


# Agents in the CDE

## How AI Is Redrawing the PCI DSS Scope Boundary

# Christabel Oderele

**PCI Lead, Checkout.com**



# Agenda

What we will cover in this session

- **The Problem** why AI agents do not stay inside the boundary you drew for them
- **Where Agents Live** the points in a payment stack that already touch cardholder data
- **Threats and Scope** the four actors to defend against, and what your tools can actually reach
- **Keeping PAN Away From the Model** the one design decision that removes most of the problem
- **Scoping and Controls** where PCI DSS runs out of road, a framework, and controls that matter
- **A Harder Question** whether the standard itself can still carry this weight

20 YEARS OF SECURING PAYMENT DATA PCI

# The Problem

## AI agents do not stay where you put them



# Traditional Software vs. AI Agents

The assumptions that no longer hold

| Traditional Software | AI Agents |
| --- | --- |
| Defined system boundaries | Dynamic, cross-system traversal |
| Predictable data flows | Runtime data access decisions |
| Static component inventory | Tool calls and memory interfaces |
| Deterministic behaviour | Non-deterministic processing paths |

20 YEARS OF SECURING PAYMENT DATA PCI

# What Makes an AI Agent Different

Four capabilities that together change the scoping problem

- **01 Perceive** reads data from systems, APIs, documents, and context windows
- **02 Reason** decides autonomously what action to take next
- **03 Act** invokes tools, calls APIs, writes to systems, triggers downstream flows
- **04 Persist** retains memory across sessions; state may span multiple systems

20 YEARS OF SECURING PAYMENT DATA PCI

# Why Agentic AI Breaks Traditional Assumptions

Five architectural properties PCI DSS scoping does not anticipate

- Tool-calling interfaces create access points into CHD systems that no static architecture diagram shows
- Model non-determinism means the same input can drive different actions on different runs
- Memory and context persistence retain data across sessions, often with no defined deletion point
- Third-party model providers extend the trust boundary to infrastructure you do not operate
- Autonomous decisions remove the human-initiated access step much of the standard assumes

20 YEARS OF SECURING PAYMENT DATA
PCI

# Where Agents Live

## Already embedded in your payment stack



# Agents in the Payment Stack: Customer-Facing

Three flows where agents touch cardholder data in real time

- **Payment Orchestration** agents route transactions, select acquirers, and retry failed payments, accessing CHD throughout
- **Fraud Decisioning** agents query cardholder history, device data, and behavioural signals to approve or decline
- **Customer Authentication** agents manage step-up authentication flows, consuming identity and payment credentials in context

20 YEARS OF SECURING PAYMENT DATA PCI

# Agents in the Payment Stack: Back Office

Three flows that are easy to leave out of a scope diagram

- **Reconciliation** agents read transaction records, match settlements, and flag discrepancies across multiple data stores
- **Dispute and Chargeback Handling** agents gather evidence, draft responses, and call card scheme APIs using transaction data
- **Compliance Monitoring** agents scan transaction logs for anomalies, ingesting CHD to do so even if only in memory

20 YEARS OF SECURING PAYMENT DATA PCI

# The Threat Model

Four things you are defending against, not one

- **External Attacker** influences the agent through untrusted content it ingests. The primary case, and the one prompt injection describes
- **Malicious Insider** uses the agent as a privilege escalation path, borrowing permissions they do not hold themselves
- **Plain Misconfiguration** over-broad data access with no adversary at all. The most common of the four in practice
- **The Over-Enthusiastic Agent** pursues its goal so persistently that it takes attacker-like action with no attacker present. No longer theoretical

20 YEARS OF SECURING PAYMENT DATA PCI

# How Agents Expand the CDE Attack Surface

Part one: data flows, execution paths, and goal-directed behaviour

- **Expanded Data Flows** CHD moves through context windows, vector stores, and tool responses, flows that never appear on a DFD
- **Non-Deterministic Paths** agents reach systems dynamically at runtime, so scope cannot be fully mapped at design time
- **Tool and API Interfaces** every tool call is a potential exfiltration point or injection vector
- **Goal-Directed Overreach** UK AISI, August 2026: agents under test created fake identities and social-engineered a real maintainer to approve malicious code. Never instructed to deceive

20 YEARS OF SECURING PAYMENT DATA PCI

# Scope: What Can the Tools Reach?

Connectivity defines scope, and capability defines connectivity

- **If a Tool Can Reach the CDE, It Is Connected** access is what counts, not use. An unused permission is still a connection, and scope follows capability rather than observed behaviour
- **Then Follow the Chain Upwards** if the tool is connected, is the agent? Is the orchestrator? Is the model? Each link needs an answer, not an assumption
- **This Was Never Only About Data Flow Diagrams** Requirement 1 is segmentation and isolation, and segmentation is proved by penetration testing, not by a diagram
- **A Prompt Is Not a Control Boundary** AISI's own finding: scope was stated in the task prompt but not enforced in the network. Containment must not depend on the agent choosing to respect it

20 YEARS OF SECURING PAYMENT DATA PCI

# Does Full PAN Ever Reach the Model?

Ask this first, because the answer decides everything downstream

- **Ask the Question Directly** does full PAN enter the prompt, the context window, or a tool response? If yes, every component downstream of that point is in scope
- **Then Ask Where Inference Runs** cloud or local? This matters because no frontier model provider is a validated service provider on the card scheme registries
- **The Recommendation Is Never** full PAN should not reach the model. Design it out rather than compensating around it
- **Use Irreversible Persistent Tokens** format-preserving tokens, or a persistent token plus BIN range, give orchestration and fraud logic what they need without PAN
- **Not One-Time-Use References** single-use references break the cross-transaction correlation fraud models depend on, so teams detokenise to work around them and reintroduce the exposure

20 YEARS OF SECURING PAYMENT DATA PCI

# Mapping Agentic Risk to the OWASP ASI Top 10

Selected categories from the Agentic Top 10, v2.01, June 2026

| OWASP ASI Category | What It Looks Like in a Payment Pipeline |
|---|---|
| ASI01 Agent Goal Hijack | An attacker steers the agent so it reroutes a payment or approves a step-up it should have declined |
| ASI02 Tool Misuse and Exploitation | Crafted input triggers unsafe tool calls, exporting CHD through a legitimate integration |
| ASI03 Agent Identity and Privilege Abuse | The agent holds broader permissions than any transaction needs, widening the blast radius |
| ASI06 Memory and Context Poisoning | Stored context or long-term memory is corrupted, exposing or misusing CHD held there |
| ASI10 Rogue Agents | The agent operates outside policy through drift or design failure, with no attacker involved |

20 YEARS OF SECURING PAYMENT DATA PCI

# Scoping and Compliance

## Where the standard runs out of road



# Where PCI DSS Meets Its Limits

The gap between what the standard assumes and what agents do

| PCI DSS Assumes | Agentic AI Introduces |
|---|---|
| Static, enumerable system components | Dynamic components with runtime scope |
| Defined network segments and data flows | Implicit connectivity via tool calls |
| Human-initiated access to CHD | Autonomous CHD access, no human trigger |
| Predictable connectivity between systems | Transitive reach unknown at design time |
| Explicit storage and transmission events | CHD in context windows, neither stored nor in transit |

20 YEARS OF SECURING PAYMENT DATA PCI

# A Three-Part Scoping Framework for Agentic AI

A structured starting point for your CDE boundary assessment

- **1 Data Flow Scoping** where does the agent touch CHD? Map every point it accesses, processes, or transmits cardholder data, including tool calls, memory, context windows, and external model providers
- **2 Connected System Scoping** how far does the agent reach? Identify every system it can touch, directly or transitively, including connectivity that segmentation does not capture
- **3 Control Applicability** do your controls still work? Test logging, access control, data minimisation, and human oversight against the agent's full operational scope

20 YEARS OF SECURING PAYMENT DATA PCI

# Applying the Framework: A Worked Example

A real-time fraud-decisioning agent, run through all three parts

- **Data Flow** the CDE footprint covers CHD read from the transaction record, behavioural signals from a vector store, and any CHD echoed into the reasoning trace or logs
- **Connected Systems** tool calls reach the case-management system, the card scheme API, and the model provider's inference endpoint, each now in scope
- **Controls** logging must capture what the agent did and why, access control must bound which tools it can call, and human oversight is needed where manual review used to sit

20 YEARS OF SECURING PAYMENT DATA PCI

# Controls That Actually Move the Needle

Four controls that address agentic risk specifically

- **Graduated Autonomy** progress agents from assisted to supervised to autonomous, with confidence thresholds and approval checkpoints at each step. This is your strongest compensating control argument
- **Agent Observability (Req 10)** trace activity, monitor decisions, and baseline behaviour. If scope drifts at runtime, monitoring is what replaces the point-in-time snapshot
- **Continuous Adversarial Testing (Req 11)** red team the agent, simulate goal hijack and tool misuse, and validate guardrails on a schedule rather than once at go-live
- **Model Supply Chain Integrity (Req 6.3.2, 12.8)** track model provenance, maintain an AI bill of materials, and scan dependencies. This is OWASP ASI04 and it applies to purchased agents too

20 YEARS OF SECURING PAYMENT DATA PCI

# Security First, Then Compliance

The objective is not losing cardholder data, not passing an assessment

- **Start With the Outcome** the goal is that CHD does not leave your control. Compliance is evidence that you tried, not proof that you succeeded
- **ISO/IEC 42001** gives the documentation spine for agent oversight and lifecycle control
- **NIST AI RMF** Govern, Map, Measure, Manage, useful vocabulary for the risk conversation
- **EU AI Act, Article 15** cybersecurity, accuracy and robustness duties for high-risk systems
- **OWASP ASI Top 10** the practitioner taxonomy, and the one your engineers will actually use
- **PCI SSC** no agentic-specific scoping guidance published as yet.

20 YEARS OF SECURING PAYMENT DATA PCI

# Key Takeaways

Four things to take back to your organisation

- **01 The Surface Is Already Bigger** cross-system agent behaviour creates flows and connectivity that DFDs and network maps will not capture
- **02 Keep PAN Away From the Model** use irreversible persistent tokens. This single decision removes most of the scope problem
- **03 Enforce Scope, Do Not State It** capability defines connectivity. Prove segmentation by penetration test, not by diagram
- **04 Then Ask the Harder Question** whether the standard itself can still carry this weight

20 YEARS OF SECURING PAYMENT DATA PCI

# A Harder Question: Is PCI DSS Still Fit for Purpose?

A position worth arguing, and worth arguing against

- **The Drift Started Before AI** PCI DSS was built for servers you could point at. Cloud, containers and ephemeral functions already strained it. Agentic AI did not create the mismatch, it exposed it
- **Metric Architecture, Imperial Standard** we keep converting between two systems that were never designed to interoperate, and the conversion loss shows up as compensating controls and interpretation
- **The Provocation** if a standard needs this much reinterpretation to apply, the honest response may be to start afresh rather than patch further. Whilst PCI DSS might not fit, there are 14 others that would
- **The Counter-Argument** the standard is deliberately technology-neutral, it has absorbed virtualisation and cloud before, and twenty years of accumulated assessor practice has real value

20 YEARS OF SECURING PAYMENT DATA PCI

# A few things to consider

I want to leave you with these thoughts

- How are you handling AI agent components in scoping conversations with your QSA today?
- Has your organisation classified any AI agent as a system component under PCI DSS?
- What controls are you applying to agent memory and context window content?

20 YEARS OF SECURING PAYMENT DATA
PCI




# Europe
## Community Meeting

## EDINBURGH | 2026