Prompt Injection Defense Checklist

Practical mitigations for direct and indirect prompt injection

Prepared by	Nazline Mwita, CompTIA Security+
Practice	HarLyn Digital Partners
Scope	Self-assessment starting point — not a substitute for a scoped audit

Direct Injection (user input)

- User messages are never concatenated directly into a system prompt with implicit trust.
- There's a tested boundary for “ignore previous instructions”-style attempts.
- Sensitive actions (payments, data deletion, permission changes) require a non-bypassable confirmation step outside model control.

Indirect Injection (retrieved content)

- Content pulled from RAG retrieval, web pages, or tool outputs is treated as untrusted, same as user input.
- The model's tool-calling permissions are scoped — retrieved text cannot itself grant new tool access.
- Retrieved documents are source-tagged so the model (and a reviewer) can tell instruction from data.

Output Validation

- Tool-call arguments are validated against a strict schema before execution, not trusted as-is.
- Outputs that trigger external side effects (emails, API calls, file writes) pass through a validation layer.
- There's logging of what the model actually did, not just what it was asked to do.

Operational

- There's a documented incident path for when an injection attempt is detected or suspected.
- Agent permission boundaries are reviewed whenever a new tool or data source is added.

How to use this checklist

This is a starting-point self-assessment, not a substitute for a scoped engagement. Items checked “no” or “unsure” are candidates for a Secure Digital Workflow Assessment.

Nazline Mwita, CompTIA Security+ certified Cybersecurity Assurance Lead and Co-Founder at HarLyn Digital Partners. Specializing in authorized web & API security assessments, KDPA compliance reviews, and defensive cloud architecture in Nairobi, Kenya.