards.

## 4.6. Instructor lens

Teach this chapter by asking learners to separate observed evidence, interpretation, and forecast. Require every claim to name its asset, owner, security property, time horizon, and evidence source. A suggested 90-minute pattern is 25 minutes of concepts, 35 minutes of guided evidence work, 20 minutes of team interpretation, and a 10-minute debrief.

**Certification alignment.** CEH: footprinting, scanning, and enumeration; OPST: channel testing; PenTest+: reconnaissance and enumeration.

## 4.7. Hands-on laboratory

Complete companion lab02-crypto-inventory in the labs directory. Begin with its rules of engagement, preserve the generated evidence, answer the interpretation questions, and compare the result with the educator solution only after submitting a first analysis. The laboratory is local-only by design.

## Review and discussion

1. Why is a certificate inventory not a complete CBOM?
2. What makes a discovery false negative more dangerous than a low-severity configuration finding?
3. List the minimum owner and data-context fields needed to prioritise an algorithm finding.
4. State one limitation of the chapter's laboratory and propose a safe next test.

# Chapter 5

# Laboratory Toolchain and Safe Experiment Design

**Chapter overview:** This chapter develops the post-quantum pentesting knowledge and evidence practices needed to construct an isolated laboratory with explicit safety boundaries.

**Keywords:** post-quantum cryptography, penetration testing, cryptographic agility, evidence, education

After reading this chapter, you should be able to:

- Construct an isolated laboratory with explicit safety boundaries.
- Choose between containers, static fixtures, and Python simulations.
- Capture evidence without collecting unnecessary secrets.
- Assess laboratory work using observable outcomes rather than tool screenshots.

## 5.1. Design for isolation and repeatability

A post-quantum pentest book must be practical, but practice must be safe. The right approach is to create local laboratories that reproduce the kind of evidence a tester needs without attacking third-party systems. The base-

line laboratory should include a Linux workstation or VM, Docker, Python, OpenSSL 3, Open Quantum Safe components, a test web server, a test certificate authority, intentionally classical certificates, hybrid TLS configurations, a local Git repository with classical crypto examples, and ResilQ/RQmod scripts that convert evidence into CBOM and QARS inputs. The objective is not to become a cryptographer in one afternoon. The objective is to learn how standards, protocols, libraries and risk scoring interact.

Open Quantum Safe is the natural toolchain for prototyping because it provides liboqs and integrations into protocols and applications such as OpenSSL [Open Quantum Safe Project, 2026]. OQS is not a magic production approval stamp. It is a research and testing ecosystem. That distinction is important in the book. A pentester may use OQS to demonstrate hybrid TLS negotiation, compare key sizes, run regression tests, and educate teams. Production deployments require vendor support, validated modules where applicable, policy approval and lifecycle management. In a lab, however, OQS gives testers the hands-on feel that ordinary standards documents cannot provide.

## 5.2. Fixtures before live targets

A simple lab progression begins with classical TLS. The tester creates a local certificate authority, issues an RSA certificate and configures a web server. The tester then scans the endpoint and records the evidence. Next, the tester runs an OQS-enabled server or provider and negotiates a hybrid group, comparing the handshake, key-share sizes and compatibility. The purpose is not to declare one setup universally superior. The purpose is to understand what changes when PQC enters the protocol. Larger handshake messages may affect

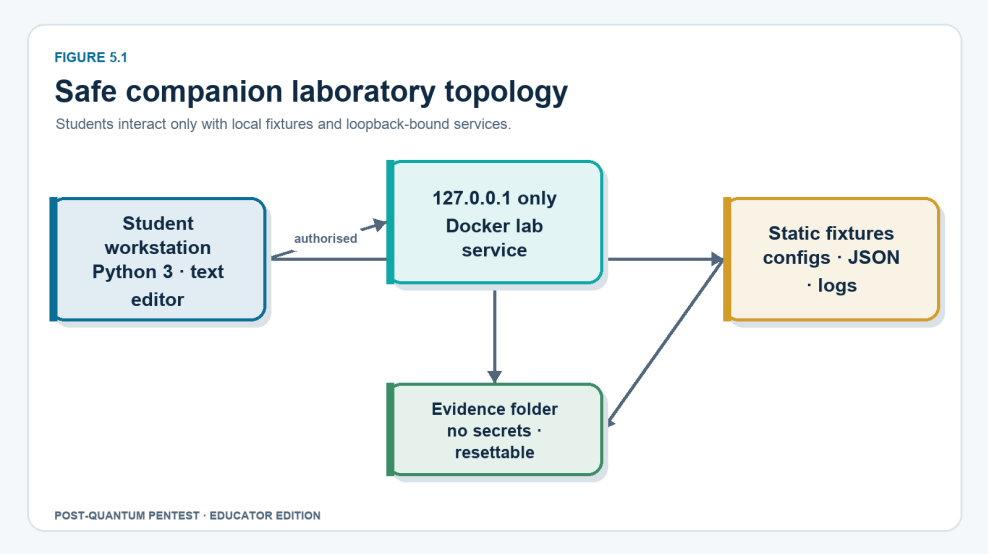


**Figure 5.1:** Safe laboratory topology used by the companion exercises.

middleboxes, packet fragmentation, latency and monitoring. The tester should learn to observe those changes rather than repeat marketing claims.

The lab should also include code discovery. A small repository can contain Python, JavaScript and Java examples that use RSA for JWT signing, ECDH for shared secrets, ECDSA for signatures, and modern symmetric encryption. The task is to generate a CBOM, classify each usage and decide which uses are quantum-vulnerable. For example, AES-GCM with appropriate key sizes is not the same problem as RSA-OAEP. HMAC is not the same problem as ECDSA. The tester should learn not to label all cryptography as equally broken. Good PQC pentesting is precise.

## 5.3. Evidence capture and deterministic outputs

ResilQ can tie the lab together. An RQmod job can declare the TLS endpoint, repository path, data lifetime and sensitivity. The output can include cbom.json, qars-input.json, a findings table and a management summary. A second run after remediation should show fewer quantum-vulnerable findings or lower scores. This teaches the most important operational idea: post-quantum readiness is not a binary badge. It is a measurable, evidence-backed change in the environment.

The lab should include a SpinQ-style demonstration if available. A two-qubit desktop quantum computer will not break RSA, but it can make quantum behaviour visible to executives, students and practitioners. A Bell-state or Deutsch-Jozsa demonstration can be connected to the narrative: quantum computing changes what computation can do, while PQC changes how we prepare security. ResilQ can present this as an educational module, not as a fake attack. The distinction protects credibility. Leaders should see the quantum threat without being misled into thinking a small educational quantum device is a cryptanalytic weapon.

## 5.4. Assessment rubrics for practical work

A safe experiment design template should include purpose, asset, authorization, expected evidence, safety limits, rollback, data handling and reporting. For example, "Test hybrid TLS handshake in local lab" is safe. "Attempt to trigger decapsulation failure behaviour on a production appliance" is not ac-

ceptable unless the vendor and owner explicitly authorize it in a controlled environment. "Review source code for RSA usage" is safe with repository authorization. "Extract private keys from devices" is a separate hardware-security engagement with stronger legal and safety controls.

The book's labs should therefore follow three rules. First, never test systems you do not own or have written authorization to assess. Second, never handle production secrets unless the engagement explicitly requires it and secure handling procedures exist. Third, separate cryptographic evidence from exploit theatre. PQC pentesting is valuable because it produces defensible evidence. It loses value when it becomes a performance of fear.

## 5.5. Current practice and project connections

Every companion lab defaults to local fixtures or loopback services. Where Docker is used, the published port is bound to 127.0.0.1. Scripts reject arbitrary remote targets unless an instructor deliberately changes the code and assumes responsibility for a separately authorised environment. This design teaches useful evidence handling without turning a classroom exercise into an Internet scanner.

A strong practical submission contains the scope statement, command or script version, raw evidence, interpretation, limitation, and remediation. Screenshots may support a narrative, but they should not replace machine-readable artefacts or a reproducible procedure.

## 5.6. Instructor lens

Teach this chapter by asking learners to separate observed evidence, interpretation, and forecast. Require every claim to name its asset, owner, security property, time horizon, and evidence source. A suggested 90-minute pattern is 25 minutes of concepts, 35 minutes of guided evidence work, 20 minutes of team interpretation, and a 10-minute debrief.

**Certification alignment.** CEH Practical: controlled tool use; OPST/OPSA: reproducible operations; PenTest+: tool selection and scripting.

## 5.7. Hands-on laboratory

Complete companion lab04-transit-profile in the labs directory. Begin with its rules of engagement, preserve the generated evidence, answer the interpretation questions, and compare the result with the educator solution only after submitting a first analysis. The laboratory is local-only by design.

## Review and discussion

1. When is a static packet or configuration fixture preferable to a live service?
2. What must be reset between student cohorts?
3. Which evidence proves a result without exposing a private key?
4. State one limitation of the chapter's laboratory and propose a safe next test.

# Chapter 6

# Testing Data in Transit

**Chapter overview:** This chapter develops the post-quantum pentesting knowledge and evidence practices needed to establish a defensible classical transport-security baseline.

**Keywords:** post-quantum cryptography, penetration testing, cryptographic agility, evidence, education

After reading this chapter, you should be able to:

- Establish a defensible classical transport-security baseline.
- Inspect key establishment and certificate evidence across common protocols.
- Test hybrid negotiation without confusing support with actual use.
- Evaluate availability and observability regressions introduced by migration.

## 6.1. Classical baseline before quantum analysis

Data in transit is the most visible starting point for PQC pentesting because protocol handshakes expose measurable evidence. TLS endpoints advertise supported versions, groups, certificates and signature algorithms. VPNs and SSH systems reveal key exchange and host-key choices. APIs use TLS, JWTs, mTLS, OAuth tokens, SAML assertions and service certificates. The tester's task is to map the cryptographic path from client to server, identify

quantum-vulnerable components, test migration options and detect downgrade or compatibility risks.

The TLS 1.3 hybrid drafts are particularly useful for testers. The general hybrid-design draft describes a construction where multiple key exchange algorithms are used together and their outputs are combined to preserve security if at least one component remains secure [Stebila et al., 2025]. The ECDHE-MLKEM draft defines concrete hybrid mechanisms such as X25519MLKEM768, SecP256r1MLKEM768 and SecP384r1MLKEM1024 [Kwiatkowski et al., 2026]. These drafts give the tester a checklist. Which hybrid groups are supported? Does the server negotiate them only with compatible clients? What happens if the client offers only classical groups? Can policy enforce hybrid-only for high-risk services? Do middleboxes pass the handshake? Do logs capture enough evidence? Are failure modes safe?

## 6.2. TLS, SSH, VPN, and API evidence

A transit-layer finding should distinguish between key establishment and authentication. A server may support hybrid key exchange but still present a classical certificate chain. This improves resistance to harvest-now-decrypt-later confidentiality attacks for sessions where the hybrid group is used, but it does not make the authentication chain post-quantum. Conversely, a future hybrid certificate scheme may help authentication but not protect a connection if the negotiated key exchange is classical. Testers must avoid simplistic pass/fail labels. The report should state which security property is improved and which remains classical.

VPNs add additional complexity. IPsec, WireGuard-like systems, SSL VPNs

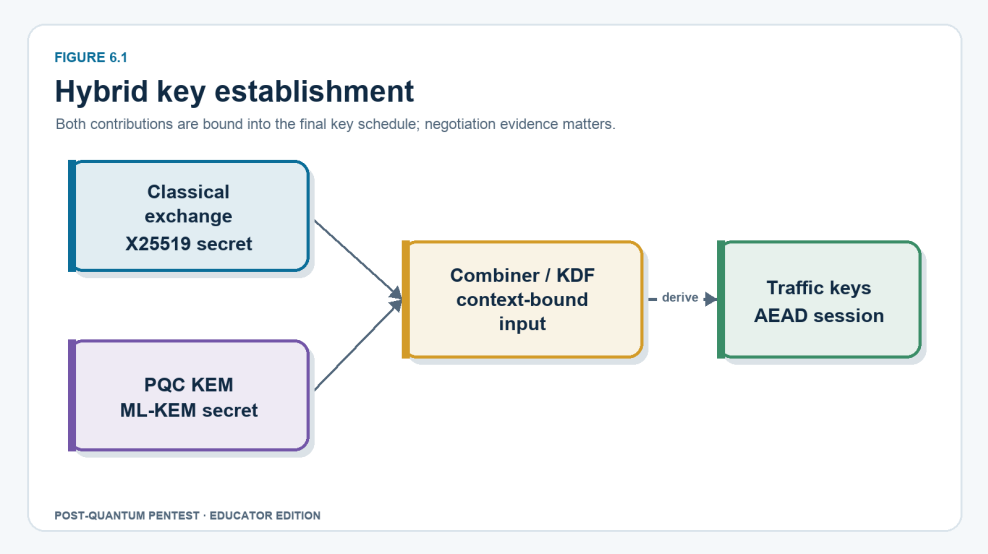


**Figure 6.1:** Hybrid key establishment combines independent classical and post-quantum contributions before deriving traffic keys.

and proprietary remote-access appliances have different migration paths. Some rely on certificates, some on pre-shared keys, some on ECDH, and some on vendor-specific key exchange. A PQC pentest should request vendor roadmaps, inspect configuration, check whether cryptographic agility exists, and test whether the platform can update algorithms without replacing hardware. If the VPN protects long-lived sensitive traffic, lack of a PQC roadmap becomes a strategic risk even if the appliance has no ordinary CVE.

## 6.3. Hybrid negotiation and downgrade paths

SSH is often overlooked. SSH host keys and user keys may remain in place for years, and internal automation may depend on them. The tester should inventory RSA, ECDSA and Ed25519 keys, key lifetimes, known_hosts distribution, bastion host policies and automation scripts. Post-quantum SSH support is still maturing, but the absence of a migration plan should be recorded. The tester should also identify systems where SSH tunnels protect sensitive administrative traffic that may be logged or captured by adversaries.

APIs require attention beyond TLS. JWTs signed with RS256 or ES256, SAML assertions signed with RSA or ECDSA, webhook signatures, OAuth client assertions, mTLS certificates and service-mesh identities all carry authenticity assumptions. The tester should determine whether token lifetimes are short, whether signatures are used only for immediate validation or long-term legal proof, whether key rotation is automated, and whether algorithms are hardcoded. A quantum-era attacker may not need to break a token today if the token expires in minutes, but a long-term document-signing or audit-proof system is different.

## 6.4. Performance, middleboxes, and observability

Downgrade testing must be controlled and precise. In a lab or authorized staging environment, the tester can configure clients to offer different group lists,

observe server selection, and verify whether policy allows classical-only fallback. The objective is not to break TLS; it is to verify that migration policy is real. If a server claims "PQC ready" but negotiates classical ECDHE with every ordinary client and has no policy for high-sensitivity routes, the claim is weak. If a server supports hybrid groups but fails under normal packet sizes, the migration may create availability risk.

Transit-layer reporting should include a table with endpoint, protocol, observed classical algorithms, observed PQC or hybrid support, certificate chain status, data sensitivity, data lifetime, exposure, downgrade behaviour, owner and recommended action. ResilQ can convert this table into QARS scoring. The best reports are visual. A board should see which services protect long-lived data with quantum-vulnerable key exchange, which services are low risk, and which services are ready for staged hybrid testing.

## 6.5. Current practice and project connections

Hybrid key establishment should be tested as a protocol state machine. Record what each peer advertises, what is selected, how the two secrets are combined, what happens when either contribution is missing or malformed, and what telemetry is generated. Merely observing a PQC library in a software bill of materials does not prove that production traffic uses it.

Transport migration can introduce larger messages, additional CPU work, unexpected fragmentation, middlebox rejection, and monitoring gaps. A good pentest therefore includes availability guardrails and rollback evidence. Quantum-

safe confidentiality that makes a critical service unreliable is not a successful migration.

## 6.6. Instructor lens

Teach this chapter by asking learners to separate observed evidence, interpretation, and forecast. Require every claim to name its asset, owner, security property, time horizon, and evidence source. A suggested 90-minute pattern is 25 minutes of concepts, 35 minutes of guided evidence work, 20 minutes of team interpretation, and a 10-minute debrief.

**Certification alignment.** CEH: sniffing and network attacks; OPST: communications channel testing; PenTest+: reconnaissance, protocol analysis, and attacks.

## 6.7. Hands-on laboratory

Complete companion lab05-hybrid-negotiation in the labs directory. Begin with its rules of engagement, preserve the generated evidence, answer the interpretation questions, and compare the result with the educator solution only after submitting a first analysis. The laboratory is local-only by design.

## Review and discussion

1. How do you prove that a hybrid group was negotiated rather than merely advertised?

2. Why must a failed PQ component cause the intended failure behaviour?
3. What operational evidence would reveal a middlebox problem?
4. State one limitation of the chapter's laboratory and propose a safe next test.

# Chapter 7

# Testing Data in Use

**Chapter overview:** This chapter develops the post-quantum pentesting knowledge and evidence practices needed to find cryptographic calls and dependencies in source and build pipelines.

**Keywords:** post-quantum cryptography, penetration testing, cryptographic agility, evidence, education

After reading this chapter, you should be able to:

- Find cryptographic calls and dependencies in source and build pipelines.
- Prioritise token, code-signing, firmware, and identity uses by lifetime and impact.
- Test whether algorithms can be changed without redesigning the application.
- Use an auditable text workflow for assessment models and evidence.

## 7.1. Source, dependency, and runtime discovery

Data in use is where cryptography becomes application logic. A TLS scan may show the perimeter, but the application decides how tokens are signed, how payloads are encrypted, how keys are generated, how certificates are pinned, how firmware is verified, how messages are authenticated and how secrets are stored during execution. PQC pentesting in this layer combines code review,

dependency analysis, CI/CD inspection, runtime observation and developer interviews.

The first objective is to find public-key cryptography in source code. Search patterns should include algorithm names, curve names, imports, provider configuration, keystore references, JWT signing algorithms, certificate parsing logic, key agreement calls and custom wrappers. Automated discovery is useful, and recent research on quantum-safe code auditing and LLM-assisted cryptographic discovery shows how tools can help reduce manual workload [Shaw, 2026]. But the tester must classify findings. A library that supports RSA is not the same as production use of RSA. A test fixture is not the same as a payment token signer. A hardcoded algorithm policy in production is more important than a generic dependency.

## 7.2. Signing, tokens, and build pipelines

The second objective is to inspect build and signing pipelines. Modern software depends on code signing, container signing, package signing, firmware signing and CI/CD attestations. Many of these systems depend on RSA or ECDSA. The tester should ask how signatures are generated, where keys live, which HSM or KMS protects them, how timestamps are applied, what happens if algorithms must change, and whether older devices or clients can verify post-quantum signatures. Code signing is a quantum authenticity issue with supply-chain implications. A product may be secure today but unable to transition because the update mechanism cannot deliver a new verification root.

The third objective is runtime key handling. Applications often generate ephemeral ECDH keys, load PEM files, call cloud KMS APIs, decrypt envelope

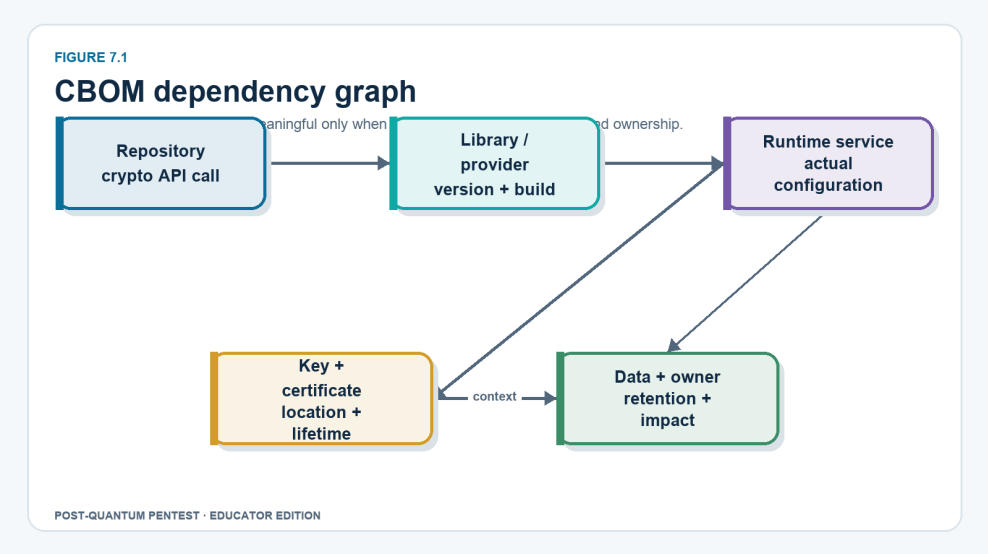


**Figure 7.1:** A CBOM links code, libraries, runtime services, keys, data, and owners.

keys, validate certificates and cache cryptographic material. A post-quantum assessment should identify whether cryptography is centralized or scattered. Centralized cryptographic services can migrate more easily. Scattered custom code creates hidden technical debt. Crypto-agility is therefore an architectural finding, not just an algorithm property.

## 7.3. Cryptographic agility as a testable property

The fourth objective is to inspect dependency freshness. PQC libraries are evolving quickly. Open Quantum Safe, provider integrations and vendor SDKs change as standards and drafts stabilize [Open Quantum Safe Project, 2026]. The tester should check whether the team has a dependency-monitoring pro-

cess, whether cryptographic libraries are pinned, whether security advisories are tracked, and whether test suites detect algorithm or provider changes. A PQC migration that cannot receive updates is not resilient.

The fifth objective is to review error handling. Implementation-security literature repeatedly warns that decryption, decapsulation and signature verification failures can leak information if handled incorrectly [Cintas Canto et al., 2023][Wang et al., 2024]. Even when the tester is not performing physical side-channel work, code review should look for branching, timing differences, verbose failure messages, oracle-like behaviours, retry logic, and logs that reveal sensitive intermediate states. In a lab, developers can instrument toy examples to understand why constant-time and failure-handling discipline matters.

## 7.4. Editing and preserving assessment models

A ResilQ code assessment should produce a repository-level CBOM and a finding map. Each finding should include file path, line or component, algorithm, function, security purpose, data protected, exposure, recommended migration path and confidence. For example: "payments/auth/tokens.py uses RS256 for JWT signing; tokens expire in fifteen minutes; impact is near-term authenticity rather than long-term confidentiality; recommend preparing ML-DSA-compatible roadmap for service-to-service tokens only after ecosystem support is available; immediate mitigation is key rotation and algorithm agility." This is much better than "RSA found."

CI/CD tests can become continuous PQC controls. A build can fail if new

code introduces disallowed algorithms in high-risk contexts. A CBOM can be generated on each release. A ResilQ job can compare current cryptographic inventory against the previous release. QARS scores can be tracked over time. This turns a pentest from a once-a-year document into a feedback loop. The tester's report should propose such controls because manual migration governance does not scale.

## 7.5. Current practice and project connections

Qmacs Editor (https://q-macs-editor.vercel.app/) is a native macOS, offline-first workspace for encrypted notes and programmable buffers. Its current published design uses Argon2id and XChaCha20-Poly1305 for the local vault and includes an RQmod editing mode. It describes ML-KEM and ML-DSA as explicit future integration gates. In teaching material it should therefore be called post-quantum-minded, not represented as proof that every stored note is already protected by PQC.

Students can use Qmacs to edit Markdown findings and .rqmod models while keeping ordinary files separate from password-gated material. The security lesson is broader than the product: classify working notes, minimise captured secrets, encrypt sensitive evidence, and preserve a plain-text, versionable model of the assessment.

## 7.6. Instructor lens

Teach this chapter by asking learners to separate observed evidence, interpretation, and forecast. Require every claim to name its asset, owner, security property, time horizon, and evidence source. A suggested 90-minute pattern is 25 minutes of concepts, 35 minutes of guided evidence work, 20 minutes of team interpretation, and a 10-minute debrief.

**Certification alignment.** CEH: application and system hacking; OPSA: technical analysis; PenTest+: vulnerability analysis, scripting, and remediation.

## 7.7. Hands-on laboratory

Complete companion lab06-source-discovery in the labs directory. Begin with its rules of engagement, preserve the generated evidence, answer the interpretation questions, and compare the result with the educator solution only after submitting a first analysis. The laboratory is local-only by design.

## Review and discussion

1. Why can searching only for the string RSA miss important dependencies?
2. How would you test algorithm agility in a build pipeline?
3. What should an encrypted editor claim - and not claim - when its current vault uses classical authenticated encryption?
4. State one limitation of the chapter's laboratory and propose a safe next test.

# Chapter 8

# Testing Data at Rest

**Chapter overview:** This chapter develops the post-quantum pentesting knowledge and evidence practices needed to distinguish quantum-vulnerable key wrapping from quantum-resistant bulk encryption.

**Keywords:** post-quantum cryptography, penetration testing, cryptographic agility, evidence, education

After reading this chapter, you should be able to:

- Distinguish quantum-vulnerable key wrapping from quantum-resistant bulk encryption.
- Assess archive and backup exposure using retention and replication evidence.
- Threat-model a local-first password vault without treating a product claim as assurance.
- Recommend rewrapping and recovery tests that preserve availability.

## 8.1. Separate bulk encryption from key protection

At-rest cryptography is where the harvest-now-decrypt-later threat becomes most tangible. Encrypted archives, backups, document stores, data lakes, medical records, legal evidence, intelligence records, financial histories and industrial designs may remain sensitive long after the systems that created them are

retired. A conventional pentest may focus on whether access controls prevent exfiltration. A post-quantum pentest adds another question: if the ciphertext is already copied, how long will it remain safe?

The tester should begin by classifying data lifetime. Data lifetime is not the same as retention policy. A system may delete operational logs after one year, but backups, legal holds, analytics exports or vendor support packages may preserve copies. Some data has a confidentiality lifetime that exceeds its storage lifetime because disclosure would harm individuals, national interests or intellectual property even after formal retention ends. QARS explicitly makes timeline a risk dimension [Grigaliunas and Bruzgiene, 2025]. The assessment should therefore document how long confidentiality matters, not merely how long the file exists.

## 8.2. Archives, backups, and rewrapping

At-rest encryption often uses symmetric algorithms, which are not the primary target of Shor-type attacks. The risk usually enters through key wrapping, public-key envelope encryption, access-control systems, certificates, KMS integrations or backup-transfer channels. A database encrypted with AES-256 may be sound from a post-quantum symmetric perspective, but if its data encryption keys are wrapped by an RSA key, or if backup access depends on classical certificates, quantum risk remains. The tester must trace the full key hierarchy.

Backups are a high-value target. They may be offline, poorly inventoried, encrypted once and forgotten, or managed by third-party services. The tester should ask what algorithms protect backup keys, how backup keys are ro-

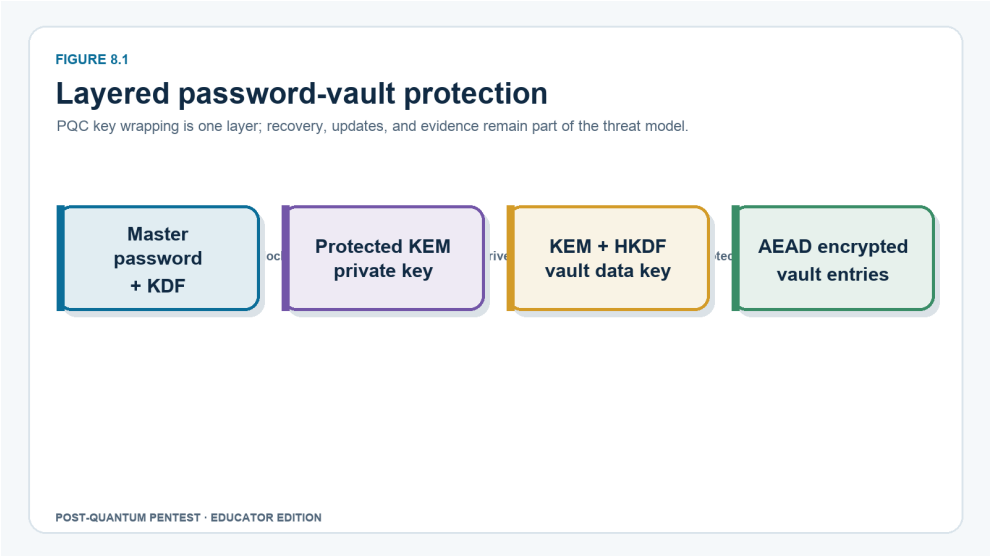


**Figure 8.1:** Layered protection of a stored secret: password derivation, KEM wrapping, symmetric data key, and encrypted payload.

tated, whether old backup sets can be re-encrypted, whether vendor escrow exists, whether administrators can export encrypted blobs, and whether backup metadata reveals sensitive information. For regulated environments, the report should identify which backup classes require early PQC migration because re-encrypting decades of data later may be operationally impossible.

## 8.3. Secrets managers and password vaults

Secrets vaults and KMS systems require a different view. Many organizations assume that using a cloud KMS solves cryptographic migration. It helps, but it does not remove responsibility. The tester should inspect key types, wrapping algorithms, certificate dependencies, API authentication, HSM support, rotation policies, audit logs, residency constraints and vendor PQC roadmaps. For

sovereign or critical infrastructure deployments, ResilQ's on-prem profile can help evaluate whether key material and evidence remain under organizational control.

Document signing and archival integrity require special attention. A signed legal document, forensic report or evidence package may need verification for years. If the signature algorithm becomes quantum-vulnerable, the organization needs trusted timestamping, archival re-signing, or a long-term validation strategy. The tester should not simply recommend replacing every signature immediately. Instead, the report should classify which signatures are transactional and short-lived, which are archival, and which are legal or evidentiary.

## 8.4. Retention, recovery, and deletion evidence

A useful at-rest test case is an archive with envelope encryption. The tester maps the data encryption algorithm, key wrapping method, KMS policy, certificate chain, backup copy locations, data lifetime and re-encryption feasibility. The finding may state: "Archive uses AES-256 for content encryption, but the data-encryption key is wrapped with RSA-2048 in a system containing health records with a twenty-five-year confidentiality horizon. Exposure is high because encrypted archives are replicated to vendor-managed storage. Recommendation: migrate key-wrapping architecture to approved KEM or hybrid key management when production-ready, prioritize new archives first, and plan staged re-wrapping of legacy archives."

At-rest reporting should avoid panic. Not every symmetric encryption sys-

tem is broken. The value lies in tracing dependencies and prioritizing long-lived, high-sensitivity stores. A good PQC pentest can reduce unnecessary work by showing which assets are not urgent. That credibility makes high-risk findings more persuasive.

## 8.5. Current practice and project connections

PassQ (https://passq-app.vercel.app/) is a local-first password manager whose published design wraps a vault key using ML-KEM-768 and protects payload material with symmetric authenticated encryption. It provides command-line and local web interfaces. The project is useful for threat modelling because it exposes the layers of password derivation, private-key protection, encapsulation, key derivation, and payload encryption.

A product page is not an independent audit. Before production recommendation, assess code provenance, release integrity, platform support, cryptographic-module validation needs, secure memory behaviour, backup and recovery, export controls, update signing, and the consequences of a lost master password. The lab asks students to distinguish architectural claims from evidence.

## 8.6. Instructor lens

Teach this chapter by asking learners to separate observed evidence, interpretation, and forecast. Require every claim to name its asset, owner, security property, time horizon, and evidence source. A suggested 90-minute pattern

is 25 minutes of concepts, 35 minutes of guided evidence work, 20 minutes of team interpretation, and a 10-minute debrief.

**Certification alignment.** CEH: cryptography and credential security; OPST/OPSA: data controls; PenTest+: secrets, key rotation, and remediation.

## 8.7. Hands-on laboratory

Complete companion lab09-passq-threat-model in the labs directory. Begin with its rules of engagement, preserve the generated evidence, answer the interpretation questions, and compare the result with the educator solution only after submitting a first analysis. The laboratory is local-only by design.

## Review and discussion

1. Why does AES-256 not automatically make an RSA-wrapped archive quantum-safe?
2. Which copies must be included in a rewrapping plan?
3. What evidence is needed before recommending a new password manager for production?
4. State one limitation of the chapter's laboratory and propose a safe next test.

# Chapter 9

# Implementation Security, Side Channels, and Fault Thinking

**Chapter overview:** This chapter develops the post-quantum pentesting knowledge and evidence practices needed to explain how a secure algorithm can fail in an implementation.

**Keywords:** post-quantum cryptography, penetration testing, cryptographic agility, evidence, education

After reading this chapter, you should be able to:

- Explain how a secure algorithm can fail in an implementation.
- Recognise timing, power, cache, fault, and randomness risk indicators.
- Run a safe timing demonstration without attacking a real product.
- Escalate high-risk devices to specialist evaluation with a clear evidence request.

## 9.1. Algorithm security is not implementation security

The post-quantum transition will fail if organizations treat algorithm selection as the only security question. Implementation matters. Side channels, faults, randomness failures, decapsulation behaviour, branch timing, memory han-

dling, serialization ambiguity and hardware leakage can undermine otherwise well-designed cryptography. The literature is clear that algorithmic security is insufficient when real systems leak information [Cintas Canto et al., 2023]. For pentesters, the challenge is to incorporate implementation-security thinking without overstating what a normal engagement can prove.

A full side-channel assessment is specialized. It may require oscilloscopes, probes, trigger logic, target boards, controlled firmware, statistical analysis and physical access. Most enterprise PQC pentests will not perform this work. However, every PQC pentest should classify whether side-channel risk matters. A cloud API endpoint, a payment terminal, an HSM, a smart card, an IoT device and a firmware update server have different exposure. A physically exposed device using ML-KEM may require stronger assurance evidence than a server-side library in a controlled data centre.

## 9.2. Timing, cache, power, and error channels

The tester can ask practical questions. Does the implementation claim constant-time behaviour? Is it a reference implementation, optimized implementation or vendor-modified fork? Has it undergone independent review? Are decapsulation failures handled uniformly? Is randomness generated by approved entropy sources? Are secrets zeroized? Are private keys stored in hardware? Are logs free of sensitive intermediate data? Are error messages indistinguishable to remote clients? Are there regression tests for malformed inputs? These questions often reveal more useful risk than pretending to break the algorithm.

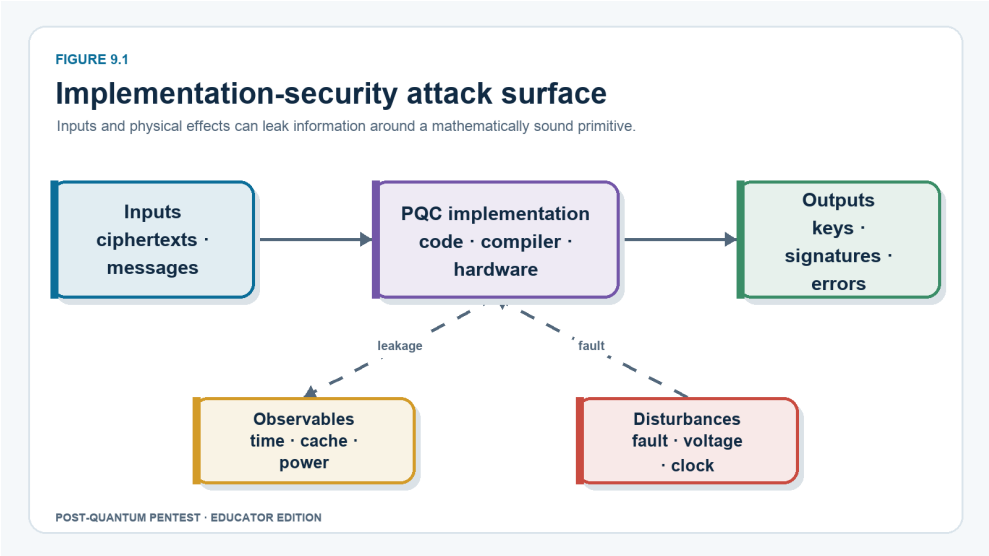


**Figure 9.1:** Implementation assurance model: inputs and faults can create observable leakage around an otherwise sound algorithm.

Fault thinking is also important. Active attacks may manipulate power, voltage, clocking, memory, ciphertexts, signatures or inputs to induce abnormal behaviour. In a lab, toy examples can demonstrate why failure handling matters. In production, the tester should not perform destructive fault tests without explicit authorization and a controlled device. But the pentest report can recommend assurance levels. For example, a consumer IoT device exposed to physical attackers may require side-channel-hardened implementations and vendor evidence. A back-office server may require code review, dependency assurance and configuration controls.

## 9.3. Randomness and failure handling

Randomness is a universal concern. PQC algorithms depend on high-quality randomness for key generation, encapsulation and signatures. Some signature schemes are particularly sensitive to nonce or sampling failures. The tester should inspect entropy source configuration, FIPS validation where required, container entropy behaviour, embedded-device boot-time entropy, virtualization constraints and failure monitoring. A PQC deployment with poor randomness may be worse than a delayed deployment with disciplined engineering.

Serialization and parsing are underestimated. Larger keys, signatures and ciphertexts create new parser paths. Certificates, protocol messages, APIs and hardware interfaces must handle new sizes and encodings. The tester should look for truncation, buffer assumptions, maximum-size limits, unsupported object identifiers, certificate-chain parsing failure, logging truncation and monitoring blind spots. The ECDHE-MLKEM draft's concrete key-share sizes are useful because they show that hybrid handshakes are measurably larger than classical equivalents [Kwiatkowski et al., 2026].

## 9.4. Triage before specialist laboratory work

Implementation-security findings should be written with humility. A tester can say "no evidence of side-channel hardening was provided," "vendor documentation does not specify constant-time properties," "decapsulation failure behaviour was not tested," or "the device is physically exposed and uses an unreviewed implementation." Those statements are defensible. A tester should

not claim that a PQC implementation is secure merely because a scan completed successfully. The right conclusion is assurance-based: what evidence exists, what is missing, and what level of assurance is appropriate for the asset.

## 9.5. Current practice and project connections

The timing laboratory compares an early-exit byte comparison with a constant-time comparison over repeated local measurements. It is deliberately noisy and does not recover a real secret. Its purpose is to teach experimental controls, distributions, false conclusions, and the difference between a demonstration and a product-level side-channel assessment.

Implementation assurance questions should include compiler and library provenance, constant-time testing, randomness health, zeroisation expectations, error handling, key reuse, fault response, hardware exposure, and the vendor's process for responding to new cryptanalytic or implementation findings.

## 9.6. Instructor lens

Teach this chapter by asking learners to separate observed evidence, interpretation, and forecast. Require every claim to name its asset, owner, security property, time horizon, and evidence source. A suggested 90-minute pattern is 25 minutes of concepts, 35 minutes of guided evidence work, 20 minutes of team interpretation, and a 10-minute debrief.

**Certification alignment.** CEH: evasion and system attack concepts; OPST/OPSA: operational analysis; PenTest+: attacks, exploits, and evidence quality.

## 9.7. Hands-on laboratory

Complete companion lab07-timing-side-channel in the labs directory. Begin with its rules of engagement, preserve the generated evidence, answer the interpretation questions, and compare the result with the educator solution only after submitting a first analysis. The laboratory is local-only by design.

## Review and discussion

1. What is the difference between constant-time intent and constant-time evidence?
2. Why can decapsulation failure handling be security critical?
3. Which device characteristics justify specialist side-channel testing?
4. State one limitation of the chapter's laboratory and propose a safe next test.

# Chapter 10

# PKI, Certificates, and Hybrid Identity

**Chapter overview:** This chapter develops the post-quantum pentesting knowledge and evidence practices needed to map certificate authorities, profiles, relying parties, and trust stores.

**Keywords:** post-quantum cryptography, penetration testing, cryptographic agility, evidence, education

After reading this chapter, you should be able to:

- Map certificate authorities, profiles, relying parties, and trust stores.
- Assess signature lifetimes and long-term validation requirements.
- Compare dual, composite, and staged signature transitions.
- Test verifier agility and recovery paths for long-lived devices.

## 10.1. Map the trust system, not only certificates

Public key infrastructure is one of the hardest parts of the post-quantum transition. Key establishment can be tested in a protocol lab, but identity depends on certificate authorities, certificate profiles, validation libraries, browsers, operating systems, devices, policies, hardware tokens, revocation, timestamping

and user trust. PQC pentesting must therefore treat PKI as an ecosystem rather than a single endpoint configuration.

The tester begins by inventorying certificate authorities, intermediates, leaf certificates, private CAs, certificate templates, issuance workflows, renewal automation, revocation methods, HSM storage and validation clients. Each certificate should be mapped to algorithm, key size, signature algorithm, subject, issuer, validity period, business process and owner. Classical public web certificates are expected today; the question is whether the organization understands where certificates are used and how migration would occur when ecosystems support it.

## 10.2. Long-term validation and archival authenticity

Hybrid certificate schemes are under active study. Comparative work on hybrid X.509 schemes discusses designs such as composite, catalyst and chameleon approaches, showing that certificate size, validation compatibility and migration feasibility must be evaluated [Chen, 2025a]. For pentesters, the important point is not to choose a universal winner. The important point is to test whether the organization's validation stack can handle larger or different certificate structures, whether applications reject unknown algorithms, and whether certificate management tools can store, renew and monitor future profiles.

Internal PKI may be more urgent than public PKI because organizations control it. If an enterprise issues certificates for VPNs, service meshes, devices, Wi-Fi, mTLS APIs or industrial systems, it can begin planning templates,

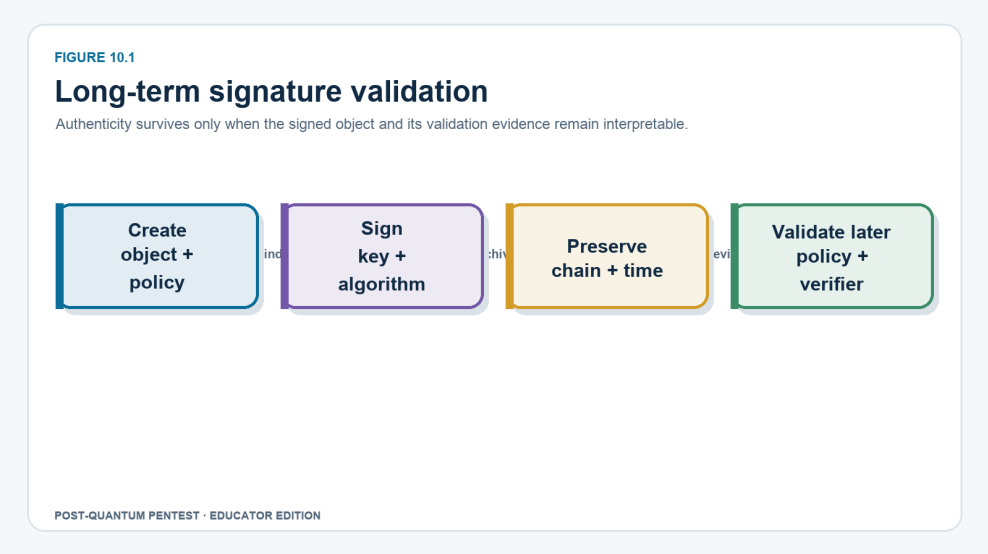


**Figure 10.1:** Signature lifecycle and the evidence required to preserve authenticity through algorithm transition.

agility and test environments. The tester should ask whether the CA software supports new algorithms, whether clients can validate them, whether certificate inventory is complete, and whether legacy devices block migration. A finding might state: "Private CA inventory is incomplete; certificates are issued with RSA-2048 for device identity; device fleet has ten-year field lifetime; no tested migration path exists." That is a strong PQC pentest finding even without a current CVE.

## 10.3. Hybrid and composite transition choices

Identity tokens require similar treatment. JWTs, SAML assertions, OAuth client assertions and signed webhooks often rely on RSA or ECDSA. Many tokens are short-lived, reducing long-term quantum risk, but the signing infrastructure itself may be strategically important. The tester should distinguish token lifetime from signing-key lifetime. A five-minute token signed by a ten-year RSA key is not a harvest-now-decrypt-later confidentiality issue, but it may be a future authenticity and key-governance issue.

Timestamping and long-term validation are essential for legal and evidentiary workflows. If a signature must prove that a document existed and was signed at a time when the algorithm was trusted, trusted timestamping and archival validation become migration controls. A PQC pentest should identify whether such controls exist, whether they depend on classical algorithms, and how they will be preserved. This is especially important in government, finance, health, academia and legal sectors.

## 10.4. Firmware, document, and software signing

Reporting for PKI should avoid premature demands. The tester should not tell every client to deploy post-quantum certificates immediately in production if the ecosystem is not ready. Instead, the report should recommend inventory, test CA, validation experiments, vendor roadmaps, certificate-management up-

grades, policy agility, and a staged path for high-risk internal identities. PQC PKI migration is a programme, not a switch.

## 10.5. Current practice and project connections

Long-term authenticity is a system property. The signed object, algorithm identifiers, certificate path, revocation evidence, trusted time, validation policy, and verifier implementation all contribute. A migration plan that names only the new signature algorithm omits most of the trust system.

For constrained products, the decisive question is often verifier agility. If a boot ROM recognises only one classical signature format, a twelve-year product may require hardware replacement or a carefully designed bridge. This risk belongs in product safety and lifecycle governance, not only in the cryptography appendix.

## 10.6. Instructor lens

Teach this chapter by asking learners to separate observed evidence, interpretation, and forecast. Require every claim to name its asset, owner, security property, time horizon, and evidence source. A suggested 90-minute pattern is 25 minutes of concepts, 35 minutes of guided evidence work, 20 minutes of team interpretation, and a 10-minute debrief.

**Certification alignment.** CEH: PKI and cryptography; OPSA: trust analysis; PenTest+: certificate management, key rotation, and remediation.

## 10.7. Hands-on laboratory

Complete companion lab08-signature-lifecycle in the labs directory. Begin with its rules of engagement, preserve the generated evidence, answer the interpretation questions, and compare the result with the educator solution only after submitting a first analysis. The laboratory is local-only by design.

## Review and discussion

1. Why is replacing a CA signing algorithm insufficient if relying parties cannot validate it?
2. What does a trusted timestamp contribute to long-term validation?
3. How would you test a field device whose boot verifier cannot be replaced?
4. State one limitation of the chapter's laboratory and propose a safe next test.

# Chapter 11

# QKD, Quantum Communication, and Hybrid Assurance

**Chapter overview:** This chapter develops the post-quantum pentesting knowledge and evidence practices needed to differentiate post-quantum cryptography from quantum key distribution.

**Keywords:** post-quantum cryptography, penetration testing, cryptographic agility, evidence, education

After reading this chapter, you should be able to:

- Differentiate post-quantum cryptography from quantum key distribution.
- Identify QKD endpoint, trusted-node, authentication, and key-management risks.
- Place Lithuania's QCI activity within the EuroQCI architecture.
- Design an assessment that treats QKD as one control in a larger system.

## 11.1. Keep QKD and PQC claims distinct

Post-quantum cryptography and quantum key distribution are often confused in public discussion. PQC uses classical algorithms believed to resist quantum

attacks. QKD uses quantum communication properties to establish keys under specific physical and operational assumptions. They solve related but different problems, and both can appear in quantum-safe architectures. A post-quantum pentest book should include QKD because critical infrastructure and EuroQCI-style projects may combine QKD, KMS, PQC and classical networks.

A QKD-aware pentest does not attempt to "break quantum physics." It assesses the system around the quantum link. This includes the optical path, trusted nodes, key management system, authentication channels, classical control channels, device configuration, monitoring, physical security, operational procedures, availability, key consumption, key-buffer policy, incident response and integration with applications. The weakest point may be the classical management interface, not the quantum channel.

## 11.2. Trusted nodes, endpoints, and key management

Hybrid PQC/QKD research explores combining mathematically based PQC with physically grounded QKD. Recent work proposes hybrid key exchange and signature ideas that combine NIST-standardized PQC algorithms with QKD protocols [Chen, 2025b]. Other work investigates QKD-KEM integration into TLS using OpenSSL providers and ETSI API concepts [Blanco-Romero et al., 2025]. These ideas are relevant to ResilQ because they show how future infrastructure may mix multiple trust models. A pentest methodology must be ready to test integration boundaries.

In the author's broader ecosystem, Established Quantum Link thinking is

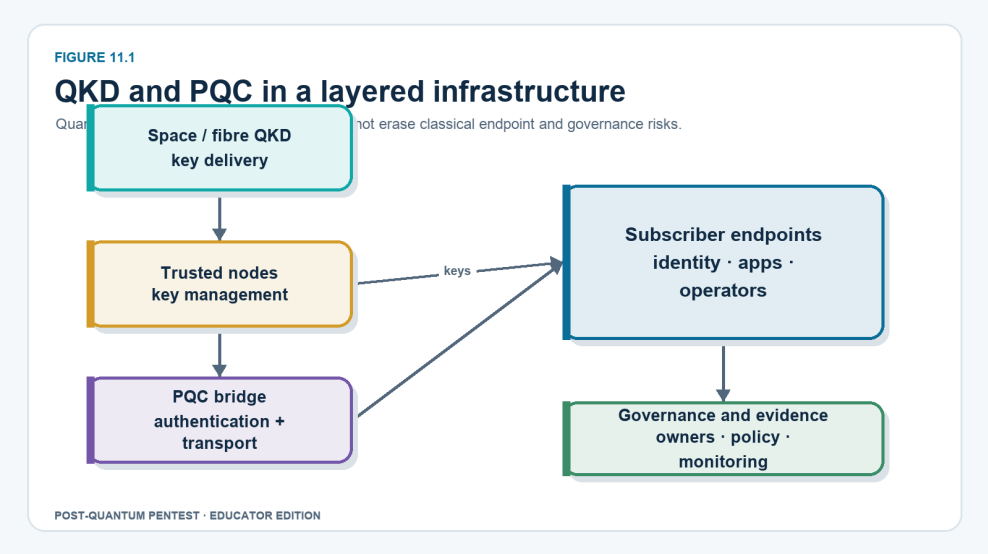


**Figure 11.1:** Layered QKD and PQC assurance: optical key delivery does not remove endpoint, authentication, or governance obligations.

useful. An EQL is not merely a fibre connection. It is a time-bounded, authenticated, calibrated, monitored and policy-compliant QKD service relationship with defined classical and quantum channels, KMS state, key-pool constraints, hybrid PQC contribution and evidence lifecycle. A tester can assess whether the link has sufficient secure-key rate for the intended application, whether key consumption exceeds reserve, whether trusted nodes are governed, and whether monitoring detects degradation.

## 11.3. The Lithuanian and European QCI context

QKD also creates availability and operational risks. Fibre attenuation, temperature changes, vibration, device calibration, trusted-node throughput, KMS integration and key buffer exhaustion can all affect service. A post-quantum pentest should therefore include resilience testing: what happens when key rate drops, when a trusted node fails, when the application consumes keys too quickly, when the QKD service is unavailable, or when policy requires fallback to PQC-only? The test should verify that fallback does not silently reduce assurance without logging and governance.

ResilQ can represent QKD and PQC as layered controls. Transit may include TLS with hybrid ML-KEM and QKD-derived key material. Use may include applications that request keys through a KMS. Rest may include key-wrapping policies for archives. QARS can score the residual risk, while RQmod can declare backend profiles for OQS, QKD APIs or SpinQ demonstrations. The important principle is evidence: the report should show which keys were requested, which policy was applied, what rate was observed, what fallback occurred and what data was protected.

## 11.4. Assurance for hybrid infrastructures

QKD material in this book should be realistic. It should explain constraints, not sell miracles. QKD can add strong properties in specific architectures, but it does not replace authentication, endpoint security, application hardening, key governance or PQC migration. A QKD system with weak management

credentials is still vulnerable. A QKD link without a migration plan for authentication is incomplete. A hybrid architecture is strongest when each layer is tested according to its own assumptions.

## 11.5. Current practice and project connections

QKD distributes key material using quantum-physical mechanisms; PQC uses algorithms designed to resist both classical and quantum adversaries on ordinary computing platforms. They can complement one another, but neither term should be used as a blanket adjective for an entire system. Endpoints, authentication, key stores, management planes, trusted nodes, applications, and operators remain testable attack surfaces.

The project https://qci.lt/ presents Lithuania's role in EuroQCI, including terrestrial and space-segment context, cross-border projects, and a PQC bridge concept. The European Commission describes EuroQCI as an EU-wide infrastructure with terrestrial fibre and space segments. Use the site as a teaching map and validate material claims against primary project and Commission sources [European Commission, 2026].

The Lithuanian project https://www.pqc.lt/ and QCI portal should be taught together: one focuses on the software and governance transition to post-quantum cryptography, while the other makes quantum communication infrastructure visible. Their overlap is key management, identity, interoperability, and national resilience.

## 11.6. Instructor lens

Teach this chapter by asking learners to separate observed evidence, interpretation, and forecast. Require every claim to name its asset, owner, security property, time horizon, and evidence source. A suggested 90-minute pattern is 25 minutes of concepts, 35 minutes of guided evidence work, 20 minutes of team interpretation, and a 10-minute debrief.

**Certification alignment.** CEH: network architecture and controls; OPST/OPSA: physical and communications channels; PenTest+: engagement scoping and network analysis.

## 11.7. Hands-on laboratory

Complete companion lab11-qci-architecture in the labs directory. Begin with its rules of engagement, preserve the generated evidence, answer the interpretation questions, and compare the result with the educator solution only after submitting a first analysis. The laboratory is local-only by design.

## Review and discussion

1. Which parts of a QKD system remain classical attack surfaces?
2. Why does QKD still need authenticated classical communication?
3. What should a report say when testing claims that combine QKD and PQC?
4. State one limitation of the chapter's laboratory and propose a safe next test.

# Chapter 12

# Reporting, Scoring, and Executive Communication

**Chapter overview:** This chapter develops the post-quantum pentesting knowledge and evidence practices needed to write findings that separate observation from time-based inference.

**Keywords:** post-quantum cryptography, penetration testing, cryptographic agility, evidence, education

After reading this chapter, you should be able to:

- Write findings that separate observation from time-based inference.
- Use risk scoring without hiding judgement behind a number.
- Produce technical and executive views from the same evidence.
- Define closure criteria that can be independently retested.

## 12.1. Observed fact, inference, and forecast

A post-quantum pentest report must serve multiple audiences. Engineers need precise evidence. CISOs need prioritized risk and control ownership. Executives need a clear decision narrative. Regulators may need traceability. Procurement teams need vendor questions. Developers need actionable remediation. A single vulnerability list cannot satisfy these needs. The report should be structured

around PAREX phases and QARS scoring.

The executive summary should begin with the business question. For example: "The assessment evaluated whether payment, identity and archive systems are prepared for the transition from quantum-vulnerable public-key cryptography to NIST-standardized PQC." It should then provide the most important facts: number of cryptographic assets identified, number of high-risk quantum-vulnerable exposures, presence or absence of CBOM, state of TLS and PKI migration, availability of vendor roadmaps, and top three remediation actions. The summary should not overdramatize quantum timelines. It should say that migration takes time and that high-risk long-lived data should be prioritized now.

## 12.2. QARS and Quantum-Safe Index context

The technical findings should be evidence-rich. Each finding should include title, affected assets, observed algorithm, security property, evidence source, data sensitivity, data lifetime, exposure, QARS score, likelihood rationale, impact rationale, remediation, owner and validation method. A good finding is testable. If remediation occurs, the team should be able to rerun the ResilQ job and see the evidence change.

QARS scoring should be transparent. The report should explain timeline, sensitivity and exposure factors. It should avoid false precision. A score is a decision aid, not an oracle. The best practice is to show both numeric and categorical outputs, along with the evidence behind them. For example: "QARS 84,

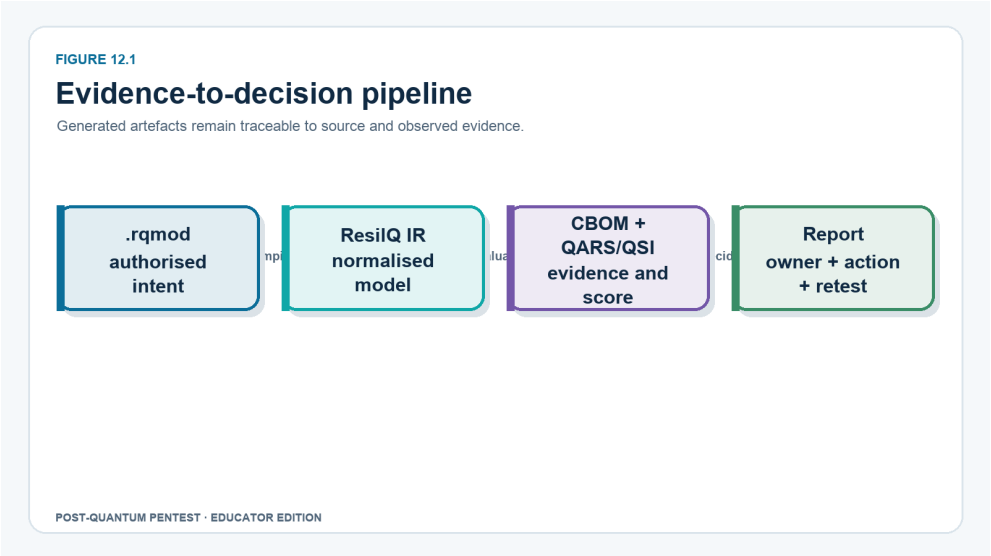


**Figure 12.1:** Evidence pipeline from RQmod source through normalised findings, CBOM, risk scoring, and decision-ready reporting.

High, because the archive contains regulated health data with a twenty-year confidentiality horizon, is replicated to vendor storage, and uses RSA-based key wrapping." This gives a board enough context to fund remediation.

## 12.3. From finding to decision

The report should include a migration roadmap. Immediate actions may include inventory completion, certificate template review, vendor evidence requests, disabling obsolete algorithms, creating a PQC test environment, adding CBOM generation to CI/CD, and classifying data lifetimes. Medium-term actions may include hybrid TLS pilots, internal PKI agility testing, signing-pipeline upgrades, archive key-wrapping redesign and procurement language updates. Long-term actions include policy integration, continuous monitoring,

tabletop exercises, production PQC rollout and periodic reassessment.

DORA, NIS2, GDPR and sector rules should be mapped carefully. The report should not claim that PQC alone satisfies regulatory obligations. Instead, it should show how PQC migration supports resilience, risk management, data protection, supplier governance, incident readiness and secure development. For a financial entity, DORA language may focus on ICT risk management, third-party risk and resilience testing. For critical infrastructure, NIS2 language may focus on risk-management measures and continuity. For personal data, GDPR language may focus on confidentiality over the data lifecycle.

## 12.4. Retest criteria and evidence retention

The report should also include vendor questions. "Does the product use RSA, DH, ECDH, ECDSA or DSA?" "Can you provide a CBOM?" "Which NIST PQC standards are supported?" "Is hybrid TLS supported?" "What is the roadmap for signatures?" "How are keys generated and stored?" "Has the implementation been reviewed for side channels?" "Can algorithms be updated without hardware replacement?" These questions turn pentest findings into procurement control.

Finally, the report should include an exercise plan. A quantum-risk tabletop might ask what the organization would do if a regulator requested a cryptographic inventory, if a vendor announced a vulnerable PQC implementation, if a high-risk archive had to be rewrapped, or if hybrid TLS caused availability issues. Exercises make the transition operational. They also reveal whether

ownership is real.

## 12.5. Current practice and project connections

The Quantum-Safe Index (https://quantumsafeindex.com/) is a transparent PAREK-aligned assessment instrument. Its published model reports six weighted dimensions - inventory, exposure, migration, agility, validation, and governance - with critical gates and evidence confidence. It explicitly describes the result as an assessment, not an authority certification. That distinction should be preserved in reports.

ResilQ Studio and RQmod (https://rqmod.com/) model post-quantum risk as text and compile it into evidence-oriented artefacts such as normalised IR, audit jobs, QARS-NI inputs, CBOM skeletons, and optional bounded QASM demonstrations. A report should still retain the raw source and disclose which outputs are generated, observed, or inferred.

## 12.6. Instructor lens

Teach this chapter by asking learners to separate observed evidence, interpretation, and forecast. Require every claim to name its asset, owner, security property, time horizon, and evidence source. A suggested 90-minute pattern is 25 minutes of concepts, 35 minutes of guided evidence work, 20 minutes of team interpretation, and a 10-minute debrief.

**Certification alignment.** CEH: vulnerability reporting; OPSA: STAR-

style analysis; PenTest+: engagement reporting and remediation.

## 12.7. Hands-on laboratory

Complete companion lab10-reporting in the labs directory. Begin with its rules of engagement, preserve the generated evidence, answer the interpretation questions, and compare the result with the educator solution only after submitting a first analysis. The laboratory is local-only by design.

## Review and discussion

1. Which statement in a finding is directly observed and which depends on a forecast?
2. Why must a QSI or QARS score remain explainable at the control level?
3. Write one measurable closure criterion for a missing cryptographic inventory finding.
4. State one limitation of the chapter's laboratory and propose a safe next test.

# Chapter 13

# Practical CTF and Training Scenarios

**Chapter overview:** This chapter develops the post-quantum pentesting knowledge and evidence practices needed to build a safe challenge with a story, evidence target, and scoring rubric.

**Keywords:** post-quantum cryptography, penetration testing, cryptographic agility, evidence, education

After reading this chapter, you should be able to:

- Build a safe challenge with a story, evidence target, and scoring rubric.
- Reward scoping, evidence quality, and remediation - not only flag capture.
- Use PQC demonstrations without claiming unrealistic cryptanalytic capability.
- Run a debrief that converts challenge activity into professional practice.

## 13.1. Teach decisions, not tool theatre

Training is essential because post-quantum security is conceptually difficult and operationally new. A CTF-style environment can teach testers how to discover cryptographic assets, classify risk, test hybrid protocols and write evidence-based findings. The objective should be competence, not spectacle. Each challenge should include a story, authorized target, evidence artefact,

scoring rubric and debrief.

Challenge 1 is "Find the Future Breach." Participants receive a small web application, TLS endpoint and archive. The TLS scan looks acceptable in classical terms, but the archive contains long-lived regulated data wrapped with RSA. The winning team identifies the mismatch between current strength and future confidentiality. The learning point is that harvest-now-decrypt-later is about data lifetime and exposure.

## 13.2. Challenge patterns and evidence flags

Challenge 2 is "The False PQC Badge." Participants receive a service that claims to be PQC ready. In the lab, the service supports an OQS-enabled endpoint, but production configuration always negotiates classical ECDHE. Participants must prove which clients receive hybrid key exchange and which do not. The learning point is that marketing claims must be tested at the handshake level.

Challenge 3 is "The Hidden Signer." A repository contains JWT signing, container signing and firmware signing. Participants generate a CBOM and classify RSA and ECDSA usage. Some findings are low priority because tokens expire quickly. Others are high priority because firmware signatures must remain verifiable for years. The learning point is that authenticity risk depends on context.

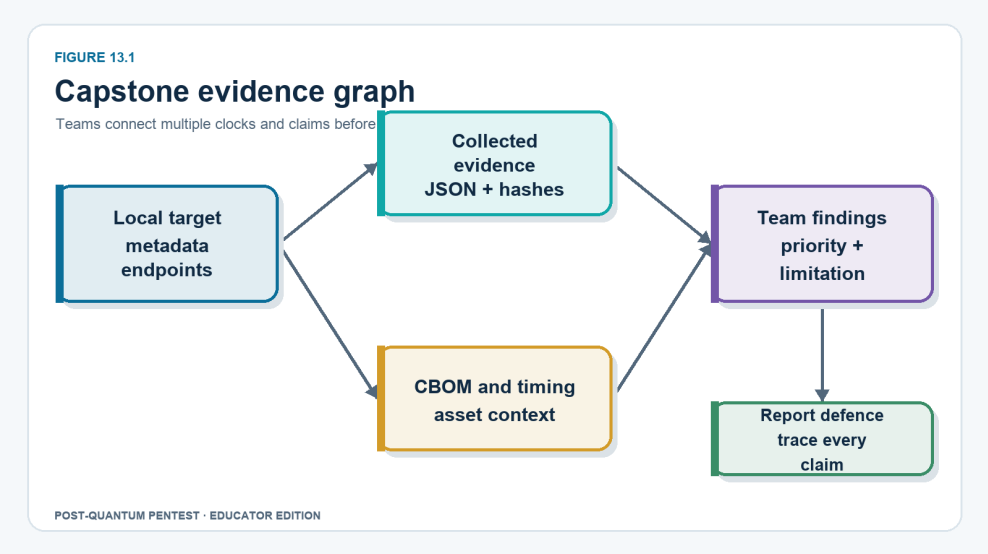


**Figure 13.1:** Capstone laboratory evidence graph and team workflow.

## 13.3. Team roles and safe competition

Challenge 4 is "The Broken Migration." A hybrid TLS configuration causes intermittent failures because a middlebox rejects larger handshakes. Participants identify the failure, document evidence and recommend a staged rollout. The learning point is that PQC migration is also an availability and interoperability project.

Challenge 5 is "QARS Board Brief." Teams receive raw findings and must produce a one-page executive summary with QARS scoring. The winner is not the team with the longest report, but the team that explains why the top three findings matter and what decision is needed. The learning point is that technical evidence must translate into governance.

## 13.4. Debriefing and transfer to the workplace

A SpinQ demonstration can be integrated as a leadership module. The trainer runs a simple two-qubit experiment, then connects it to the idea that quantum computation changes assumptions. Immediately afterward, participants run a ResilQ job that scores a classical cryptographic exposure. This avoids the common error of pretending that the demo machine breaks encryption. It uses the demo to create intuition, then uses ResilQ to create actionable security understanding.

The training environment should include flags that reward good ethics. For example, a team loses points for scanning outside scope or for extracting private keys not needed for the challenge. This reinforces the principle that post-quantum pentesting is authorized, evidence-based and controlled. The Qacker mindset is useful for threat modelling, not for careless behaviour.

## 13.5. Current practice and project connections

The capstone deliberately includes findings with different clocks: a short-lived token, a long-lived archive, a firmware signer, and an unverified hybrid claim. Teams receive more credit for prioritisation and evidence quality than for the number of strings they find. This mirrors professional work, where a long list of uncontextualised algorithms can distract from the few dependencies that determine strategic risk.

## 13.6. Instructor lens

Teach this chapter by asking learners to separate observed evidence, interpretation, and forecast. Require every claim to name its asset, owner, security property, time horizon, and evidence source. A suggested 90-minute pattern is 25 minutes of concepts, 35 minutes of guided evidence work, 20 minutes of team interpretation, and a 10-minute debrief.

**Certification alignment.** CEH Practical: task execution; OPST/OPSA: operational reasoning; PenTest+: performance-based integration across all domains.

## 13.7. Hands-on laboratory

Complete companion lab12-capstone in the labs directory. Begin with its rules of engagement, preserve the generated evidence, answer the interpretation questions, and compare the result with the educator solution only after submitting a first analysis. The laboratory is local-only by design.

## Review and discussion

1. What behaviour should lose points even when a team captures a flag?
2. How can a challenge assess reporting rather than only enumeration?
3. Why should a quantum-computing demonstration be separated from a cryptographic break claim?
4. State one limitation of the chapter's laboratory and propose a safe next test.

# Chapter 14

# Ethics, Legal Boundaries, and Professional Responsibility

**Chapter overview:** This chapter develops the post-quantum pentesting knowledge and evidence practices needed to apply authorisation and minimisation to cryptographic evidence handling.

**Keywords:** post-quantum cryptography, penetration testing, cryptographic agility, evidence, education

After reading this chapter, you should be able to:

- Apply authorisation and minimisation to cryptographic evidence handling.
- Recognise misleading or exaggerated quantum-security claims.
- Define safe escalation paths for implementation testing.
- Use the handbook alongside, but not as a substitute for, official certification objectives.

## 14.1. Authorisation, minimisation, and accuracy

Post-quantum pentesting operates in a sensitive space. Cryptography protects privacy, national security, finance, health, identity and critical infrastructure. Testing it requires discipline. The first ethical rule is authorization. Every scan,

code review, protocol experiment, device test and key-handling activity must be within written scope. The second rule is minimization. The tester should collect enough evidence to prove the finding but avoid unnecessary exposure of secrets, private keys, personal data or sensitive architecture. The third rule is accuracy. Quantum risk is serious, but exaggeration harms trust.

A tester should never claim that a small educational quantum computer can break RSA, or that every encrypted record is already compromised, or that buying a single product solves post-quantum migration. The correct message is more nuanced. Cryptographically relevant quantum computers are not generally available today, but migration takes years, and long-lived data faces harvest-now-decrypt-later risk. NIST standards are ready for use in appropriate contexts, but implementation, interoperability and governance remain hard. PQC reduces quantum risk, but it does not replace access control, endpoint security, monitoring, incident response or supplier management.

## 14.2. Avoid quantum fear, uncertainty, and doubt

The tester must also be careful with implementation-attack knowledge. Discussing side channels and faults is necessary for assurance, but detailed offensive procedures against real products can cause harm. This book uses implementation attacks to motivate secure engineering and test planning. Any physical or fault assessment should be a separate authorized engagement with safe hardware, vendor coordination where appropriate and strong evidence controls.

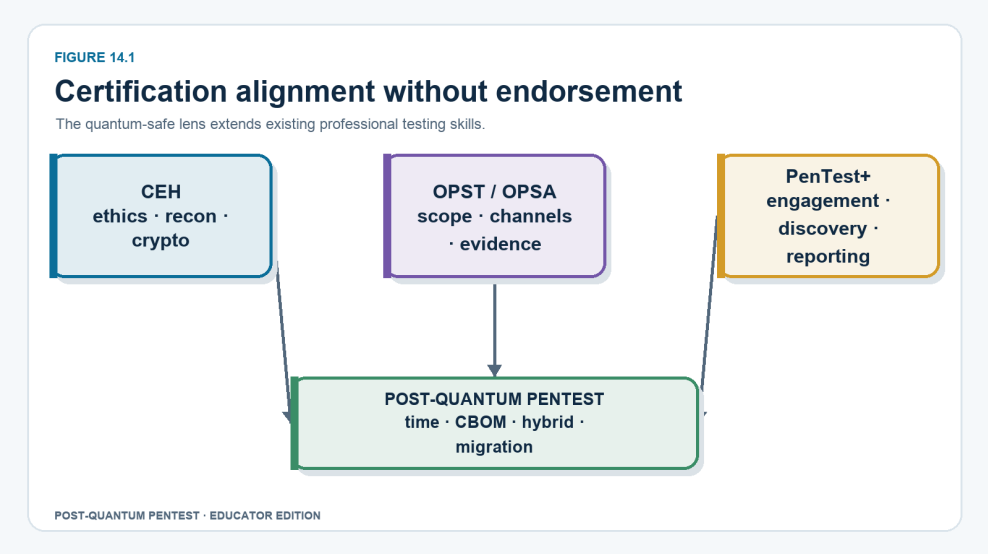


**Figure 14.1:** Certification and educator alignment across engagement, discovery, testing, analysis, and reporting.

Professional responsibility includes uncertainty management. Timelines for cryptographically relevant quantum computers are uncertain. Standards evolve. Protocol drafts change. Vendor claims vary. A report should state assumptions and confidence. It should separate observed evidence from projections. It should recommend monitoring standards bodies, NIST updates, IETF progress, vendor advisories and sector guidance. Good PQC pentesting is not a one-time prophecy; it is a recurring assurance process.

## 14.3. Responsible implementation-security work

Finally, post-quantum transition is a public-good problem. The systems that need migration include schools, hospitals, small businesses, municipalities and critical services. A book by Prof. Dr. Šarūnas Grigaliūnas should therefore balance academic rigour with social mission. The goal is not to make quantum security an elite topic. The goal is to make it testable, understandable and actionable for the organizations that protect society.

## 14.4. Certification alignment and professional limits

## 14.5. Current practice and project connections

This handbook can supplement Certified Ethical Hacker study, ISECOM OPST/OPSA preparation, and CompTIA PenTest+ preparation. It is not official courseware, an exam dump, or a guarantee of certification success. CEH is an EC-Council credential, OPST and OPSA are ISECOM credentials, and PenTest+ is a CompTIA credential. Educators and candidates should always use the current official blueprint because exam domains and weights change.

The closest PenTest+ PT0-003 alignments are engagement management, reconnaissance and enumeration, vulnerability discovery and analysis, attacks

and exploits, and post-exploitation/lateral movement. The handbook adds a quantum-era cryptographic lens to those skills. CEH alignment is strongest in ethical hacking foundations, reconnaissance, scanning, enumeration, vulnerability analysis, cryptography, and reporting. OPST/OPSA alignment is strongest in operational scope, channel testing, evidence, metrics, and analysis [CompTIA, 2024, EC-Council, 2024, ISECOM, 2026].

## 14.6. Instructor lens

Teach this chapter by asking learners to separate observed evidence, interpretation, and forecast. Require every claim to name its asset, owner, security property, time horizon, and evidence source. A suggested 90-minute pattern is 25 minutes of concepts, 35 minutes of guided evidence work, 20 minutes of team interpretation, and a 10-minute debrief.

**Certification alignment.** Crosswalk chapter for CEH, OPST/OPSA, and CompTIA PenTest+; no affiliation or endorsement is implied.

## 14.7. Hands-on laboratory

Complete companion lab01-scope-and-ethics in the labs directory. Begin with its rules of engagement, preserve the generated evidence, answer the interpretation questions, and compare the result with the educator solution only after submitting a first analysis. The laboratory is local-only by design.

# Review and discussion

1. What is the smallest evidence set that proves a finding without collecting secret material?
2. Rewrite an exaggerated quantum claim as a bounded, evidence-based statement.
3. When must an educator stop a practical exercise?
4. State one limitation of the chapter's laboratory and propose a safe next test.

# Chapter 15

# From Book to Field Programme

**Chapter overview:** This chapter develops the post-quantum pentesting knowledge and evidence practices needed to deliver the handbook as a fourteen-week course or intensive boot camp.

**Keywords:** post-quantum cryptography, penetration testing, cryptographic agility, evidence, education

After reading this chapter, you should be able to:

- Deliver the handbook as a fourteen-week course or intensive boot camp.
- Integrate Qmacs, RQmod, PassQ, PQC.lt, QCI.lt, and the Quantum-Safe Index responsibly.
- Translate laboratory evidence into repeatable field-service packages.
- Define a research and reassessment loop for a fast-moving standards landscape.

## 15.1. A fourteen-week educator pathway

A book is useful only if it changes practice. The field programme proposed here turns Post-Quantum Pentest into a repeatable service, training path and

research agenda. The service path begins with a two-week rapid assessment for organizations that have no cryptographic inventory. The output is a first CBOM, high-level QARS map and executive briefing. The training path uses CTF labs, OQS demos and ResilQ scoring to teach practitioners how to test without exaggerating. The research path investigates gaps that current standards and tools do not yet solve: hybrid certificate validation, PQC in constrained devices, QKD/PQC key-management integration, implementation assurance evidence and sector-specific QARS calibration.

The first service package is PQC Exposure Discovery. It focuses on Transit, Use and Rest evidence. For Transit, the tester scans authorized TLS, VPN and SSH endpoints. For Use, the tester reviews repositories, CI/CD signing and token issuance. For Rest, the tester maps archives, backups, database encryption and KMS dependencies. The deliverable is not a full migration project. It is a clear map of where quantum-vulnerable cryptography exists and why it matters.

The second service package is Hybrid Readiness Testing. It uses a staging environment to test OQS or vendor-supported hybrid protocols, certificate handling, middlebox compatibility, performance, logging and rollback. The deliverable is a pilot report. This is valuable because many organizations will not fail because they chose the wrong algorithm. They will fail because their operational environment cannot handle the transition without planned testing.

## 15.2. Tools that preserve models and secrets

The third service package is Quantum Governance and Board Readiness. It translates CBOM and QARS findings into policy, procurement and regulatory language. It includes vendor questionnaires, control mapping, roadmap, tabletop exercise and board presentation. This package is especially important for financial entities, public-sector bodies, critical infrastructure, health systems and universities, where leadership must make funding decisions before the threat becomes concrete.

The fourth service package is Implementation Assurance Triage. It does not replace a specialist lab, but it determines whether specialist analysis is needed. The tester reviews library provenance, implementation claims, randomness, error handling, hardware exposure, vendor evidence and device lifecycle. The output is an assurance gap report. For a server-side application, the recommendation may be dependency review and regression testing. For a smart card or IoT device, the recommendation may be side-channel laboratory evaluation.

ResilQ can support all four packages by preserving evidence. A client should be able to see that the same target was assessed over time, that findings changed because controls changed, and that risk scores remain linked to data classes and cryptographic assets. This matters for credibility. Post-quantum readiness will become a market claim. Evidence will distinguish serious programmes from slogans.

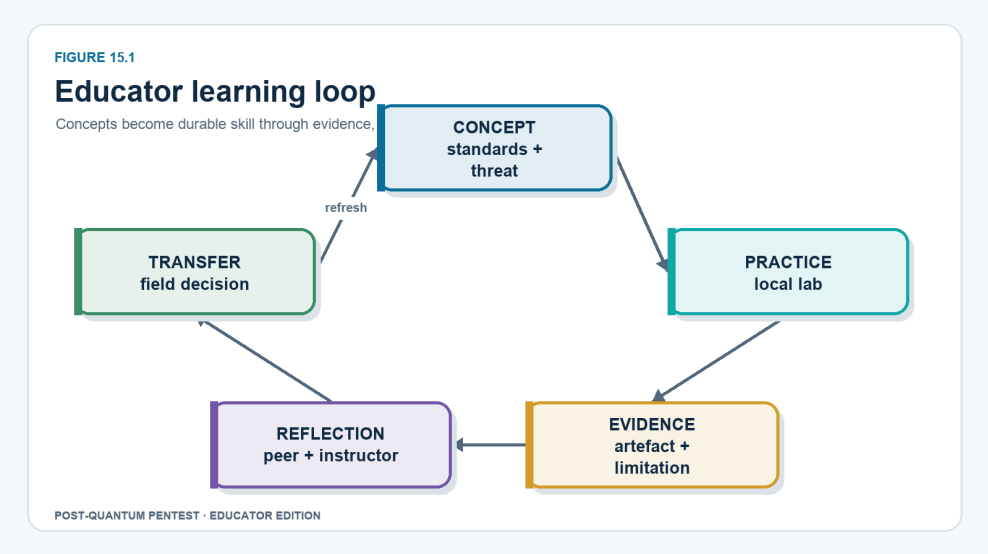


**Figure 15.1:** Educator learning loop from concept and laboratory evidence to assessment, reflection, and field transfer.

## 15.3. Lithuanian projects and public infrastructure

The book should also support education. Each chapter can have a companion lab. Chapter 1 gets a threat-timeline worksheet. Chapter 2 gets a publication-analysis exercise. Chapter 3 gets a PAREX scoping template. Chapter 4 gets a CBOM lab. Chapter 6 gets hybrid TLS packet analysis. Chapter 7 gets source-code discovery. Chapter 8 gets archive key-wrapping analysis. Chapter 10 gets PKI migration mapping. Chapter 12 gets report writing. This structure turns the manuscript into a course.

The research agenda is broad. We need better measurement of PQC migration cost by sector. We need test methods for cryptographic agility. We need standard CBOM semantics for post-quantum evidence. We need safe CTFs

that teach implementation-security concepts without encouraging misuse. We need frameworks for combining PQC and QKD without hand-waving. We need formal links between risk scoring and regulatory controls. We need case studies from real migrations, including failures. These are the problems that will make Post-Quantum Pentest more than a book title.

The final message is pragmatic. The quantum threat should not paralyze organizations, and it should not be sold as a miracle-product opportunity. It should be treated like a difficult engineering transition with strategic security consequences. That is exactly the kind of problem penetration testers, CISOs and researchers can help solve when they work with evidence, ethics and method.

## 15.4. Service patterns, research, and continuous improvement

## 15.5. Current practice and project connections

A fourteen-week delivery can use one chapter per week for Weeks 1-12, a team capstone in Week 13, and report defence in Week 14. A five-day intensive can combine foundations and scope on Day 1, discovery on Day 2, Transit/Use/Rest on Day 3, assurance and QCI on Day 4, and capstone reporting on Day 5. In either format, at least half of assessed marks should come from evidence interpretation and communication rather than command execution.

Recommended project sequence: use Qmacs for protected notes and RQ-mod editing; use the companion scripts to generate local evidence; use the Quantum-Safe Index for a transparent maturity discussion; examine PassQ as a layered vault case study; consult PQC.lt for the Lithuanian migration pathway; and use QCI.lt to situate software migration alongside EuroQCI. Each tool has a bounded role, and none replaces independent security assurance.

The field programme should be versioned. At the start of each academic year, verify NIST publication status, IETF protocol status, EU and Lithuanian timelines, tool releases, and official certification blueprints. Record the cut-off date in the syllabus and preserve earlier lab fixtures so that students can reproduce historical results.

## 15.6. Instructor lens

Teach this chapter by asking learners to separate observed evidence, interpretation, and forecast. Require every claim to name its asset, owner, security property, time horizon, and evidence source. A suggested 90-minute pattern is 25 minutes of concepts, 35 minutes of guided evidence work, 20 minutes of team interpretation, and a 10-minute debrief.

**Certification alignment.** Integrated revision for CEH, OPST/OPSA, and PenTest+; use the current official blueprint for exam-specific study.

## 15.7. Hands-on laboratory

Complete companion lab12-capstone in the labs directory. Begin with its rules of engagement, preserve the generated evidence, answer the interpretation

questions, and compare the result with the educator solution only after submitting a first analysis. The laboratory is local-only by design.

# Review and discussion

1. Which parts of the course should be updated every academic year?
2. How can students prove learning without exposing live systems?
3. What evidence would justify moving a pilot into a production migration programme?
4. State one limitation of the chapter's laboratory and propose a safe next test.

# Appendix A

# PAREX Field Checklist

Prepare. Confirm written authorization, systems in scope, test windows, data classes, contact paths, safety limits, evidence handling and reporting format. Identify the quantum risk question: long-term confidentiality, authenticity, supplier readiness, regulatory readiness, product assurance, critical infrastructure resilience or executive awareness.

Asset-discover. Enumerate TLS endpoints, VPNs, SSH services, APIs, certificates, private CAs, code repositories, signing pipelines, containers, firmware, device identities, databases, backups, archives, secrets vaults, KMS policies and vendor products. Produce a CBOM with algorithms, parameters, protocols, libraries, keys, certificates, owners and data context.

Risk-evaluate. Score each exposure using QARS dimensions: timeline, sensitivity and exposure. Add migration complexity, owner readiness and compensating controls. Distinguish confidentiality from authenticity. Distinguish public exposure from internal exposure. Distinguish short-lived tokens from long-term archives.

Evidence-test. Validate claims through authorized scans, packet captures, certificate parsing, source-code review, dependency analysis, OQS lab tests, hybrid TLS negotiation, downgrade checks, performance observations, configuration review, implementation-security questions and vendor evidence. Keep

evidence reproducible.

eXecute transition. Provide roadmap, owners, timeframes, quick wins, pilot plan, vendor questions, procurement language, CI/CD controls, CBOM maintenance, policy updates and tabletop exercises. Define how remediation will be retested.

## A.1. Minimum engagement record

Record the authorising party, systems and networks in scope, excluded systems, permitted techniques, test window, stop conditions, evidence handling, retention, emergency contacts, and reporting audience.

## A.2. Minimum CBOM fields

Record asset identifier, business owner, technical owner, data class, security property, algorithm, parameter set, protocol or format, library or provider, key location, certificate lifetime, retention horizon, exposure, migration dependency, evidence reference, and last verified date.

# Appendix B
# Sample Findings and Evidence Language

Finding 1: RSA key wrapping protects long-lived archive keys. Evidence: archive key-management configuration shows RSA-2048 wrapping for data encryption keys; archive contains regulated records with twenty-year confidentiality horizon; copies are replicated to vendor-managed storage. QARS: High. Recommendation: prioritize new archive key wrapping architecture, request vendor PQC roadmap, design staged rewrapping plan, and maintain evidence in CBOM.

Finding 2: Public API supports only classical ECDHE and classical certificate chain. Evidence: TLS scan shows TLS 1.3 with X25519 and ECDSA chain; no hybrid group observed in staging. Data lifetime: transaction data retained for seven years. QARS: Medium to High depending on data class. Recommendation: create hybrid TLS staging test, monitor IETF ECDHE-MLKEM progress, evaluate middlebox compatibility and align public rollout with ecosystem support.

Finding 3: Firmware signing uses ECDSA with field-device lifetime of twelve years. Evidence: build pipeline signs firmware with ECDSA P-256; devices verify signatures locally; no evidence of verifier agility. QARS: High for authenticity and product safety. Recommendation: assess bootloader update feasibility,

design hybrid or PQ signature transition, implement trusted timestamping and require vendor validation plan.

Finding 4: CBOM process is absent. Evidence: no inventory of algorithms, certificates, libraries or key owners exists beyond ad hoc TLS scans. QARS: Programme risk. Recommendation: adopt ResilQ/RQmod inventory jobs, integrate CBOM generation into CI/CD, assign cryptographic asset owners and repeat quarterly.

## B.1. Finding quality pattern

Use the order: claim, observed evidence, affected asset and owner, security property, time horizon, exploitation or failure condition, business impact, confidence and limitations, recommendation, and measurable retest criteria.

# Appendix C
# Glossary and Teaching Vocabulary

CBOM: Cryptographic Bill of Materials, an inventory of algorithms, protocols, certificates, keys, libraries and dependencies used by a system.

CRQC: Cryptographically relevant quantum computer, a quantum computer capable of breaking deployed cryptographic algorithms at useful scale.

Hybrid key exchange: A construction that combines traditional and next-generation or post-quantum key exchange so that the final secret remains protected if at least one component remains secure [Stebila et al., 2025].

ML-KEM: Module-Lattice-Based Key-Encapsulation Mechanism standardized in FIPS 203 [National Institute of Standards and Technology, 2024a].

ML-DSA: Module-Lattice-Based Digital Signature Algorithm standardized in FIPS 204 [National Institute of Standards and Technology, 2024b].

SLH-DSA: Stateless Hash-Based Digital Signature Algorithm standardized in FIPS 205 [National Institute of Standards and Technology, 2024c].

PAREK: Post-quantum asset and algorithm inventory, risk assessment, road mapping, execution and key governance [Grigaliunas and Bruzgiene, 2025].

PAREX: Pentesting extension of PAREK used in this book: Prepare, Asset-discover, Risk-evaluate, Evidence-test and eXecute continuous transition.

QARS: Quantum-Adjusted Risk Score, a risk model using timeline, sensi-

tivity and exposure dimensions [Grigaliunas and Bruzgiene, 2025].

ResilQ: A resilience platform concept for PQC readiness, CBOM generation, QARS scoring, RQmod audit jobs, Transit/Use/Rest assessment and quantum demonstration workflows.

**Quantum-Safe Index (QSI).** A PAREK-aligned, evidence-led readiness assessment with six weighted dimensions and critical gates. It is an assessment instrument, not an authority certification.

**RQmod.** A declarative language and ResilQ Studio workflow for modelling post-quantum audit intent and compiling evidence artefacts.

# Appendix D

# Educator Rubrics and Certification Crosswalk

## D.1. Suggested assessment weighting

## D.2. Certification crosswalk

**Table D.1:** Suggested course assessment weighting.

| Component | Weight | Evidence |
|---|---|---|
| Weekly laboratory portfolio | 35% | Scope, raw artefact, interpretation, limitation, remediation |
| Standards and source brief | 15% | Primary-source comparison and status check |
| Team capstone | 30% | CBOM, risk map, validated findings, executive summary |
| Oral report defence | 10% | Trace claims to evidence and answer limitations |
| Ethics and reflection | 10% | Stop conditions, minimisation, professional judgement |

**Table D.2:** High-level certification crosswalk; always verify the current official blueprint.

| Programme | Closest handbook coverage | Important limitation |
|---|---|---|
| CEH | Ethics, reconnaissance, scanning, enumeration, vulnerability analysis, cryptography, reporting | This book is not EC-Council courseware and does not reproduce exam items. |
| OPST/OPSA | Scope, operational channels, evidence, metrics, analysis, reporting | Use the current OSSTMM and ISECOM syllabi for certification-specific terminology. |
| CompTIA PenTest+ | Engagement management, reconnaissance, vulnerability analysis, attacks, scripting, remediation | Quantum-safe testing extends rather than replaces the current PT0-003 objectives. |

# Bibliography

Gorjan Alagic, Elaine Barker, Lily Chen, Dustin Moody, Angela Robinson, Hamilton Silberg, and Noah Waller. NIST SP 800-227: Recommendations for Key-Encapsulation Mechanisms. Technical report, 2025.

Y. Baseri, A. Hafid, and A. Habibi Lashkari. Future-proofing cloud security against quantum attacks, 2025. arXiv:2509.15653.

A. Blanco-Romero, D. Nunez, V. Martin, and V. Costela. Qkd-kem, 2025. arXiv:2503.07196.

A. C. H. Chen. A comparative study of hybrid post-quantum cryptographic x.509 certificate schemes, 2025a. arXiv:2511.00111.

A. C. H. Chen. Hybrid schemes of nist post-quantum cryptography and quantum key distribution, 2025b. arXiv:2510.02379.

Alvaro Cintas Canto, Jasleen Kaur, Mehran Mozaffari Kermani, and Reza Azarderakhsh. Algorithmic security is insufficient: A comprehensive survey on implementation attacks haunting post-quantum security, 2023. arXiv:2305.13544.

CISA and NSA and NIST. Quantum-readiness: Migration to post-quantum cryptography. Technical report, Cybersecurity and Infrastructure Security Agency, 2023.

CompTIA. Comptia pentest+ pt0-003 certification exam objectives, 2024. Use the current official version.

EC-Council. Ceh exam blueprint v5.0, 2024. URL https://cert.eccouncil.org/wp-content/uploads/2024/04/CEH-Exam-Blueprint-v5.pdf.

European Commission. European quantum communication infrastructure - euroqci, 2026. URL https://digital-strategy.ec.europa.eu/en/policies/european-quantum-communication-infrastructure-euroqci.

European Commission and NIS Cooperation Group. A coordinated implementation roadmap for the transition to post-quantum cryptography, 2025. URL https://digital-strategy.ec.europa.eu/en/library/coordinated-implementation-roadmap-transition-post-quantum-cryptography.

Sarunas Grigaliunas. Passq - post-quantum password manager, 2026a. URL https://passq-app.vercel.app/.

Sarunas Grigaliunas. Pqc.lt: Lithuania's post-quantum security path, 2026b. URL https://www.pqc.lt/.

Sarunas Grigaliunas. Lithuanian qci, 2026c. URL https://qci.lt/.

Sarunas Grigaliunas. Qmacs editor, 2026d. URL https://q-macs-editor.vercel.app/.

Sarunas Grigaliunas. Quantum-safe index, 2026e. URL https://quantumsafeindex.com/.

Sarunas Grigaliunas. Rqmod - resilq studio, 2026f. URL https://rqmod.com/.

Sarunas Grigaliunas and Rasa Bruzgiene. Towards a unified quantum risk assessment. *Electronics*, 14(17):3338, 2025. doi: 10.3390/electronics14173338.

ISECOM. Osstmm professional security tester and analyst certifications, 2026. URL https://www.isecom.org/certification.html.

John Wiley and Sons. Prepare your manuscript for submission, 2026. URL https://www.wiley.com/en-ie/publish/book/steps/prepare-manuscript/.

Kris Kwiatkowski, Panos Kampanakis, Bas Westerbaan, and Douglas Stebila. Post-quantum hybrid ecdhe-mlkem key agreement for tlsv1.3, 2026. IETF Internet-Draft.

B. LaMacchia, M. Campagna, and W. Gropp. The post-quantum cryptography transition, 2025. arXiv:2503.04806.

Dustin Moody, Ray Perlner, Andrew Regenscheid, Angela Robinson, and David Cooper. NIST IR 8547 (Initial Public Draft): Transition to Post-Quantum Cryptography Standards. Technical report, National Institute of Standards and Technology, 2024.

National Cybersecurity Center of Excellence. Migration to post-quantum cryptography, 2026. URL https://www.nccoe.nist.gov/crypto-agility-considerations-migrating-post-quantum-cryptographic-algorithms.

National Institute of Standards and Technology. FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard. Technical report, 2024a.

National Institute of Standards and Technology. FIPS 204: Module-Lattice-Based Digital Signature Standard. Technical report, 2024b.

National Institute of Standards and Technology. FIPS 205: Stateless Hash-Based Digital Signature Standard. Technical report, 2024c.

National Institute of Standards and Technology. Nist selects hqc as fifth algorithm for post-quantum encryption, 2025. URL https://www.nist.gov/news-events/news/2025/03/nist-selects-hqc-fifth-algorithm-post-quantum-encryption.

National Institute of Standards and Technology. Post-quantum cryptography project, 2026. URL https://csrc.nist.gov/Projects/post-quantum-cryptography.

Open Quantum Safe Project. Software for prototyping quantum-resistant cryptography, 2026. URL https://openquantumsafe.org/.

OWASP Foundation. Web security testing guide, 2026. URL https://owasp.org/www-project-web-security-testing-guide/.

J. Rassekhnia. Qers: Quantum encryption resilience score, 2026. arXiv:2601.13399.

A. Shaw. Quantum-safe code auditing, 2026. arXiv:2604.00560.

Douglas Stebila, Scott Fluhrer, and Shay Gueron. Hybrid key exchange in tls 1.3, 2025. IETF Internet-Draft.

C. Turino, W. J. Buchanan, O. Lo, and C. Thuummler. Pqc-leo, 2026. arXiv:2603.06149.

K. Wang, D. Xu, and J. Tian. An improved two-step attack on kyber, 2024. arXiv:2407.06942.