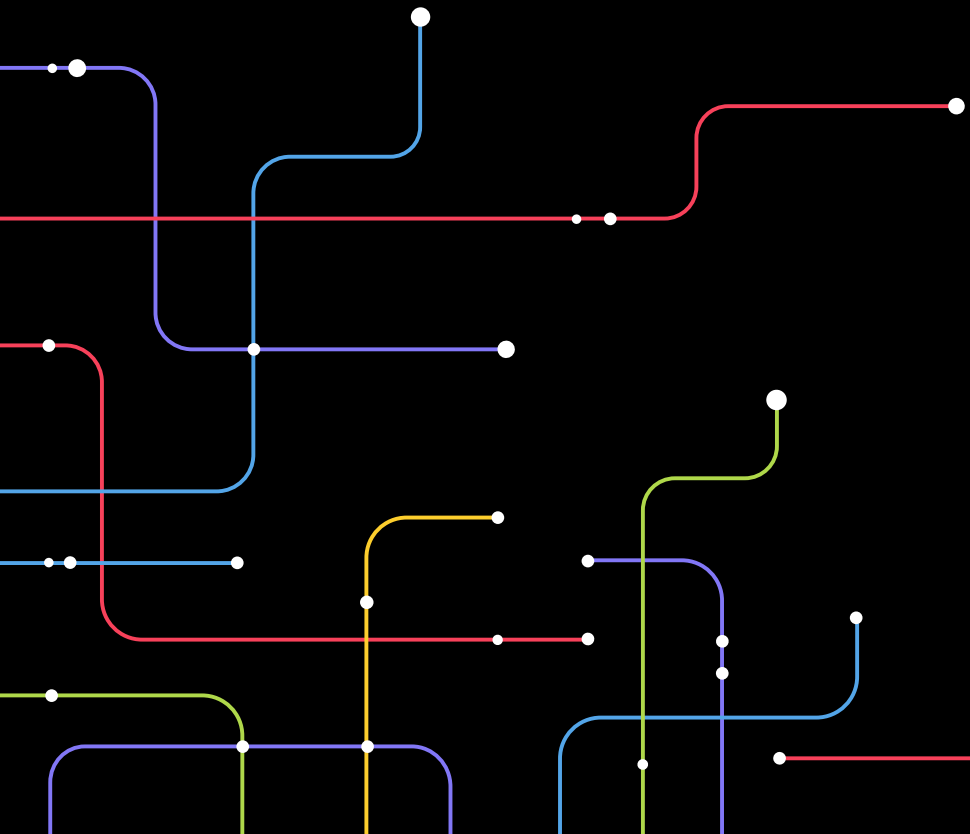
tarmac.io

Practical PCI DSS in AWS: **From Requirements to Reality**



Practical PCI DSS in AWS:
From Requirements to Reality

Tarmac DevOps Team

Content

Introduction

This booklet is written for CTOs, Engineering Managers, platform teams, DevOps leaders, and security stakeholders who need a practical understanding of how PCI DSS applies to AWS environments. It is not intended to replace the PCI DSS standard, a Qualified Security Assessor, or a detailed implementation plan. Instead, it provides a technical management-level interpretation of the controls, with enough AWS-specific context to support better architectural and operational decisions.

The goal is to help readers understand what each PCI DSS requirement is really asking for, how that maps to AWS-native patterns, where the shared responsibility model matters, and where common mistakes can quietly expand scope or increase audit risk.

PCI DSS v4.0.1 places significant emphasis on governance, scoping, secure configuration, access control, monitoring, vulnerability management, incident response, and the protection of account data. In AWS, these requirements do not disappear; they shift into architecture decisions, service configuration, identity design, logging strategy, deployment practices, and operational ownership.

PCI DSS compliance in AWS is rarely a question of whether the cloud platform is secure. AWS provides mature security primitives, compliant infrastructure, and a broad set of managed services. The real question is whether the environment built on top of AWS is designed, configured, operated, and evidenced in a way that satisfies PCI DSS.

That distinction matters.

Overview

The booklet follows the structure of PCI DSS v4.0.1 and groups each principal requirement into a concise interpretation. Each section starts with the official control family, then explains the practical meaning of those controls in an AWS context.

The focus is not on copying the standard or providing step-by-step deployment instructions. Instead, each requirement is translated into engineering concerns: network boundaries, secure configurations, data storage, encryption, malware protection, software delivery, identity, physical access, logging, testing, and security governance.

Where relevant, the booklet also highlights how AWS shared responsibility applies. Some responsibilities belong primarily to AWS, such as physical data center controls and underlying infrastructure security. Others remain firmly with the customer, including data handling, IAM design, application security, encryption configuration, logging, monitoring, change management, and evidence collection. Many areas are shared, which is where misunderstandings most often occur.

A recurring theme throughout the booklet is scope. A well-designed AWS environment can reduce PCI DSS scope by isolating the Cardholder Data Environment, minimizing stored account data, using managed services appropriately, and controlling access paths. A poorly designed environment can do the opposite, pulling unrelated systems, teams, tools, and networks into scope.

How to Use This Booklet

Use this booklet as a guided interpretation of PCI DSS in AWS, not as a compliance checklist. The PCI DSS standard itself remains the authoritative source, and any formal compliance program should be validated

with qualified assessors and the relevant payment brands or acquiring institutions.

For executives and technology leaders, this booklet can help frame the right questions: where cardholder data exists, what systems can affect it, which teams own the controls, what evidence is available, and whether AWS services are being used in a way that reduces or increases compliance burden.

For engineering and platform teams, the sections can be used to review architecture and operational practices before deeper implementation work begins. The intent is to identify design gaps early, especially around segmentation, identity, logging, secure configuration, data retention, and change control.

For security and compliance stakeholders, this booklet can serve as a conversation bridge between PCI control language and AWS implementation reality. It should help turn abstract requirements into concrete areas for review, without prescribing one-size-fits-all architecture.

The most effective way to use the booklet is to read it in three passes. First, read it end to end to understand the full PCI DSS operating model in AWS. Second, revisit each requirement against your actual environment and mark which controls are customer-owned, AWS-owned, shared, or not applicable. Third, use those findings to create a prioritized roadmap for scope reduction, remediation, evidence collection, and long-term compliance operations.

PCI DSS in AWS is not solved by a single service, tool, or diagram. It is solved by clear architecture, disciplined operations, and evidence that proves the controls work over time.

Requirement 1 - Install and Maintain Network Security Controls PCI DSS v4.0.1 (Controls 1.1 – 1.5)

Control Overview

1.1 – Processes and mechanisms for installing and maintaining network security controls are defined and understood
1.2 – Network security controls (NSCs) are configured and maintained
1.3 – Network access to and from the CDE is restricted
1.4 – Network connections between trusted and untrusted networks are controlled
1.5 – Risks from systems connecting to both untrusted networks and the CDE are mitigated

What These Controls Mean (Aggregated)

Taken together, Requirement 1 enforces a single principle:
All network traffic must be intentional, controlled, and governed - especially around the CDE boundary.

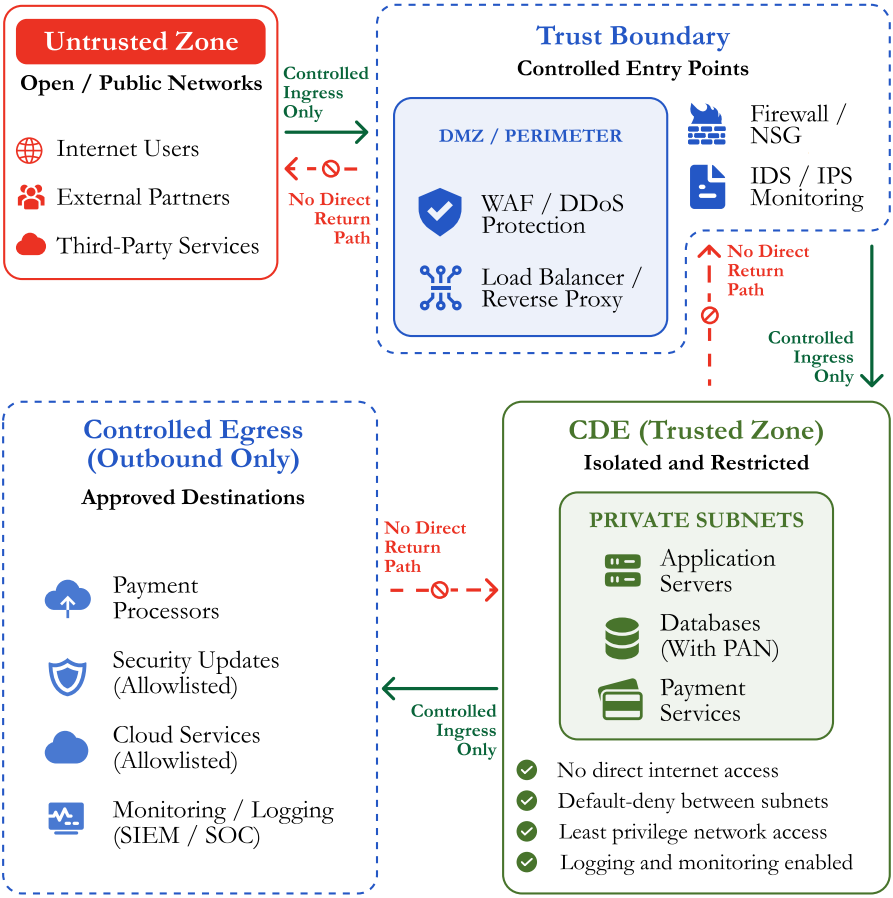
This spans three layers:

- **Governance (1.1, 1.2)**
  Network controls must be defined, owned, documented, and consistently maintained over time.
- **Segmentation & Boundaries (1.3, 1.4)**
  The CDE must be isolated, and all traffic crossing trust boundaries must be explicitly controlled.
- **Bypass Prevention (1.5)**
  No system should be able to unintentionally bridge secure and un trusted networks.

AWS Interpretation

In AWS, this translates into a structured and layered network model:

- **Clear CDE isolation**
  - Dedicated VPCs or accounts for in-scope systems
  - Private subnets with no direct internet exposure
- **Explicit traffic control**
  - Default-deny posture with tightly scoped allow rules
  - Defined ingress and egress paths only
- **Controlled trust boundaries**
  - Internet and third-party integrations treated as untrusted
  - All boundary crossings routed through controlled entry/exit points
- **No implicit connectivity**
  - No reliance on "internal = trusted" assumptions
  - Every allowed connection must be justified



Isolate the CDE

Keep the CDE on dedicated networks with no direct exposure to untrusted zones.

Controll All Access

Only explicitly allowed traffic is permitted across trust boundaries.

Prevent Bypass

No system may connect from untrusted networks directly to the CDE.

Monitor Continuously

Inspect, detect, and alert on unauthorized or suspicious traffic.

Maintain & Review

Keep rules, boundaries, and trust models documented and up to date.

Implementation Approach

To satisfy Requirement 1 effectively, network security must be treated as a managed system, not a collection of configurations:

- **Define everything as IaC**
  - Network rules, routing, and segmentation are version-controlled
  - Eliminates drift and enables auditability
- **Enforce change control**
  - All modifications go through review and approval workflows
  - No direct manual changes in production environments
- **Standardize patterns**
  - Reusable network designs across environments
  - Consistent enforcement of segmentation and access rules
- **Continuously validate**
  - Monitor traffic patterns (e.g., flow logs)
  - Regularly review rules for necessity and scope creep

Common Failure Modes

- Flat networks where everything can reach everything
- Overly permissive rules (especially outbound)
- Uncontrolled connectivity via peering or routing
- Admin systems (laptops, bastions) bridging trusted and untrusted networks
- Manual changes bypassing defined processes

Auditor's Lens

Assessment will focus on:

- Whether network controls are documented and governed
- Whether access to the CDE is explicitly restricted
- Whether trust boundaries are enforced
- Whether any system can bypass or weaken controls

Positioning Insight

Requirement 1 is less about firewalls and more about **control over network intent.**

Organizations that implement it well:

- Reduce PCI scope significantly
- Avoid cascading compliance complexity
- Establish a repeatable foundation for all other controls
  This is where strong platform engineering practices directly translate into compliance maturity.

AWS Shared Responsibility Note
Within AWS, Requirement 1 is **mostly customer responsibility.** AWS is responsible for the physical network, facilities, and underlying cloud infrastructure, but the customer remains responsible for VPC design, segmentation, routing, security groups, network ACLs, firewall policy, and controlling traffic to and from the CDE. In other words, AWS provides the network primitives; meeting PCI Requirement 1 depends primarily on how those primitives are designed and governed. In fully managed services, some lower-layer network mechanics are abstracted away, but the responsibility for defining secure connectivity and restricting scope does not disappear.

AWS-Only / Not Applicable Considerations
Some lower-level infrastructure controls are not directly implemented by the customer in AWS because they fall under AWS's responsibility for the underlying cloud infrastructure. However, that does not make Requirement 1 "not applicable" in AWS. The requirement still applies; it is simply satisfied through the customer's control of AWS-native networking constructs and through shared-responsibility evidence where relevant.

Requirement 2 - Apply Secure Configurations to All System Components PCI DSS v4.0.1 (Controls 2.1 – 2.3)

Control Overview

2.1 – Processes and mechanisms for applying secure configurations to all system components arve defined and understood
2.2 – System components are configured and managed securely
2.3 – Wireless environments are configured and managed securely

What These Controls Mean (Aggregated)

Requirement 2 enforces a single principle:
All systems must be securely configured by design and continuously maintained in that state.

This breaks down into:

- **Governance (2.1)**
  Secure configuration standards must be defined, owned, and consis tently applied.
- **System Hardening (2.2)**
  All components must follow a hardened baseline and avoid insecure defaults.
- **Wireless Risk Management (2.3)**
  Wireless introduces additional exposure and must be explicitly secured or excluded.

AWS Interpretation

In AWS, Requirement 2 shifts from traditional "server hardening" to **configuration discipline across all service layers:**

- **No reliance on defaults**
  - Vendor defaults are not considered secure
  - All services must be explicitly configured
- **Minimal attack surface**
  - Only required services, ports, and protocols are enabled
  - Unused features are disabled or not provisioned at all
- **Consistent baselines**
  - Standardized configurations across environments
  - No drift between dev, staging, and production
- **Service-first hardening**
  - Focus moves from OS-level hardening to:
    - Service configuration (e.g., database settings, access policies)
    - Network exposure controls
    - Encryption and logging defaults

Layer 1

Service Configuration

Configure each service securely (compute, storage, database, serverless, messaging, etc.)

- Secure settings & parameters
- Disable unused features
- Enable encryption & logging
- Use hardened baselines

Key Practices

- No reliance on defaults
- Monitor for configuration drift
- Apply CSI / AWS security baselines
- Review and test configurations

Layer 2

Network Exposure

Limit and control what is reachable across networks (internet, VPC, peering, VPN, third parties, wireless).

- Minimize open ports & protocols
- Use firewalls, SGs, NACLs
- Segment and isolate
- Protect edge & API endpoints

Key Practices

- Default deny all traffic
- Implement segmentation
- Only required services exposed
- Verify connectivity regularly

Layer 3

Access Controls

Control who and what can access systems and services (users, roles, applications, service accounts).

- Least privilege access
- Strong IAM policies & roles
- MFA for privileged access
- Manage keys, secrets, certificates

Key Practices

- Grant minimum necessary access
- Enforce MFA for admin and users
- Avoid broad or wildcard permissions
- Regular access reviews

Secure by Design

Build with security in mind. Avoid insecure defaults.

Consistent & Repeatable

Use standard baselines across all environments.

Continuously Maintained

Detect drift, remediate, and re-validate.

Consistent & Repeatable

Continuously monitor configu-raitons and network exposure.

Implementation Approach

Requirement 2 is primarily about **standardization and enforcement:**

- **Define secure configuration baselines**
  - Per service type (compute, database, storage, etc.)
  - Include access, encryption, logging, and exposure settings
- **Enforce via IaC**
  - All configurations are declarative and version-controlled
  - Prevents configuration drift and enforces consistency
- **Reduce surface area aggressively**
  - Disable unused services and protocols
  - Avoid "just in case" configurations
- **Continuously validate**
  - Detect deviations from baseline
  - Regularly review configurations against current requirements

Common Failure Modes

- Leaving default configurations unchanged
- Over-enabling services or features without clear need
- Configuration drift between environments
- Legacy or insecure protocols still enabled
- Assuming managed services are secure without validation

Auditor's Lens

Assessment will focus on:

- Whether secure configuration standards are defined and maintained
- Whether systems are consistently hardened
- Whether unnecessary services and features are disabled
- Whether configurations are controlled and auditable

AWS Shared Responsibility Considerations

Within AWS, Requirement 2 is **shared**, but the majority of responsibility remains with the customer.

- AWS is responsible for:
  - Underlying infrastructure
  - Physical hardware and facilities
  - Host OS and platfor m layers for managed services
- The customer is responsible for:
  - Service configuration (databases, storage, compute settings)
  - Guest OS hardening (for EC2 and similar)
  - Network exposure and access controls
  - Encryption, logging, and security settings

The exact split depends on the service model, but **secure configuration always remains a customer-controlled outcome.**

AWS-Only / Applicability Notes

- **Managed Services**
  Some traditional hardening steps (e.g., OS patching, host-level config uration) are not directly applicable when using fully managed services. Responsibility shifts to **secure service configuration**, not infrastruc ture control.
- **Wireless (2.3)**
  Often not applicable in AWS-native-only environments. However, it becomes relevant in:
  - Hybrid environments
  - Office or retail networks
  - Any environment where wireless connects into the CDE

Requirement 2 is therefore **not eliminated in AWS** - it is **translated** from infrastructure hardening into **configuration governance across cloud services.**

Positioning Insight

Requirement 2 is about **eliminating unnecessary risk at the system level.**

Organizations that implement it well:

- Minimize their attack surface
- Prevent misconfiguration-driven vulnerabilities
- Create a consistent, repeatable security baseline

In practice, this is one of the highest-leverage areas for improving both **security posture and operational maturity.**

Requirement 3 - Protect Stored Account Data
PCI DSS v4.0.1 (Controls 3.1 – 3.7)

Control Overview

3.1 – Processes and mechanisms for protecting stored account data are defined and understood
3.2 – Storage of account data is kept to a minimum
3.3 – Sensitive authentication data (SAD) is not stored after authorization
3.4 – Access to displays of full PAN and ability to copy PAN are restricted
3.5 – Primary account number (PAN) is secured wherever it is stored
3.6 – Cryptographic keys used to protect stored account data are secured
3.7 – Where cryptography is used to protect stored account data, key management processes and procedures covering all aspects of the key lifecycle are defined and implemented

What These Controls Mean (Aggregated)

Requirement 3 enforces a single principle:
Stored account data must be minimized, tightly controlled, and rendered unusable if accessed without authorization.

This breaks down into:

- **Governance (3.1)**
  Policies, procedures, and ownership for stored account data protec tion must be defined and followed.
- **Data Minimization (3.2, 3.3)**
  Keep as little account data as possible, and do not retain SAD after authorization.

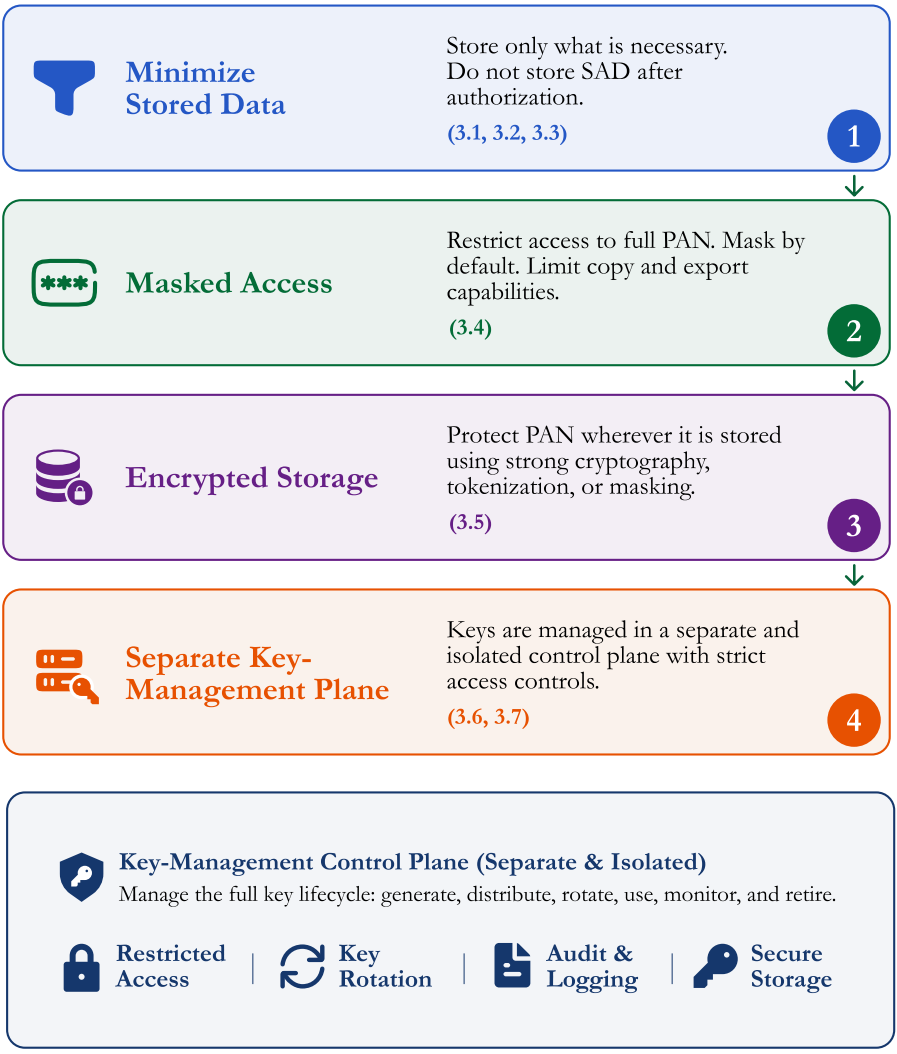
- **Exposure Reduction (3.4, 3.5)**
  Wireless introduces additional exposure and must be explicitly se cured or excluded.
- **Cryptographic Protection (3.6, 3.7)**
  Protect the keys as rigorously as the data, and manage the full key lifecycle formally.

AWS Interpretation

In AWS, Requirement 3 is primarily about **data architecture and cryptographic control**, not just encryption toggles.

Key expectations:

- **Store less**
  - Avoid retaining PAN unless there is a clear business need
  - Eliminate accidental storage in logs, exports, snapshots, queues, and support tooling
- **Never retain SAD after authorization**
  - This is a strict boundary, not a best practice
  - Temporary handling paths must not become persistent storage paths
- **Restrict visibility of PAN**
  - Full PAN should be masked in UIs, reports, and support work flows
  - Copy/export capability should be limited to explicitly authorized roles and use cases
- **Secure PAN at rest**
  - Where stored, PAN must be protected using approved methods such as strong cryptography, truncation, masking, or hashing as applicable
- **Treat key management as a separate control plane**
  - Encryption is not enough if keys are poorly governed
  - Key access, rotation, storage, and usage must be tightly controlled



Reduce Risk
Store less.
Expose less.

Limit Exposure
Mask by default.
Restrict full PAN access.

Protect Data at Rest
Strong cryptography or equivalent protection.

Protect the Keys
Keys are protected as rigorously as the data.

Implementation Approach

Requirement 3 is best implemented by combining data minimization, service design, and cryptographic discipline:

- **Reduce retained data first**
  - Define exactly where account data is allowed to exist
  - Remove it everywhere else
- **Design for non-persistence where possible**
  - Prevent account data from landing in logs, caches, message pay loads, temporary files, or analytics sinks
- **Apply strong storage protections**
  - Encrypt stored PAN where required
  - Mask or truncate where full PAN is not operationally needed
- **Separate data access from key access**
  - Personnel and systems that can access stored data should not auto matically have access to cryptographic keys
- **Enforce through IaC and standard patterns**
  - Storage services, encryption settings, access policies, and key usage should be consistently defined and reviewed

Common Failure Modes

- Keeping account data "just in case" without a defined retention need
- Accidentally storing PAN in logs, error traces, exports, or backups
- Treating encrypted data as low risk while key access remains broad
- Allowing too many users or systems to view full PAN
- Failing to distinguish between cardholder data and SAD retention rules

Auditor's Lens

Assessment will focus on:

- Whether stored account data is **minimized**
- Whether SAD is **not retained after authorization**
- Whether access to full PAN is **restricted and justified**
- Whether stored PAN is **properly protected**
- Whether cryptographic keys and key lifecycle processes are **securely managed**

AWS Shared Responsibility Considerations

Within AWS, Requirement 3 is shared, but the customer owns nearly all decisions that determine compliance.

- AWS is responsible for:
  - Underlying infrastructure security
  - Security of the managed service platform beneath customer work loads
- The customer is responsible for:
  - Deciding what account data is stored
  - Preventing prohibited retention of SAD
  - Configuring encryption, masking, truncation, and access controls
  - Defining and enforcing retention policies
  - Controlling cryptographic key usage and lifecycle within the chosen architecture

In other words, AWS provides encryption and storage primitives, but **Requirement 3 compliance depends primarily on customer data handling design.**

AWS-Only / Applicability Notes

- **Managed services do not remove Requirement 3**
  - Using managed databases, storage, or messaging services does not make stored account data protection AWS's responsibility
  - The customer still owns what data is stored, how it is protected, and who can access it

- **Non-persistent memory**
  - The standard notes that if account data is present only in non-per sistent memory, encryption of PAN is not required, but controls must ensure that the memory remains non-persistent and the data is removed once the business purpose is complete
- **Encryption alone does not solve scope or compliance**
  - Requirement 3 is broader than at-rest encryption; it also covers retention discipline, masking, display control, and key management

Positioning Insight

Requirement 3 is where many cloud payment environments either become **cleanly defensible** or **quietly dangerous.**

Organizations that implement it well:

- Reduce breach impact substantially
- Shrink the amount of sensitive data they must defend
- Avoid turning backups, logs, and internal tooling into compliance liabilities

In practice, the strongest move is usually not better encryption alone - it is **storing less, exposing less, and controlling keys properly.**

Requirement 4 - Protect Cardholder Data with Strong Cryptography During Transmission Over Open, Public Networks PCI DSS v4.0.1 (Controls 4.1 – 4.2)

Control Overview

4.1 – Processes and mechanisms for protecting cardholder data with strong cryptography during transmission over open, public networks are defined and understood
4.2 – PAN is protected with strong cryptography during transmission

What These Controls Mean (Aggregated)

Requirement 4 enforces a single principle:
Cardholder data must never be exposed in cleartext during transmission across untrusted networks.

This breaks down into:

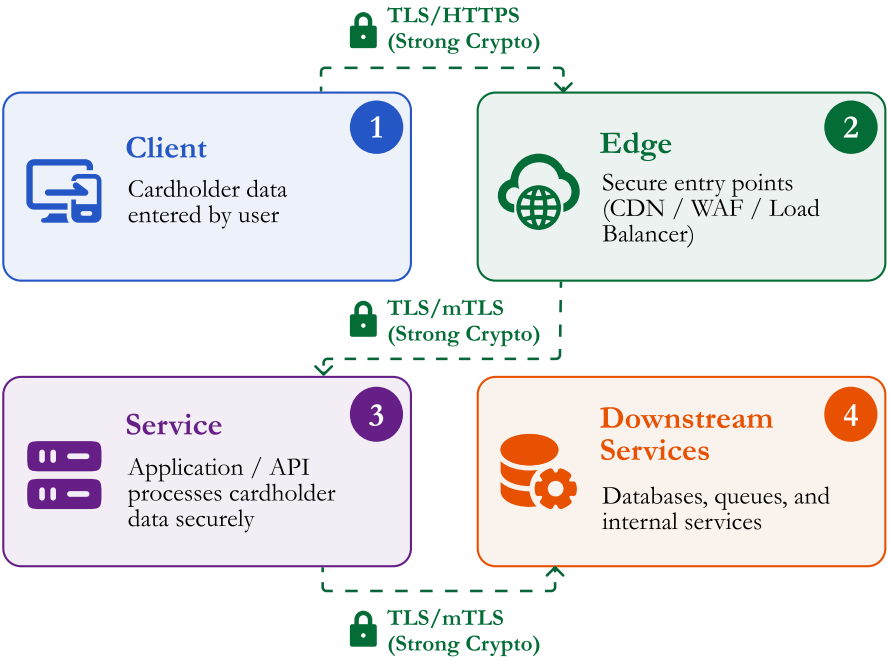
- **Governance (4.1)**
  Policies, procedures, and responsibilities for secure transmission must be defined and consistently applied.
- **Cryptographic Protection (4.2)**
  PAN must be protected using strong cryptography whenever it tra verses open or public networks.

AWS Interpretation

In AWS, Requirement 4 is primarily about **securing data in motion across all boundaries**, not just internet-facing traffic.

Key expectations:

- **Encrypt all external transmission paths**
  - Internet-facing endpoints (APIs, load balancers, web apps) must enforce strong TLS
  - No fallback to insecure protocols or cipher suites
- **Treat "untrusted" broadly**
  - Internet is untrusted by default
  - Third-party integrations are untrusted
  - Any external connectivity must be assumed interceptable
- **Control certificate and protocol hygiene**
  - Only trusted certificates are accepted
  - Expired or revoked certificates are rejected
  - Weak protocols (e.g., legacy TLS versions) are disabled
- **Consider internal transmission scope**
  - While not always mandatory, encrypting internal traffic is strongly recommended
  - Any internal network carrying PAN is automatically in PCI scope



Encrypt Everywhere

Cardholder data (including PAN) is protected with strong cryptography across all boundaries.

Strong Cryptography

Use strong TLS protocols and ciphers. Disable weak or insecure versions.

Verify & Enforce

Validate certificates and enforce secure protocol configurations.

Implementation Approach

Requirement 4 is best approached as end-to-end transport security:

- **Enforce strong protocols everywhere**
  - TLS-only communication for all external endpoints
  - No downgrade or fallback paths
- **Standardize certificate management**
  - Centralized certificate issuance and rotation
  - Validation of trust chains and expiration
- **Eliminate cleartext transmission paths**
  - No HTTP, FTP, or other insecure protocols for PAN transmission
  - Ensure redirects or misconfigurations do not expose cleartext
- **Secure service-to-service communication**
  - Apply encryption consistently across service boundaries
  - Especially important for hybrid or multi-account architectures
- **Enforce via IaC**
  - Protocols, listeners, certificates, and encryption policies defined declaratively
  - Prevents drift and misconfiguration

Common Failure Modes

- Allowing fallback to insecure TLS versions or cipher suites
- Expired or invalid certificates still in use
- Partial encryption (e.g., encrypting edge but not downstream services)
- Misconfigured endpoints allowing cleartext access
- Assuming internal traffic does not need encryption

Auditor's Lens

Assessment will focus on:

- Whether transmission security policies are **defined and enforced**
- Whether PAN is **always encrypted over open/public networks**

- Whether only **strong cryptography and secure protocols** are used
- Whether certificate and key usage is **properly managed**

AWS Shared Responsibility Considerations

Within AWS, Requirement 4 is **shared**, but enforcement is largely customer-driven.

- AWS is responsible for:
  - Security of the underlying network infrastructure
  - Availability of secure transport mechanisms within services
- The customer is responsible for:
  - Enforcing TLS on endpoints
  - Selecting secure protocols and cipher suites
  - Managing certificates and trust chains
  - Ensuring no cleartext transmission paths exist
  - Securing service-to-service communication

Even when using managed services, **encryption in transit must be explicitly configured and validated.**

AWS-Only / Applicability Notes

- **Managed services do not guarantee compliance**
  - Many AWS services support encryption in transit, but it must be explicitly enabled and enforced
  - Default configurations may still allow insecure access paths if not restricted
- **Internal traffic considerations**
  - Encrypting internal traffic is not always strictly required, but is strongly recommended
  - Any internal transmission of PAN brings that network into PCI scope and must be evaluated accordingly

- **Protocol configuration is critical**
  - Requirement 4 is not just about "using TLS"
  - It requires correct implementation (trusted certs, strong versions, no fallback)

Positioning Insight

Requirement 4 is about **eliminating interception risk during transmission**.

Organizations that implement it well:

- Prevent exposure of sensitive data in transit
- Reduce risk from misconfigured endpoints and integrations
- Establish consistent, secure communication patterns across systems

In practice, the biggest failures are rarely about missing encryption - they are about **misconfigured encryption.**

Requirement 5 - Protect All Systems and Networks from Malicious Software PCI DSS v4.0.1 (Controls 5.1 – 5.4)

Control Overview

5.1 – Processes and mechanisms for protecting systems and networks from malicious software are defined and understood
5.2 – Malicious software (malware) is prevented, or detected and addressed
5.3 – Anti-malware mechanisms and processes are active, maintained, and monitored
5.4 – Anti-phishing mechanisms protect users against phishing attacks

What These Controls Mean (Aggregated)

Requirement 5 enforces a single principle:
Systems must be continuously protected against malware and user-targeted attacks, with detection and response capabilities in place.

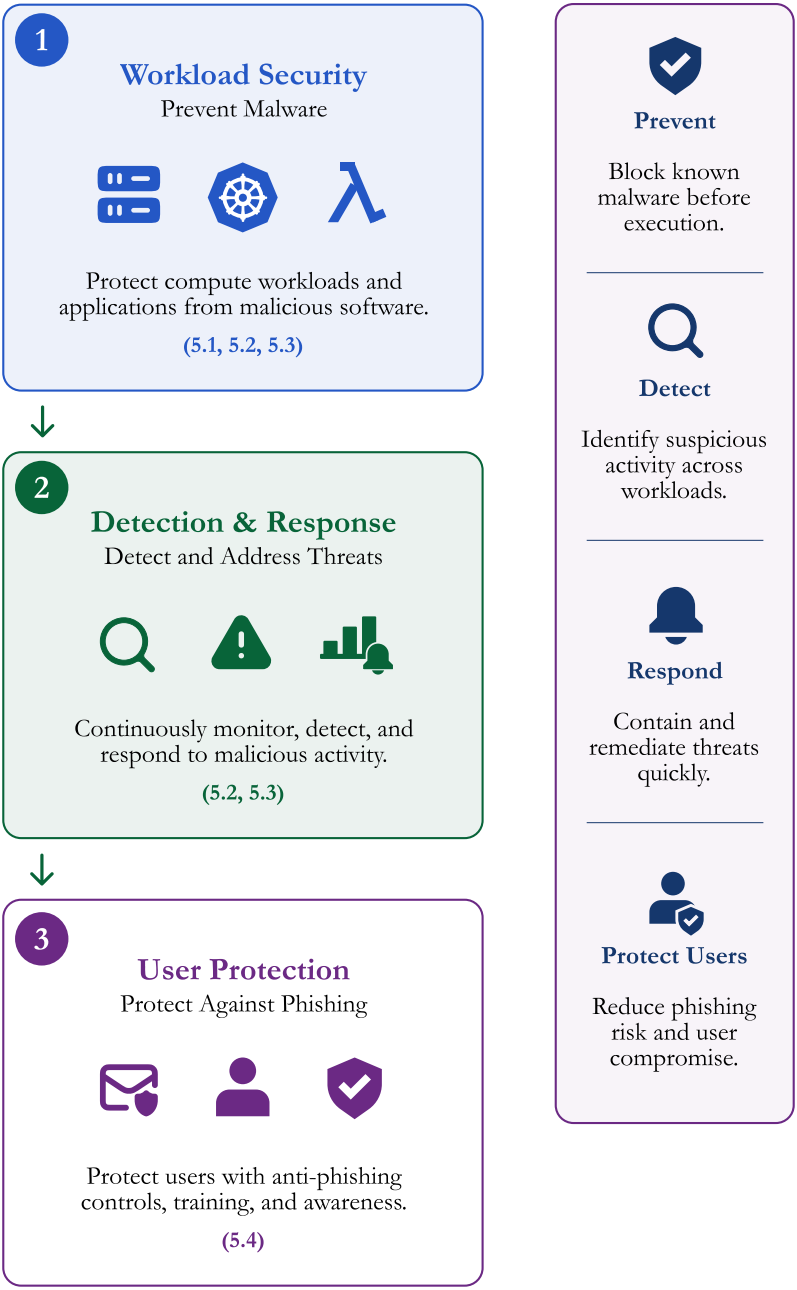
This breaks down into:

- **Governance (5.1)**
  Anti-malware and anti-phishing strategies must be defined, owned, and consistently applied.
- **Malware Protection (5.2, 5.3)**
  Systems must actively prevent or detect malware and respond to threats.
- **Human Attack Surface (5.4)**
  Users must be protected against phishing, which is a primary attack vector.

AWS Interpretation

In AWS, Requirement 5 shifts from traditional "install antivirus everywhere" to a **layered protection model across compute, workloads, and users:**

- Workload protection
  - Compute resources (e.g., EC2, containers) must have malware protection or compensating controls
  - Serverless and managed services reduce traditional malware exposure but do not eliminate risk
- Detection over assumption
  - Systems must assume malware is possible and implement detection/response mechanisms
  - Logging, monitoring, and alerting become critical
- User-focused protection
  - Phishing is often the initial entry point
  - Identity systems and communication channels must be protected
- Service-aware security model
  - Not all AWS services require traditional anti-malware agents
  - Controls must be adapted based on service type and risk profile


1
Workload Security
Prevent Malware
Protect compute workloads and applications from malicious software.
(5.1, 5.2, 5.3)
2
Detection & Response
Detect and Address Threats
Continuously monitor, detect, and respond to malicious activity.
(5.2, 5.3)
3
User Protection
Protect Against Phishing
Protect users with anti-phishing controls, training, and awareness.
(5.4)
Prevent
Block known malware before execution.
Detect
Identify suspicious activity across workloads.
Respond
Contain and remediate threats quickly.
Protect Users
Reduce phishing risk and user compromise.

Implementation Approach

Requirement 5 is best approached through **defense in depth:**

- **Define where malware protection is required**
  - Identify systems susceptible to malware (e.g., general-purpose OS)
  - Apply appropriate controls per system type
- **Deploy detection and prevention mechanisms**
  - Anti-malware tools where applicable
  - Behavioral monitoring and anomaly detection
- **Ensure continuous operation**
  - Mechanisms must be actively running
  - Signatures/configurations must be up to date
  - Failures must be detected and remediated
- **Protect the user layer**
  - Implement anti-phishing controls
  - Secure email, identity, and access flows
- **Enforce via IaC and standard patterns**
  - Security tooling, configurations, and monitoring defined consistently
  - No ad-hoc or optional deployment of protections

Common Failure Modes

- Assuming managed or serverless services eliminate malware risk entirely
- Anti-malware tools installed but not monitored or updated
- Gaps in container or ephemeral workload protection
- No visibility into malware detection events
- Weak or nonexistent phishing protections for users

Auditor's Lens

Assessment will focus on:

- Whether anti-malware policies and processes are defined and main tained
- Whether systems are protected or compensating controls are justified
- Whether anti-malware mechanisms are active and up to date
- Whether malware detection events are monitored and responded to
- Whether users are protected against phishing attacks

AWS Shared Responsibility Considerations

Within AWS, Requirement 5 is **shared**, but responsibility varies significantly by service model.

- AWS is responsible for:
  - Security of the underlying infrastructure
  - Protection of managed service platforms
- The customer is responsible for:
  - Malware protection on compute workloads (e.g., EC2, containers)
  - Deciding where anti-malware is required vs. where compensating controls apply
  - Monitoring and responding to detection events
  - Implementing anti-phishing protections for users

For fully managed and serverless services, AWS reduces the attack surface, but **the responsibility shifts to configuration, monitoring, and identity protection rather than disappearing.**

AWS-Only / Applicability Notes

- **Managed and serverless services**
  - Traditional anti-malware agents may not apply
  - Requirement is satisfied through:
    - Platform security
    - Monitoring and detection controls
    - Reduced attack surface
- **Containers and ephemeral workloads**
  - Require alternative approaches (image scanning, runtime monitoring)
  - Traditional host-based tools may not be sufficient
- **Phishing (5.4) is always applicable**
  - Even in fully cloud-native environments
  - Identity compromise remains one of the highest-risk attack vectors

Requirement 5 is therefore **not about tools alone**, but about ensuring **effective protection coverage across all system types and user interactions.**

Positioning Insight

Requirement 5 is about acknowledging that compromise attempts will happen - and being prepared for them.

Organizations that implement it well:

- Detect threats early and respond effectively
- Reduce reliance on perimeter defenses alone
- Protect both systems and users as part of a unified security model

In practice, the strongest posture comes from combining **workload protection, detection capabilities, and user-focused security controls.**

Requirement 6 - Develop and Maintain Secure Systems and Software PCI DSS v4.0.1 (Controls 6.1 – 6.5)

Control Overview

6.1 – Processes and mechanisms for developing and maintaining secure systems and software are defined and understood
6.2 – Bespoke and custom software are developed securely
6.3 – Security vulnerabilities are identified and addressed
6.4 – Public-facing web applications are protected against attacks
6.5 – Changes to system components are managed securely

What These Controls Mean (Aggregated)

Requirement 6 enforces a single principle:
Security must be built into systems and software throughout their entire lifecycle - not added afterward.

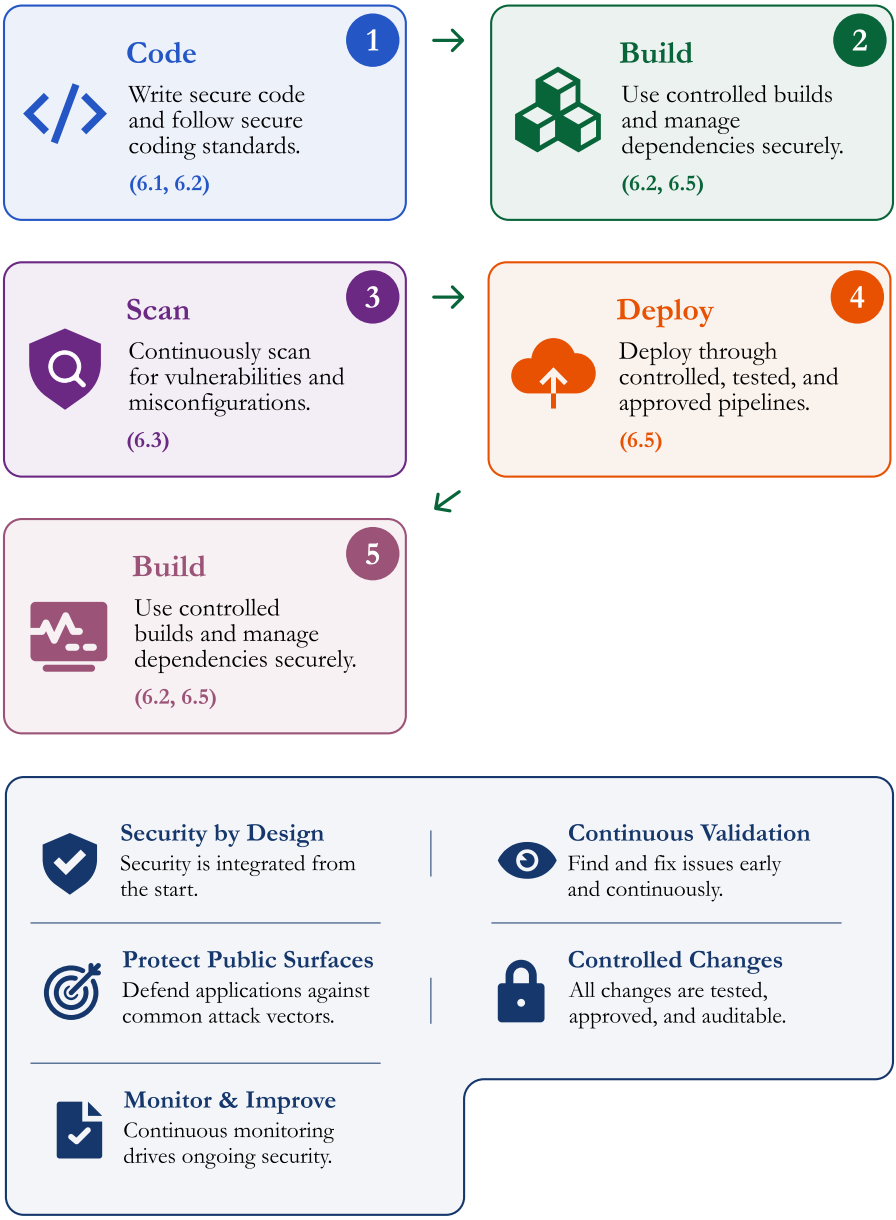
This breaks down into:

- **Governance (6.1)**
  Secure development and maintenance practices must be defined and consistently followed.
- **Secure Development (6.2)**
  Custom software must be designed and built with security in mind.
- **Vulnerability Management (6.3)**
  Systems must be continuously monitored for vulnerabilities and remediated in a timely manner.
- **Application Protection (6.4)**
  Public-facing applications must be protected from common attack vectors.
- **Change Control (6.5)**
  All changes must be controlled, tested, and approved before deployment.

AWS Interpretation

In AWS, Requirement 6 is primarily about **secure software delivery and continuous risk management:**

- **Shift-left security**
  - Security controls must be embedded into the development lifecycle
  - Not deferred to infrastructure or runtime layers
- **Continuous vulnerability awareness**
  - Systems must be monitored for newly discovered vulnerabilities
  - Patching and remediation must be ongoing
- **Protecting exposed surfaces**
  - Public-facing endpoints must be hardened and protected
  - Application-layer attacks are a primary concern
- **Controlled delivery pipelines**
  - Changes must flow through structured, auditable pipelines
  - No direct or unreviewed changes to production systems


1
Code
Write secure code and follow secure coding standards.
(6.1, 6.2)
2
Build
Use controlled builds and manage dependencies securely.
(6.2, 6.5)
3
Scan
Continuously scan for vulnerabilities and misconfigurations.
(6.3)
4
Deploy
Deploy through controlled, tested, and approved pipelines.
(6.5)
5
Build
Use controlled builds and manage dependencies securely.
(6.2, 6.5)
Security by Design
Security is integrated from the start.
Continuous Validation
Find and fix issues early and continuously.
Protect Public Surfaces
Defend applications against common attack vectors.
Controlled Changes
All changes are tested, approved, and auditable.
Monitor & Improve
Continuous monitoring drives ongoing security.

Implementation Approach

Requirement 6 is best approached as a **secure software lifecycle (SDLC):**

- **Define secure development standards**
  - Coding guidelines aligned with secure practices
  - Peer review and security validation requirements
- **Integrate security into pipelines**
  - Static and dynamic analysis
  - Dependency and vulnerability scanning
  - Automated enforcement of policies
- **Continuously manage vulnerabilities**
  - Track known vulnerabilities across systems and dependencies
  - Apply patches based on risk and severity
- **Protect public-facing applications**
  - Apply application-layer protections
  - Detect and block common attack patterns
- **Enforce structured change management**
  - All changes reviewed, tested, and approved
  - Separation between development and production responsibilities
- **Enforce via IaC and pipelines**
  - Infrastructure and application delivery standardized
  - Prevents ad-hoc or untracked changes

Common Failure Modes

- Treating security as a post-deployment concern
- Lack of visibility into third-party dependencies
- Delayed or inconsistent patching
- Weak or bypassed change control processes
- Public-facing applications exposed without adequate protection

Auditor's Lens

Assessment will focus on:

- Whether secure development practices are defined and followed
- Whether vulnerabilities are identified and remediated in a timely manner
- Whether public-facing applications are protected against common attacks
- Whether changes are controlled, tested, and auditable

AWS Shared Responsibility Considerations

Within AWS, Requirement 6 is **primarily customer responsibility.**

- AWS is responsible for:
  - Security of the underlying infrastructure
  - Patching and maintenance of managed service platforms
- The customer is responsible for:
  - Secure development of custom applications
  - Vulnerability management for workloads and dependencies
  - Configuration and protection of applications
  - Change management processes and deployment controls

Even when using managed services, **application security and lifecycle practices remain fully customer-owned.**

AWS-Only / Applicability Notes

- **Managed services reduce, but do not remove responsibility**
  - AWS handles underlying platform patching
  - Customer must still secure configurations, dependencies, and application logic
- **Serverless environments**
  - Eliminate some infrastructure concerns

- Increase importance of:
  - Dependency security
  - Input validation
  - Identity and access control

- **Public-facing application protection (6.4)**
  - Always applicable in cloud-native architectures
  - Especially critical for APIs, web apps, and external integrations

Requirement 6 is therefore **not about infrastructure security**, but about **how systems are built, maintained, and evolved securely over time.**

Positioning Insight

Requirement 6 is where compliance meets **engineering discipline.**

Organizations that implement it well:

- Catch vulnerabilities early in the lifecycle
- Reduce production risk significantly
- Enable faster, safer delivery

In practice, the strongest outcome comes from treating security as a **continuous development concern**, not a checkpoint before release.

Requirement 7 - Restrict Access to System Components and Cardholder Data by Business Need to Know
PCI DSS v4.0.1 (Controls 7.1 – 7.3)

Control Overview

7.1 – Processes and mechanisms for restricting access to system com ponents and cardholder data by business need to know are defined and understood
7.2 – Access to system components and data is appropriately defined and assigned
7.3 – Access to system components and data is managed via an access control system(s)

What These Controls Mean (Aggregated)

Requirement 7 enforces a single principle:
Access must be granted deliberately, minimally, and only where there is a valid business need.

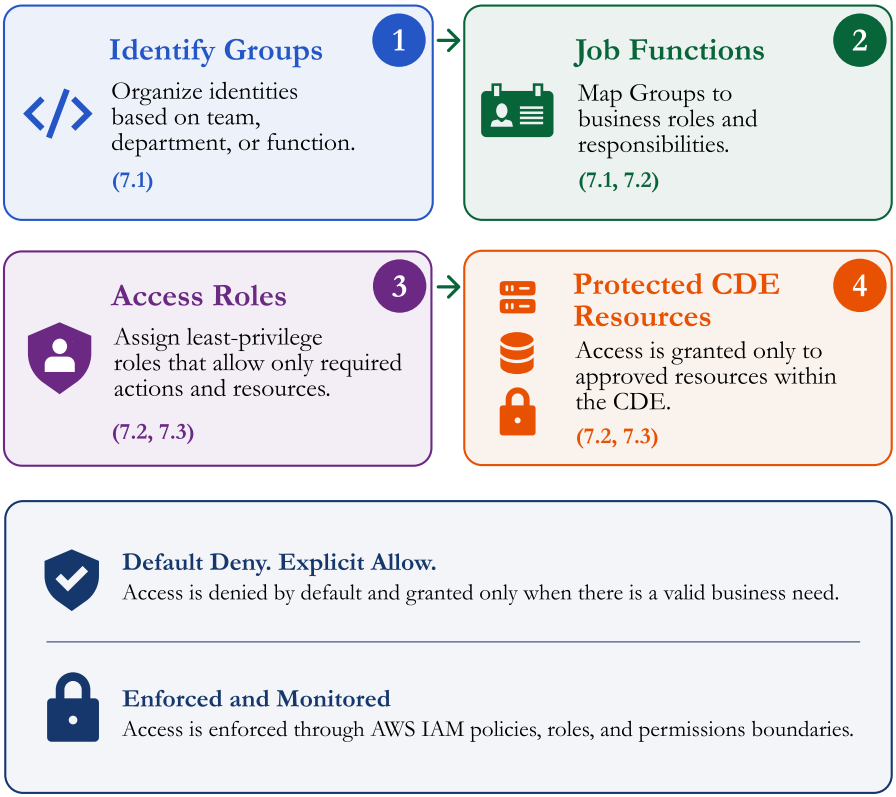
This breaks down into:

- **Governance (7.1)**
  Access control policies, procedures, roles, and ownership must be defined and understood.
- **Least Privilege (7.2)**
  Users, applications, and systems should receive only the access necessary to perform their function.
- **Access Control Enforcement (7.3)**
  Access must be enforced through access control systems that default to deny unless explicitly permitted.

AWS Interpretation

In AWS, Requirement 7 is primarily about **authorization design.** Key expectations:

- **Access must map to job function**
  - Engineers, operators, auditors, support users, applications, and automation should each have clearly defined access boundaries.
- **Least privilege must be intentional**
  - Broad policies such as administrator-style access should be exceptional, time-bound, and heavily monitored.
- **Access control must cover people and systems**
  - Human users, CI/CD roles, service roles, application roles, and cross-account access all matter.
- **Default-deny should be the operating model**
  - Access should be explicitly granted, not inherited through broad groups, wildcard policies, or convenience roles.


1
Identify Groups
Organize identities based on team, department, or function.
(7.1)
2
Job Functions
Map Groups to business roles and responsibilities.
(7.1, 7.2)
3
Access Roles
Assign least-privilege roles that allow only required actions and resources.
(7.2, 7.3)
4
Protected CDE Resources
Access is granted only to approved resources within the CDE.
(7.2, 7.3)
Default Deny. Explicit Allow.
Access is denied by default and granted only when there is a valid business need.
Enforced and Monitored
Access is enforced through AWS IAM policies, roles, and permissions boundaries.

Implementation Approach

Requirement 7 is best approached through **role-based access governance and least-privilege enforcement:**

- Define access models
  - Map job functions to required permissions
  - Separate administrative, operational, read-only, break-glass, and application roles
- Use centralized identity patterns
  - Federated access where possible
  - Role-based access instead of long-lived individual credentials
- Minimize direct access to cardholder data
  - Avoid granting full PAN visibility unless explicitly required
  - Prefer masked views, restricted queries, and controlled operational workflows
- Review access regularly
  - Validate that permissions still match current job responsibilities
  - Remove stale, excessive, or orphaned access
- Enforce via IaC
  - IAM roles, policies, groups, and access boundaries defined declaratively
  - Prevents ad-hoc privilege growth and improves auditability

Common Failure Modes

- Overuse of administrator-level permissions
- Access granted by team membership rather than actual job need
- Stale access after role changes or project completion
- Service roles with broad wildcard permissions
- CI/CD pipelines with excessive production privileges
- Support or analytics users able to view more cardholder data than necessary

Auditor's Lens

Assessment will focus on:

- Whether access control policies are **documented and understood**
- Whether access is assigned based on **business need to know**
- Whether permissions reflect **least privilege**
- Whether access is enforced through a proper **access control system**
- Whether inappropriate or excessive access is identified and removed

AWS Shared Responsibility Considerations

Within AWS, Requirement 7 is **primarily customer responsibility.**

- AWS is responsible for:
  - Security of the underlying infrastructure
  - Access control for AWS-managed internal systems and personnel
- The customer is responsible for:
  - IAM design and permissions
  - User, role, and group assignment
  - Service roles and workload identities
  - Cross-account access
  - Application-level authorization
  - Restricting access to cardholder data within customer systems

AWS provides the access control mechanisms, but **the customer defines who and what is allowed to access the CDE and cardholder data.**

AWS-Only / Applicability Notes

- **IAM is necessary but not sufficient**
  - IAM controls access to AWS resources, but does not automatically solve application-level access to cardholder data.
  - Applications still need their own authorization model where users can view, process, export, or administer cardholder data.

- **Managed services still require customer authorization design**
  - AWS controls the underlying platform, but the customer controls access to databases, storage, logs, dashboards, APIs, and adminis trative interfaces.
- **Service and automation identities are in scope**
  - CI/CD roles, Lambda execution roles, ECS task roles, EC2 in stance profiles, and cross-account roles must follow least privilege just like human users.

Requirement 7 is therefore not simply an "IAM cleanup" exercise. It is a broader authorization model covering **people, systems, services, and applications.**

Positioning Insight

Requirement 7 is about **controlling blast radius.**

Organizations that implement it well:
- Reduce the risk of accidental or malicious data exposure
- Limit what compromised users or workloads can do
- Make access reviews and audits significantly easier

In practice, the strongest access models are boring by design: **clear roles, minimal permissions, explicit approvals, and no mystery access paths.**

Requirement 8 - Identify Users and Authenticate Access to System Components PCI DSS v4.0.1 (Controls 8.1 – 8.6)

Control Overview

8.1 – Processes and mechanisms for identifying users and authenticating access to system components are defined and understood
8.2 – User identification and related accounts for users and administrators are strictly managed throughout an account's lifecycle
8.3 – Strong authentication for users and administrators is established and managed
8.4 – Multi-factor authentication (MFA) is implemented to secure access into the CDE
8.5 – Multi-factor authentication (MFA) systems are configured to prevent misuse
8.6 – Use of application and system accounts and associated authentication factors is strictly managed

What These Controls Mean (Aggregated)

Requirement 8 enforces a single principle:
Every user, administrator, application, and system account must be uniquely identifiable, properly authenticated, and accountable for its actions.

This breaks down into:

- **Governance (8.1)**
  Identity and authentication policies, procedures, responsibilities, and ownership must be defined and maintained.
- **Account Lifecycle Management (8.2)**
  Accounts must be created, modified, disabled, and removed through controlled processes.

- **Strong Authentication (8.3)**
  Access must be protected with appropriate authentication factors such as something known, something possessed, or something inherent.
- **MFA for CDE Access (8.4, 8.5)**
  MFA must protect access into the CDE, and the MFA implementation itself must be resistant to misuse.
- **Application and System Accounts (8.6)**
  Non-human identities must be managed deliberately, with tightly controlled credentials and usage.

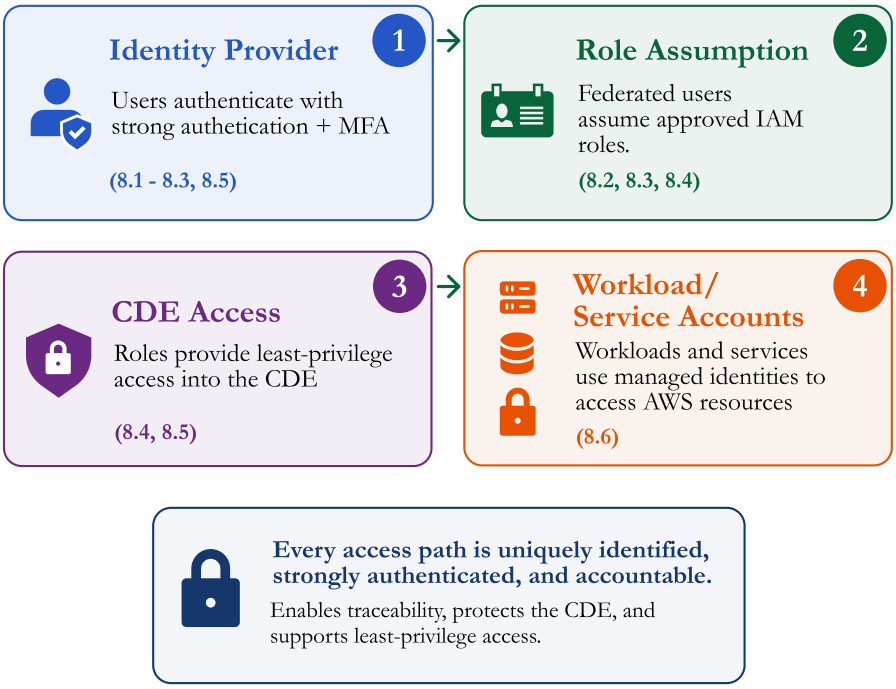
AWS Interpretation

In AWS, Requirement 8 is primarily about **identity assurance and authentication design**, not simply creating IAM users.

Key expectations:

- **Unique identities**
  - Human users should be individually identifiable.
  - Shared administrator accounts should be avoided.
  - Actions must be attributable to a person, process, application, or system identity.
- **Centralized authentication**
  - Federated identity patterns are generally preferable to long-lived local users.
  - Administrative access should flow through controlled identity pro viders and role assumption patterns.
- **MFA for privileged and CDE access**
  - MFA should protect administrative access and access paths into the CDE.
  - Remote access paths that could reach or impact the CDE require particular attention.
- **Service identity discipline**
  - Application roles, workload identities, CI/CD roles, instance pro

files, task roles, and automation credentials are in scope.
  - Non-human identities should not become permanent "invisible admins."
- **Credential lifecycle control**
  - Passwords, keys, tokens, and secrets must be rotated, protected, revoked, and monitored according to defined processes.



Implementation Approach

Requirement 8 is best approached through **centralized identity, strong authentication, and disciplined lifecycle management:**
Use named human identities
Every person gets a unique identity.
Avoid shared accounts except where explicitly justified and controlled.

- Prefer federated access
  - Centralize authentication and MFA enforcement.
  - Use short-lived role/session-based access instead of long-lived static credentials.
- Control privileged access paths
  - Require MFA for administrative and CDE-relevant access.
  - Restrict break-glass access and monitor its use closely.
- Manage non-human identities
  - Assign narrowly scoped roles to workloads.
  - Store secrets securely.
  - Avoid embedding static credentials in code, images, repositories, or configuration files.
- Review and remove stale access
  - Disable accounts when no longer required.
  - Revoke unused access keys, inactive identities, and obsolete automation credentials.
- Enforce via IaC
  - Identity policies, roles, access boundaries, and authentication-related configurations should be defined consistently and reviewed.

Common Failure Modes

- Shared admin accounts with no individual accountability
- IAM users with long-lived access keys used for automation
- MFA enabled for console access but bypassed through CLI/API credentials
- Service accounts with broad permissions and no clear owner

- Stale accounts left active after personnel or vendor changes
- Secrets stored in repositories, build artifacts, container images, or plain environment files
- Break-glass roles that are not monitored or periodically tested

Auditor's Lens

Assessment will focus on:

- Whether identity and authentication policies are **documented and understood**
- Whether users are **uniquely identified** before access is granted
- Whether authentication factors are **strong and properly managed**
- Whether MFA is required for applicable access into the CDE
- Whether MFA systems are configured to prevent bypass or misuse
- Whether application and system accounts are **strictly controlled**

PCI DSS explicitly notes that Requirement 8 applies broadly to accounts on system components, including POS accounts, administrative accounts, system and application accounts, and accounts used to access cardholder data or systems with cardholder data; it does not apply to consumer/cardholder accounts.

AWS Shared Responsibility Considerations

Within AWS, Requirement 8 is **primarily customer responsibility.**

- AWS is responsible for:
  - Security of the underlying AWS infrastructure
  - AWS internal identity and access controls for operating the cloud platform
- The customer is responsible for:
  - IAM, federation, roles, users, policies, and permissions
  - MFA configuration for administrative and CDE-relevant access
  - Workload identities and service accounts
  - Application-level authentication
  - Credential storage, rotation, and revocation

- Monitoring identity activity and suspicious authentication behavior

AWS describes the customer side of the shared model as including customer data, asset classification, and using IAM tools to apply appropriate permissions.

AWS-Only / Applicability Notes

- **Consumer accounts are out of scope for Requirement 8**
  - PCI DSS states that these requirements do not apply to accounts used by consumers/cardholders.
- **POS account exceptions are narrow**
  - Some specific requirements are not intended for POS terminal accounts that access only one card number at a time for a single transaction; this is not a general exemption for AWS workloads or administrative identities.
- **Managed services do not remove identity responsibility**
  - AWS may operate the underlying service platform, but the customer still controls who can access AWS resources, applications, databases, logs, secrets, and operational tooling.
- **IAM is not the whole answer**
  - IAM governs AWS resource access. Applications still need appro priate user authentication, session handling, authorization, and auditability for application-level access to cardholder data.

Positioning Insight

Requirement 8 is about **accountability and credential resilience.** Organizations that implement it well:

- Make every action traceable
- Reduce the blast radius of compromised credentials
- Prevent shared, stale, or invisible accounts from becoming audit and security liabilities

In practice, strong authentication is not just "turning on MFA." It is a full identity model covering **humans, workloads, automation, vendors, and emergency access.**

Requirement 9 - Restrict Physical Access to Cardholder Data
PCI DSS v4.0.1 (Controls 9.1 – 9.5)

Control Overview

9.1 – Processes and mechanisms for restricting physical access to cardholder data are defined and understood
9.2 – Physical access controls manage entry into facilities and systems containing cardholder data
9.3 – Physical access for personnel and visitors is authorized and managed
9.4 – Media with cardholder data is securely stored, accessed, distributed, and destroyed
9.5 – Point of interaction (POI) devices are protected from tampering and unauthorized substitution

What These Controls Mean (Aggregated)

Requirement 9 enforces a single principle:
Cardholder data must be protected from physical access, removal, tampering, and mishandling.

This breaks down into:

- **Governance (9.1)**
  Physical security policies, operational procedures, roles, and responsibilities must be defined and understood.
- **Facility and CDE Access Control (9.2, 9.3)**
  Physical access to facilities, sensitive areas, systems, and CDE locations must be controlled, monitored, authorized, and logged.
- **Media Protection (9.4)**
  Physical and electronic media containing cardholder data must be secured, inventoried, transported carefully, and destroyed when no longer needed.

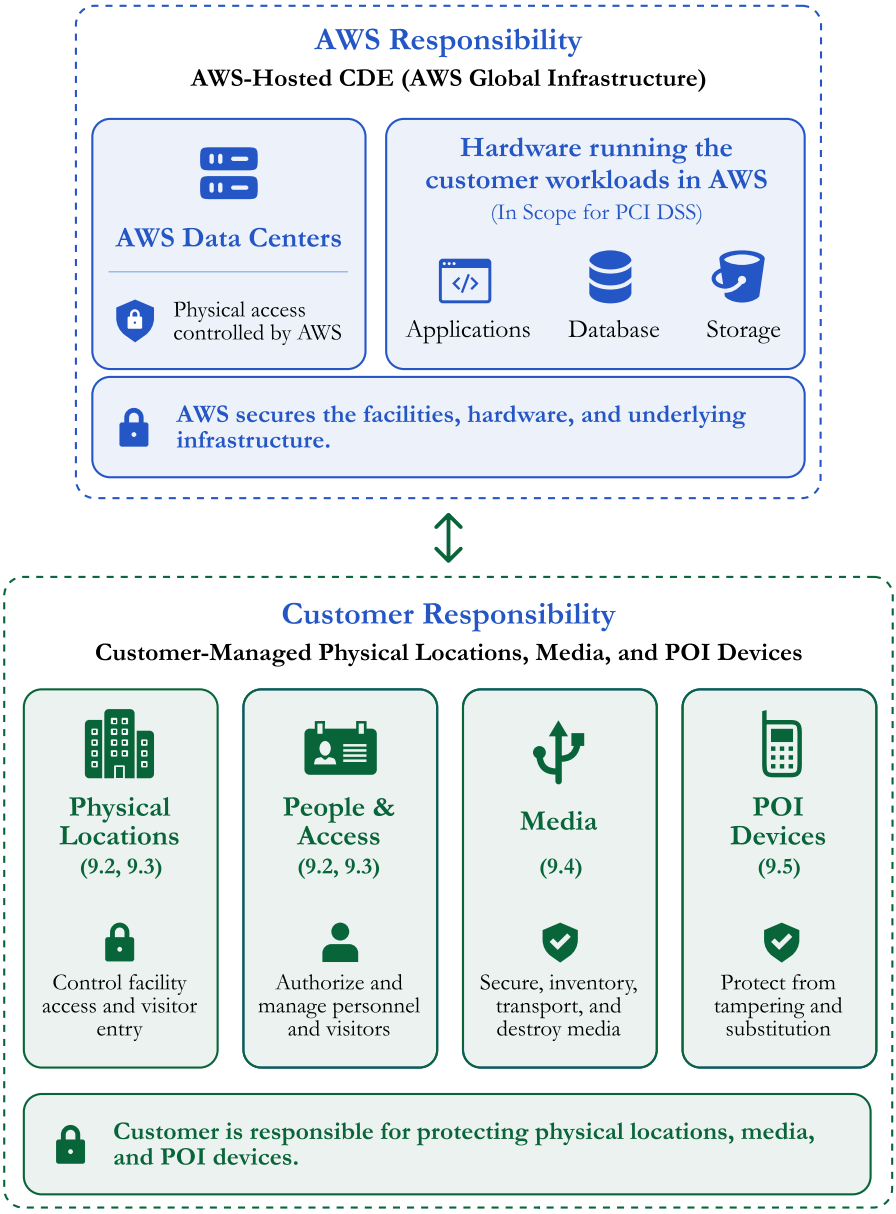
- **POI Device Protection (9.5)**
  Card-present payment devices must be inventoried, inspected, and protected against tampering, substitution, and skimming.

AWS Interpretation

In AWS-native environments, Requirement 9 is where the shared responsibility model becomes especially important.

Key expectations:

- **AWS-managed data centers are AWS responsibility**
  - Customers do not manage physical access to AWS facilities, hardware, racks, or underlying infrastructure.
  - Evidence for these controls typically comes from AWS compliance documentation, not from customer-operated access logs.
- **Customer-controlled physical locations remain in scope**
  - Offices, retail locations, call centers, support centers, or hybrid facilities may still be relevant if they contain cardholder data, systems that access the CDE, paper records, backup media, or POI devices.
- **Physical media still matters**
  - Printed reports, exported files, removable media, laptops, offlin backups, and couriered storage can all bring physical handling require ments into scope.
- **POI devices are environment-specific**
  - For e-commerce-only AWS workloads with no card-present devices, POI device controls are generally not applicable.
  - For retail or hybrid environments, POI inventory, inspection, and staff training become directly relevant.


AWS Responsibility
AWS-Hosted CDE (AWS Global Infrastructure)
AWS Data Centers
Physical access controlled by AWS
Hardware running the customer workloads in AWS
(In Scope for PCI DSS)
Applications
Database
Storage
AWS secures the facilities, hardware, and underlying infrastructure.
Customer Responsibility
Customer-Managed Physical Locations, Media, and POI Devices
Physical Locations
(9.2, 9.3)
Control facility access and visitor entry
People & Access
(9.2, 9.3)
Authorize and manage personnel and visitors
Media
(9.4)
Secure, inventory, transport, and destroy media
POI Devices
(9.5)
Protect from tampering and substitution
Customer is responsible for protecting physical locations, media, and POI devices.

Implementation Approach

Requirement 9 is best approached through **scope clarity and physical control ownership:**

- **Identify physical locations in scope**
  - Offices, data rooms, retail sites, support areas, and storage locations that can access, store, or handle cardholder data.
- **Limit and monitor physical access**
  - Use badge systems, locks, cameras, visitor logs, and restricted areas where relevant.
  - PCI specifically calls for monitoring access to sensitive areas within the CDE, and retaining collected monitoring data for at least three months unless restricted by law.
- **Manage personnel and visitor access**
  - Authorize access based on role and business need.
  - Revoke access immediately upon termination.
  - Ensure visitors are authorized, escorted, identifiable, and logged.
- **Control media lifecycle**
  - Secure, classify, inventory, transport, approve movement, and destroy media containing cardholder data.
  - Electronic media inventories must be maintained and verified at least every 12 months.
- **Protect POI devices where applicable**
  - Maintain an up-to-date device list.
  - Periodically inspect devices for tampering or substitution.
  - Train personnel to detect suspicious behavior or unauthorized maintenance attempts.

Common Failure Modes

- Assuming "we are on AWS" makes all physical security controls irrelevant
- Forgetting about paper records, exports, screenshots, printed reports, and offline backups

- Weak visitor controls in offices or support locations that can access the CDE
- No inventory or review process for removable electronic media
- POI devices deployed in the field without inspection cadence or staff training
- Retaining old hard-copy materials with cardholder data beyond business or legal need

Auditor's Lens

Assessment will focus on:

- Whether physical access policies and responsibilities are **documented and understood**
- Whether access to facilities, systems, and sensitive areas is **restricted and monitored**
- Whether personnel and visitor physical access is **authorized, logged, and revoked when needed**
- Whether media containing cardholder data is **secured, tracked, and destroyed properly**
- Whether POI devices, where present, are **inventoried, inspected, and protected from tampering**

AWS Shared Responsibility Considerations

Within AWS, Requirement 9 is **heavily shared**, and in many AWS-native architectures the physical data center portion is largely handled by AWS.

- AWS is responsible for:
  - Physical access controls for AWS data centers
  - Physical protection of AWS-managed hardware, facilities, racks, and infrastructure
  - Underlying environmental and facility-level security for AWS services

- The customer is responsible for:
  - Physical access to customer offices, workstations, support centers, retail locations, and hybrid environments
  - Paper records and printed cardholder data
  - Removable media and offline backups under customer control
  - POI devices, if used
  - Visitor management and physical access to customer-controlled CDE locations

For cloud-only workloads, the customer should not attempt to "implement" AWS data center controls. Instead, they should rely on AWS compliance evidence while focusing on physical locations, media, and devices they actually control.

AWS-Only / Applicability Notes

- **AWS data center physical controls**
  - These are AWS-controlled and not directly implemented by the customer.
  - The customer should reference AWS compliance artifacts rather than trying to evidence AWS facility controls themselves.
- **Physical facility controls may be reduced in AWS-native environments**
  - If the organization has no customer-managed facility containing systems, media, or access paths to cardholder data, large parts of 9.2 and 9.3 may have limited applicability to the customer environment.
  - This must be validated through scoping, not assumed.
- **Media controls can still apply**
  - Requirement 9.4 can apply even in cloud environments if card holder data exists in printed form, removable media, offline exports, local backups, or files distributed outside AWS.
    to components used only for manual PAN key entry or certain

- **POI controls are conditional**
  - Requirement 9.5 applies to deployed POI devices used in card-present transactions. The standard notes that these requirements apply to POI devices used with a payment card form factor such as swiped, tapped, or dipped cards, and do not apply to components used only for manual PAN key entry or certain commercial off-the-shelf devices.

Positioning Insight

Requirement 9 is about not letting physical reality undermine cloud security.

Organizations that implement it well:

- Avoid accidental scope expansion through offices, media, and support processes
- Keep AWS responsibility separate from customer responsibility
- Reduce risk from lost media, uncontrolled visitors, and tampered payment devices

In practice, Requirement 9 should start with a simple question: **where can cardholder data be physically touched, seen, removed, or tampered with?**

Requirement 10 - Log and Monitor All Access to System Components and Cardholder Data
PCI DSS v4.0.1 (Controls 10.1 – 10.7)

Control Overview

10.1 – Processes and mechanisms for logging and monitoring all access to system components and cardholder data are defined and understood
10.2 – Audit logs are implemented to support the detection of anomalies and suspicious activity, and the forensic analysis of events
10.3 – Audit logs are protected from destruction and unauthorized modifications
10.4 – Audit logs are reviewed to identify anomalies or suspicious activity
10.5 – Audit log history is retained and available for analysis
10.6 – Time-synchronization mechanisms support consistent time settings across all systems
10.7 – Failures of critical security control systems are detected, reported, and responded to promptly

What These Controls Mean (Aggregated)

Requirement 10 enforces a single principle:
Security-relevant activity must be captured, protected, retained, reviewed, and usable for investigation.

This breaks down into:

- **Governance (10.1)**
  Logging and monitoring policies, procedures, roles, and responsibilities must be documented and understood.
- **Auditability (10.2, 10.3, 10.5)**
  Logs must exist, contain useful events, be protected from tampering,

- **Operational Monitoring (10.4, 10.7)**
  Logs and security-control failures must be reviewed, alerted on, and acted upon.
- **Forensic Reliability (10.6)**
  Time must be synchronized so events can be correlated across systems.

AWS Interpretation

In AWS, Requirement 10 is primarily about **centralized observability and accountable activity tracking.**

Key expectations:

- **Capture activity across all relevant layers**
  - AWS control-plane events
  - Network activity
  - Workload logs
  - Application logs
  - Database and storage access events
  - Security-tool events
- **Centralize logs quickly**
  - Logs should not live only on the systems that generate them
  - Centralized logging reduces the risk of deletion or modification after compromise
- **Protect logs as sensitive records**
  - Access to logs must be restricted
  - Logs may contain sensitive operational detail and, if poorly designed, may accidentally contain account data
- **Review meaningful events**
  - Daily review is required for security events, CHD/SAD systems, critical system components, and systems performing security functions
  - Other in-scope system logs are reviewed periodically based on risk analysis

- **Make time consistent**
  Time synchronization is essential for correlating events across AWS accounts, applications, services, and external systems

Implementation Approach

Requirement 10 is best approached through **centralized logging, controlled retention, and actionable monitoring:**

- **Define the logging standard**
  - Required events
  - Required fields
  - Retention periods
  - Ownership and review responsibilities
- **Centralize log collection**
  - Aggregate infrastructure, application, access, network, and security logs into a controlled logging environment
  - Separate log storage from source systems where possible
- **Protect log integrity**
  - Restrict modification and deletion
  - Use immutable or write-protected storage patterns where appro priate
  - Monitor changes to logs and logging configurations
- **Review and alert**
  - Prioritize security events and critical components
  - Use automated alerting for suspicious activity and control failures
  - Ensure exceptions and anomalies are investigated, not merely recorded
- **Retain logs appropriately**
  - Maintain at least 12 months of audit log history
  - Keep at least the most recent three months immediately available for analysis
- **Enforce via IaC**
  - Logging configuration, retention policies, monitoring rules, and alerting paths should be defined consistently and reviewed.

Common Failure Modes

- Logs exist but are scattered across services and accounts
- Application logs lack user identity, request context, or useful event detail
- Logs can be modified or deleted by the same roles that generate them
- Security events are collected but not reviewed
- Alerts fire but have no clear owner or response path
- Retention is too short for investigation or audit needs
- Time drift makes cross-system incident reconstruction unreliable
- Sensitive account data accidentally appears in logs

Auditor's Lens

Assessment will focus on:

- Whether logging and monitoring procedures are **documented and assigned**
- Whether audit logs are **enabled and active for system components and cardholder data**
- Whether logs capture access to cardholder data and administrative actions
- Whether logs are **protected from unauthorized modification or destruction**
- Whether required logs are **reviewed daily or periodically as applicable**
- Whether audit log history is **retained for at least 12 months**, with recent history available
- Whether system time is **synchronized and protected**
- Whether failures of critical security controls are **detected, reported, and addressed**

AWS Shared Responsibility Considerations

Within AWS, Requirement 10 is **shared**, but the customer owns the logging and monitoring outcome for their environment.

- AWS is responsible for:
  - Logging and monitoring of AWS-operated infrastructure and internal platform operations
  - Security of the underlying services that provide logging capabilities
- The customer is responsible for:
  - Enabling logs for AWS services and workloads
  - Capturing application and database access events
  - Centralizing and protecting logs
  - Defining retention and review processes
  - Monitoring security events and responding to anomalies
  - Ensuring time synchronization across customer-managed systems

AWS provides many of the mechanisms, but PCI DSS compliance depends on whether the customer has configured them correctly and built a real review-and-response process around them.

AWS-Only / Applicability Notes

- **Managed services still require customer logging configuration**
  - AWS may operate the service, but the customer usually controls whether relevant access, query, application, or data-plane logs are enabled.
- **Control-plane logging is not enough**
  - AWS API activity is important, but Requirement 10 also expects visibility into access to system components and cardholder data. Application and data-layer logging still matter.
- **Logs must be protected from the workloads they monitor**
  - A compromised workload should not be able to erase or alter its own audit history. Centralized, restricted log storage is the safer operating pattern.

- **Service-provider timing matters**
  - Some 10.7 requirements were service-provider-only or best-prac tice until 31 March 2025, but the standard states that 10.7.2 applies to all entities after that date and includes failures of audit log review mechanisms and automated security testing tools, if used.

Positioning Insight

Requirement 10 is about **making security events visible and defensible.**

Organizations that implement it well:

- Detect suspicious activity sooner
- Preserve evidence for forensic analysis
- Reduce the chance that attackers can hide their tracks
- Turn logs from passive records into operational security controls

In practice, the weakest logging programs are not missing logs entirely; they are drowning in logs that nobody owns, reviews, protects, or understands.

Requirement 11 - Test Security of Systems and Networks Regularly
PCI DSS v4.0.1 (Controls 11.1 – 11.6)

Control Overview

11.1 – Processes and mechanisms for regularly testing security of systems and networks are defined and understood
11.2 – Wireless access points are identified and monitored, and unauthorized wireless access points are addressed
11.3 – External and internal vulnerabilities are regularly identified, prioritized, and addressed
11.4 – External and internal penetration testing is regularly performed, and exploitable vulnerabilities and security weaknesses are corrected
11.5 – Network intrusions and unexpected file changes are detected and responded to
11.6 – Unauthorized changes on payment pages are detected and responded to

What These Controls Mean (Aggregated)

Requirement 11 enforces a single principle:
Security controls must be tested continuously enough to prove they still work.

This breaks down into:

- **Governance (11.1)**
  Security testing processes, procedures, ownership, and responsibilities must be documented and understood.
- **Wireless Discovery (11.2)**
  Wireless access points must be identified and unauthorized access points must be detected and handled.

- **Vulnerability Testing (11.3)**
  Internal and external vulnerability scans must be performed regularly, after significant changes, and findings must be remediated.
- **Penetration Testing (11.4)**
  Internal, external, application-layer, network-layer, and segmentation testing must validate whether systems can actually be exploited.
- **Detection Controls (11.5)**
  Intrusions and unauthorized changes to critical files must generate actionable alerts.
- **Payment Page Integrity (11.6)**
  Payment pages must be monitored for unauthorized changes, especially browser-delivered scripts and HTTP headers.

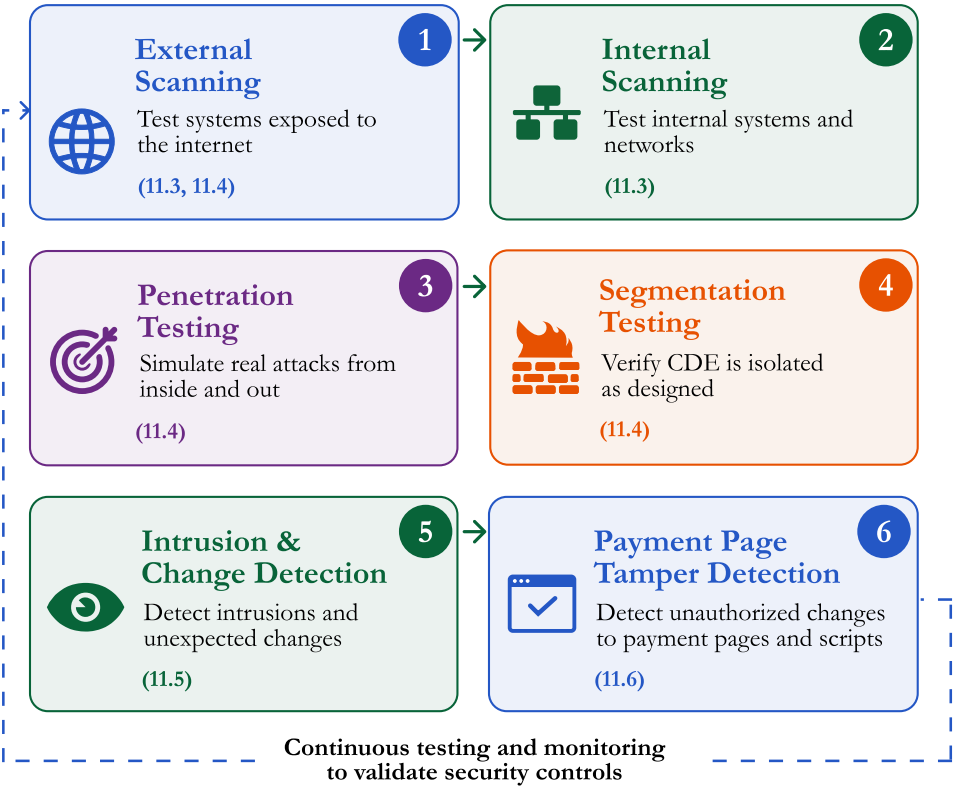
AWS Interpretation

In AWS, Requirement 11 is about **validating the actual security posture of the environment**, not assuming that cloud-native controls are correctly configured.

Key expectations:

- **Test from the right perspective**
  - Internal scans validate risk inside the environment.
  - External scans validate internet-facing exposure.
  - Penetration tests validate realistic attacker paths.
- **Scan after significant change**
  - Infrastructure, application, network, or segmentation changes can introduce new exposure.
  - Testing should be part of the change lifecycle, not an annual after thought.
- **Validate segmentation**
  - If CDE isolation is used to reduce PCI scope, it must be technically proven.
  - Segmentation testing confirms that out-of-scope systems cannot reach the CDE.

- **Detect unauthorized modification**
  - Critical files, configuration, application artifacts, and payment page components must be monitored for unexpected change.
- **Treat browser-side payment pages as part of the attack surface**
  - Client-side scripts, tag managers, redirects, and HTTP headers can become skimming vectors.
  - Requirement 11.6 specifically focuses on detecting unauthorized changes to payment pages as received by the consumer browser.



Implementation Approach

Requirement 11 is best approached through **scheduled testing, change-triggered testing, and continuous detection:**

- **Define a formal testing program**
  - Scope, frequency, roles, evidence, and remediation ownership.
  - Include vulnerability scanning, penetration testing, segmentation validation, and detection control testing.
- **Run internal vulnerability scans**
  - At least every three months and after significant changes.
  - Prioritize and resolve high-risk and critical findings.
  - Use authenticated scanning where applicable; the standard notes authenticated scanning is not applicable to components that can not accept credentials, with examples including some network/security appliances, mainframes, and containers.
- **Run external vulnerability scans**
  - At least every three months.
  - Performed by a PCI SSC Approved Scanning Vendor (ASV).
  - Findings must be resolved and rescanned as needed to meet passing-scan requirements.
- **Perform penetration testing**
  - Use a documented methodology.
  - Cover the CDE perimeter, critical systems, internal and external testing, application-layer testing, network-layer testing, and segmentation/scope-reduction controls.
  - Retain test and remediation results for at least 12 months.
- **Validate segmentation**
  - If segmentation is used to isolate the CDE, test it at least every 12 months and after changes to segmentation controls.
  - For service providers, segmentation testing is required at least every six months and after changes.
- **Deploy intrusion and change detection**
  - Use IDS/IPS, network detection, file integrity monitoring, or equivalent mechanisms where appropriate.
  - Critical file comparisons must be performed at least weekly.

- **Monitor payment page integrity**
  - Detect unauthorized modification of security-impacting HTTP headers and script contents of payment pages as received by the consumer browser.
  - Run detection at least weekly or at a frequency defined by targeted risk analysis.
- **Enforce via IaC**
  - Scanning targets, logging hooks, security tooling, alerting rules, and monitoring configuration should be consistently defined and reviewed.

Common Failure Modes

- Treating vulnerability scanning as equivalent to penetration testing
- Running scans but not remediating or rescanning
- Forgetting to scan after significant infrastructure or application changes
- External scans missing assets due to incomplete public asset inventory
- Assuming segmentation works without testing it
- Detection tooling deployed but not alerting to the right people
- File integrity monitoring applied only to servers, not application critical artifacts
- Payment pages using third-party scripts without tamper/change monitoring

Auditor's Lens

Assessment will focus on:

- Whether security testing policies and procedures are **documented and owned**
- Whether wireless access points are **identified, monitored, and un authorized devices addressed**
- Whether internal and external vulnerability scans are **performed on schedule and after significant changes**

- Whether external scans are performed by an **Approved Scanning Vendor**
- Whether vulnerabilities are **prioritized, remediated, and rescanned**
- Whether penetration testing is **methodology-driven, scoped correctly, and retained as evidence**
- Whether segmentation, if used, is **technically validated**
- Whether intrusions, critical file changes, and payment page tampering are **detected and responded to**

AWS Shared Responsibility Considerations

Within AWS, Requirement 11 is **shared**, but the customer owns the testing program for their AWS environment.

- AWS is responsible for:
  - Security testing and assurance for the underlying AWS-managed infrastructure
  - Platform-level controls for managed services
  - Providing mechanisms and policies that allow customer-side security testing within permitted boundaries
- The customer is responsible for:
  - Testing customer workloads, applications, configurations, and net work exposure
  - Running internal and external vulnerability scans against in-scope assets
  - Coordinating ASV scans for externally exposed in-scope endpoints
  - Performing penetration testing within allowed AWS rules of engagement
  - Validating CDE segmentation
  - Implementing intrusion detection, file integrity monitoring, and payment page tamper detection
  - Remediating findings and retaining evidence

AWS reduces the amount of infrastructure the customer must test directly, but it does not remove the need to test the customer's architecture, applications, configurations, and exposure.

AWS-Only / Applicability Notes

- **Wireless controls may be limited in AWS-native environments**
  - 11.2 can have reduced applicability for cloud-only workloads with no wireless network connected to the CDE.
  - It remains relevant for offices, retail sites, call centers, hybrid networks, or any wireless environment connected to systems that can impact the CDE.
- **Some scan types depend on service model**
  - EC2-based workloads are more directly scannable.
  - Containers, serverless, and managed services may require adapted approaches such as image scanning, configuration scanning, application testing, and service-level security review.
- **External ASV scans still apply to exposed in-scope endpoints**
  - Internet-facing APIs, load balancers, web applications, VPN end points, and other public attack surfaces may require ASV scanning if they are in scope.
- **Penetration testing must respect AWS boundaries**
  - Customer-owned workloads can be tested, but testing must avoid prohibited activity and must not impact other tenants or AWS-managed infrastructure.
- **Payment page tamper detection may apply even when payment processing is outsourced**
  - If the merchant page includes third-party payment scripts, redirects, or iframe-based payment flows, browser-side tamper detection may still be relevant under 11.6. The standard explicitly focuses on the payment page as received by the consumer browser.

Positioning Insight

Requirement 11 is about **proving security controls work in practice.**

Organizations that implement it well:

- Catch exposure before attackers do
- Validate that segmentation actually reduces scope
- Turn security testing into part of normal engineering operations
- Detect unauthorized changes before they become breaches

In practice, Requirement 11 is where "we believe this is secure" becomes "we have tested and can prove it."

Requirement 12 - Support Information Security with Organizational Policies and Programs
PCI DSS v4.0.1 (Controls 12.1 – 12.10)

Control Overview

12.1 – A comprehensive information security policy that governs and provides direction for protection of the entity's information assets is known and current
12.2 – Acceptable use policies for end-user technologies are defined and implemented
12.3 – Risks to the cardholder data environment are formally identified, evaluated, and managed
12.4 – PCI DSS compliance is managed
12.5 – PCI DSS scope is documented and validated
12.6 – Security awareness education is an ongoing activity
12.7 – Personnel are screened to reduce risks from insider threats
12.8 – Risk to information assets associated with third-party service provider relationships is managed
12.9 – Third-party service providers support their customers' PCI DSS compliance
12.10 – Suspected and confirmed security incidents that could impact the CDE are responded to immediately

What These Controls Mean (Aggregated)

Requirement 12 enforces a single principle:
PCI DSS must be operated as an ongoing security program, not treated as a once-a-year audit exercise.

This breaks down into:

- **Policy and Governance (12.1, 12.2, 12.4)**
  Security testing processes, procedures, ownership, and responsibilities must be documented and understood.
- **Risk and Scope Management (12.3, 12.5)**
  CDE risk, scope, data flows, system components, segmentation, and third-party access must be periodically reviewed and kept accurate.
- **People Controls (12.6, 12.7)**
  Personnel must understand their responsibilities, receive ongoing awareness training, and be screened where appropriate.
- **Third-Party Governance (12.8, 12.9)**
  Third-party service providers must be identified, assessed, contractually managed, and mapped against PCI DSS responsibilities.
- **Incident Response (12.10)**
  Security incidents that could affect the CDE must have a ready, tested, and actionable response process.

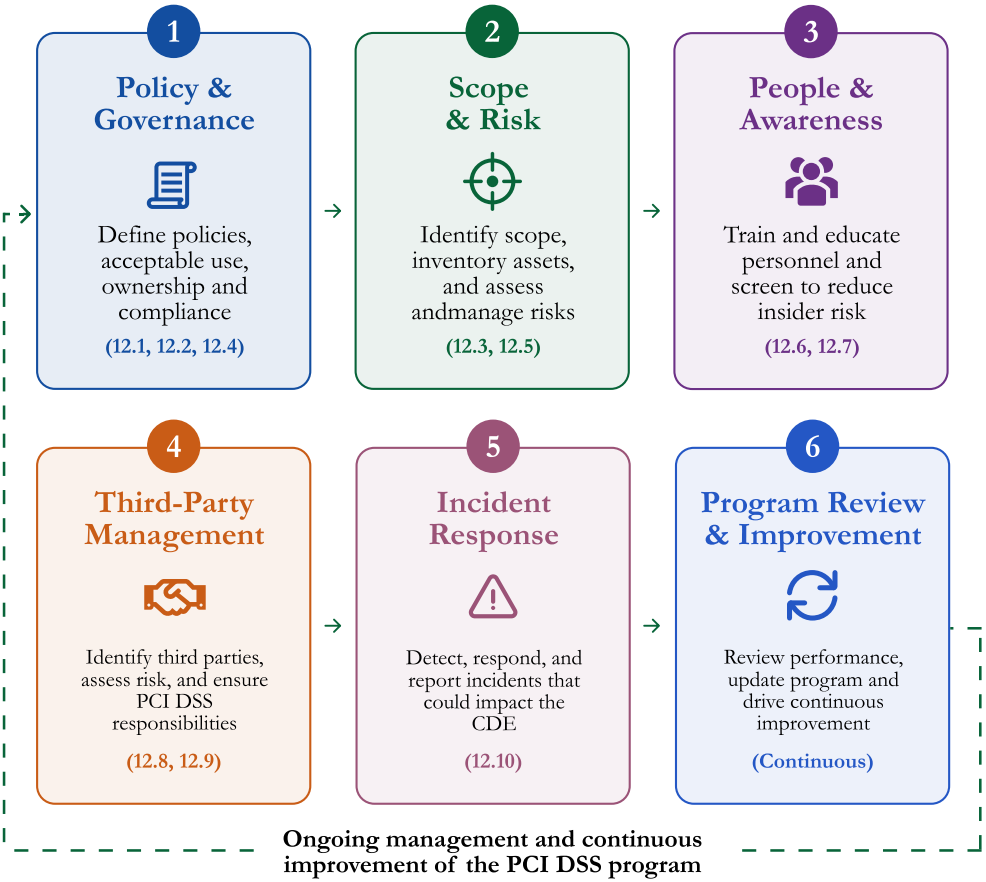
AWS Interpretation

In AWS, Requirement 12 is less about a specific AWS service and more about **operating discipline around the cloud environment.**

Key expectations:

- **Policy must match cloud reality**
  - Security policies must reflect how AWS accounts, workloads, identities, pipelines, logs, and third-party services are actually used.
  - A generic on-prem policy copied into a cloud environment will usually fail operationally.
- **Scope must be continuously maintained**
  - AWS environments change quickly.
  - New accounts, VPCs, services, APIs, data stores, integrations, and pipelines can alter PCI scope.

- **Cloud assets must be inventoried**
  - In-scope system components need a maintained inventory, including their function and use.
  - PCI DSS explicitly calls for maintaining a current inventory of in-scope system components.
- **Risk analysis must drive frequency-based controls**
  - Where PCI DSS allows an entity-defined frequency, the targeted risk analysis must be documented, reviewed at least every 12 months, and updated when needed.
- **Third-party cloud services must be governed**
  - AWS, payment processors, SaaS tools, observability vendors, support vendors, and managed service providers may all be TPSPs depending on what they touch or can impact.
  - PCI DSS includes IaaS, PaaS, SaaS, and FaaS cloud providers as examples of TPSPs that may need to be managed.



Implementation Approach

Requirement 12 is best approached as a compliance operating model:

- **Create a living security policy framework**
  - Overall information security policy.
  - Acceptable use policy for end-user technologies.
  - Cloud-specific standards for identity, logging, data handling, incident response, and deployment workflows.

- **Maintain an accurate PCI scope**
  - Track accounts, workloads, data stores, data flows, third-party connections, and segmentation boundaries.
  - Confirm scope at least every 12 months and after significant changes; for service providers, the standard also includes additional six-month scope validation expectations.
- **Run targeted risk analysis where required**
  - Document protected assets, relevant threats, likelihood/impact factors, and justification for chosen control frequencies.
  - Review targeted risk analyses at least annually.
- **Operationalize awareness**
  - Train personnel upon hire and at least annually.
  - Include phishing, social engineering, and acceptable use topics.
  - PCI DSS distinguishes user awareness training for phishing/social engineering from technical anti-phishing controls.
- **Manage TPSPs deliberately**
  - Maintain a list of TPSPs and services provided.
  - Keep written agreements with responsibility acknowledgments.
  - Perform due diligence before engagement.
  - Monitor TPSP PCI DSS compliance status at least annually.
  - Maintain a responsibility matrix showing which PCI DSS requirements are customer-owned, TPSP-owned, or shared.
- **Prepare incident response before it is needed**
  - Define roles, responsibilities, communications, escalation paths, and stakeholder notifications.
  - Ensure the plan is ready for suspected or confirmed incidents that could affect the CDE.
- **Enforce through IaC and operating controls**
  - Use IaC to standardize accounts, logging, security baselines, and monitoring.
  - Use operational procedures to maintain evidence, approvals, risk reviews, and incident-response records.

Common Failure Modes

- Policies exist but do not reflect how AWS is actually operated
- PCI scope is documented once and then allowed to drift
- New services, integrations, or pipelines are added without reassessing scope
- No maintained inventory of in-scope AWS resources
- TPSP responsibility matrices are missing, vague, or outdated
- AWS is treated as "compliant, therefore we are compliant"
- Security awareness training is generic and not tied to cardholder data responsibilities
- Incident response plans exist but are not tested, assigned, or usable under pressure

Auditor's Lens

Assessment will focus on:

- Whether the information security policy is **established, published, maintained, and disseminated**
- Whether acceptable use policies are **defined and implemented**
- Whether targeted risk analyses are **documented and reviewed**
- Whether PCI DSS responsibility and accountability are **actively managed**
- Whether PCI scope is **documented, validated, and updated after significant changes**
- Whether personnel receive **security awareness training and acknowledgments**
- Whether relevant personnel screening is performed
- Whether TPSPs are **identified, contracted, assessed, monitored, and mapped to PCI responsibilities**
- Whether incident response is **ready, actionable, and aligned to CDE-impacting incidents**

AWS Shared Responsibility Considerations

Within AWS, Requirement 12 is **strongly shared from an accountability perspective**, but most program operation remains the customer's responsibility.

- AWS is responsible for:
  - Security and compliance of the underlying AWS cloud infrastructure
  - Maintaining AWS-side control evidence for the services and infrastructure AWS operates
  - Providing compliance artifacts through AWS compliance programs and AWS Artifact
- The customer is responsible for:
  - Defining and operating the PCI compliance program
  - Maintaining policies, scope, risk analyses, inventories, and incident response processes
  - Managing AWS as a TPSP where applicable
  - Understanding which controls AWS satisfies, which controls the customer satisfies, and which are shared
  - Ensuring customer workloads, configurations, identities, data flows, and third-party integrations meet PCI DSS

AWS describes security and compliance as a shared responsibility: AWS manages and controls infrastructure from the host operating system and virtualization layer down to physical facilities, while customers retain responsibility for guest OS management, applications, and configuration of AWS-provided controls such as security groups.

AWS-Only / Applicability Notes

- **AWS compliance does not equal customer PCI compliance**
  - AWS may provide compliant infrastructure and control evidence, but the customer must still validate its own architecture, configuration, policies, data flows, and operational controls.

- **AWS should be treated as a TPSP where relevant**
  - If AWS services store, process, transmit, or could affect account data, the customer needs to understand AWS's role in the responsibility model and retain appropriate evidence.
- **Service-provider-only requirements may not apply to merchants**
  - Several Requirement 12 controls include additional service-provider-only obligations, including executive-level PCI program accountability, quarterly operational reviews, more frequent scope validation, and TPSP customer-support obligations. These must be included if the assessed entity is a service provider.
- **Cloud-native change velocity increases scope risk**
  - IaC, CI/CD, ephemeral workloads, containers, serverless services, and new AWS integrations can change PCI scope quickly. Requirement 12 is the program layer that keeps that change from becoming invisible.

Positioning Insight

Requirement 12 is about **making PCI DSS sustainable.**

Organizations that implement it well:

- Keep compliance aligned with real architecture
- Prevent scope drift
- Make third-party responsibility explicit
- Prepare teams to respond before an incident happens
- Turn PCI from a yearly evidence scramble into a manageable operating model

In practice, Requirement 12 is where compliance becomes either a paper exercise or a functioning security program.

PCI 3DS Considerations in AWS

PCI DSS and PCI 3DS are related, but they are not the same standard. PCI DSS focuses on protecting environments that store, process, or transmit payment account data. PCI 3DS focuses specifically on securing environments involved in EMV® 3-D Secure authentication flows for card-not-present transactions. The PCI 3DS Core Security Standard applies to environments where specific 3DS functions are performed, including 3DS Server, Access Control Server, and Directory Server functions.

For organizations building or operating payment platforms in AWS, this distinction matters. A system can be relevant to PCI DSS because it handles cardholder data, while also being relevant to PCI 3DS because it participates in authentication workflows. Conversely, an organization may outsource most payment data handling and still have responsibilities around the integrity, security, and operation of its 3DS integration.

How PCI 3DS Differs from PCI DSS

PCI DSS is broad. It covers account data protection, network security, access control, logging, vulnerability management, physical security, incident response, and security governance.

PCI 3DS is narrower and more specialized. It is concerned with the security of 3-D Secure components, authentication flows, transaction messages, cryptographic handling, and the systems that support card-not-present authentication. PCI SSC also maintains a separate PCI 3DS SDK Security Standard for 3DS software development kits used in mobile-based 3DS transactions.

The practical difference is that PCI DSS asks, "How do you protect payment account data?" PCI 3DS asks, "How do you protect the systems and software involved in the 3-D Secure authentication process?"

AWS Interpretation

In AWS, PCI 3DS considerations usually concentrate around a smaller set of architectural concerns:

Secure API endpoints must be designed for strong authentication, encryption in transit, controlled access, and reliable logging. 3DS-related workloads should be isolated from unrelated systems, especially where they process sensitive authentication messages or interact with payment ecosystem participants. Cryptographic operations and key material must be handled carefully, with appropriate separation of duties and lifecycle controls.

Because 3DS flows are highly integration-driven, third-party dependencies are especially important. Payment processors, issuing banks, acquiring banks, Directory Servers, Access Control Servers, SDK vendors, fraud tooling, and hosted payment components may all affect the security and responsibility model. The PCI 3DS Core and PCI DSS standards are independent, but environments can overlap, so responsibility boundaries should be explicitly documented.

Additional Considerations

PCI 3DS should not be treated as a minor add-on to PCI DSS. It introduces its own validation expectations, evidence requirements, and ecosystem responsibilities. Organizations should understand whether they are operating 3DS components directly, embedding a third-party SDK, integrating with a provider, or fully outsourcing the function

In AWS, the key considerations are usually:

Secure isolation of 3DS-related services, strong TLS configuration, careful key and certificate management, robust logging, clear third-party responsibility mapping, and disciplined change control around authentication flows. If mobile SDKs are involved, the SDK security model and validation status should also be reviewed separately from the backend cloud environment.

Positioning Insight

PCI 3DS is best approached as a specialized authentication-security layer that sits alongside PCI DSS, not underneath it.

Organizations that handle it well:

- Avoid confusing PCI DSS validation with PCI 3DS validation
- Clarify responsibility across payment ecosystem participants
- Reduce risk in card-not-present authentication flows
- Prevent third-party integrations from becoming unexamined trust paths

In practice, PCI 3DS work should start with a clear responsibility map. Once ownership is understood, the technical work becomes much easier to scope, design, and evidence.

How Tarmac Can Help

PCI DSS and PCI 3DS readiness is not only a compliance exercise. In AWS, it is an engineering problem that touches architecture, identity, networking, application delivery, data handling, monitoring, third-party integrations, and day-to-day operations.

Tarmac helps organizations turn PCI requirements into practical AWS implementation work. The focus is not on producing paperwork alone, but on designing and operating environments that are easier to secure, easier to audit, and easier to maintain over time.

For organizations starting their PCI journey, Tarmac can help assess the current AWS environment, identify where cardholder data flows, clarify the Cardholder Data Environment, and determine which systems, services, teams, and third parties are likely to fall into scope. This early scoping work is often one of the highest-value steps, because poor scope definition can make the compliance effort larger, more expensive, and more difficult than necessary.

For teams already operating payment workloads in AWS, Tarmac can support architecture reviews, gap assessments, remediation planning, and implementation. This may include network segmentation, secure configuration baselines, IAM and access control improvements, encryption and key management patterns, logging and monitoring design, vulnerability management workflows, secure CI/CD practices, and infrastructure-as-code standardization.

Tarmac can also help organizations prepare for audit conversations by improving evidence readiness. That means making sure architecture diagrams, data-flow diagrams, system inventories, access reviews, change records, logging evidence, vulnerability reports, incident response processes, and third-party responsibility mappings are aligned with the way the AWS environment actually works.

For PCI 3DS, Tarmac can help clarify how 3DS responsibilities interact with the broader PCI DSS environment. This includes reviewing AWS-hosted authentication components, payment integrations, third-party dependencies, secure API patterns, logging, encryption, key handling, and operational controls around card-not-present authentication flows.

The goal is to help organizations move from fragmented compliance activity to a more mature operating model: clear scope, secure architecture, repeatable implementation, defensible evidence, and ongoing control ownership.

Tarmac does not replace the role of a Qualified Security Assessor or the authority of the PCI standards. Instead, it helps engineering and leadership teams get the environment ready, close technical gaps, and build the operational discipline needed to sustain compliance beyond the audit window.

For organizations building or modernizing payment systems on AWS, that is where the real value lies: not just passing an assessment, but creating a secure, maintainable cloud platform that supports the business with less friction and less uncertainty.