

START WITH THE ASSET

A requirement becomes understandable when you can name what it protects, where it operates, and which relationships make it relevant.

By Scott Norton / PCI DSS QSA | Johanson Group, LLP

PRACTITIONER POSITION The requirement does not exist in isolation.

A checklist begins with a numbered requirement. An environment does not. It begins with payment channels, systems, software, identities, network paths, service providers, operational processes, and the relationships among them. If the first question is only, 'What evidence do we need for this requirement?' the team may produce documents before it understands what the control is meant to protect.

My preferred starting point is deliberately concrete: start with the asset, then follow the relationships. What is it? What service does it provide? What can reach it? What does it reach? Who administers it? Which software and providers does it depend on? Which business process changes it? Those answers give the requirement somewhere real to live.

CONTEXT CHANGES THE MEANING OF A CONTROL

The same firewall rule, user account, cloud service, script, web server, or vulnerability can carry very different significance depending on its relationship to the payment architecture. Looking at the object without its relationships encourages generic conclusions. Looking at the relationships explains why the object matters, who can influence it, and which claims depend on it.

This is why architecture-before-checklist is not an argument against requirements. It is how requirements become operational. The numbered text tells us the outcome and the testing direction. The environment tells us where that outcome must be achieved, which implementation exists, and what evidence could credibly demonstrate it.

A PCI ENVIRONMENT BEHAVES LIKE A GRAPH

Inventories are necessary, but rows hide relationships. Assets use services. Services expose protocols and ports. Applications depend on infrastructure. Identities administer systems. Providers perform activities. Requirements protect components and processes. Evidence supports claims about those activities. A spreadsheet can store many of these facts while still making the connecting edges difficult to see.

The graph does not need to begin as a graph database. A whiteboard is enough if the team draws the important edges and labels what each edge means. The point is to make impact paths inspectable. One administrative identity may explain several system relationships. One shared service may connect scope, configuration, logging, change management, and evidence. The relationship is often the unit of understanding.

BEGIN WITH WHY THIS EXISTS HERE

Before requesting evidence for a sub-requirement, ask why that requirement exists in this environment. Which assets does it protect? Which services, protocols, ports, identities, or processes are involved? Why are they necessary? Who approved them? Where are those facts maintained? What changes them? These are not delays before compliance work. They are the work required to make the compliance answer specific.

Once the relationships are visible, evidence stops being a scavenger hunt organized by requirement number. Configuration data can support the service story. Inventory can define the population. Change records can explain how the state moved. Ownership can be attached to the decisions. The assessor can then test a coherent operating model instead of reconciling isolated files whose relationship exists only in a narrative written for the assessment.

USE ONE PATH AS THE PROVING GROUND

Choose one asset that matters and trace one complete path through it. Follow a connection, an administrative workflow, a software dependency, or a provider-delivered service. Label the source, destination, purpose, owner, approval, configuration source, evidence, and change trigger. Then ask which statements are observed facts, which are assumptions, and which are conclusions.

A single well-explained path can expose the quality of the wider program. If the team can answer the questions quickly, the underlying sources of truth are probably doing useful work. If every answer requires a different meeting, an inherited spreadsheet, or the person who remembers the original implementation, the path has located a concentration of Wack worth addressing.

FIVE QUESTIONS FOR THE NEXT WORKING SESSION

- What asset or process is this requirement protecting here?
- Which relationships make the asset relevant?
- Where are the relationship facts maintained?
- Who owns the path rather than only one endpoint?
- What event changes the map and triggers review?

ONE MOVE TO MAKE NOW

Pick one important asset and trace one real connection or administrative workflow through its purpose, owner, approval, configuration, evidence, and change trigger.

THE PRACTITIONER POSITION

The checklist is easier to use after the environment has a shape. Requirements stop floating as isolated obligations and begin attaching to real assets, real relationships, real owners, and real evidence.

Start with the asset. Follow the relationships. Then let the requirement tell you what must be true about the system you now understand.

AUTHORITY AND SOURCES

Independent practitioner perspective. Scott Norton is a PCI DSS QSA with Johanson Group, LLP. Unless specifically identified as an organizational position, the analysis and recommendations are his practitioner views, not formal positions of Johanson Group or PCI SSC. Cluster of Wack and related terms are not PCI SSC terminology. This material is educational and does not provide an assessment conclusion or imply endorsement by PCI SSC.

Authority classification: PRACTITIONER_POSITION / AI_ASSISTED. Authorship and authority classification do not determine legal intellectual-property ownership.

- PCI DSS v4.0.1 - PCI Security Standards Council: https://www.pcisecuritystandards.org/document_library/?class=pcidss&doc=pci_dss