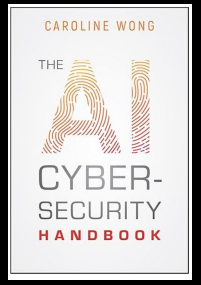
OWASP Top 10: 2025
Two Shifts That Matter More Than You Think

Caroline Wong, December 2025

Helpful Resources

Book: *The AI Cybersecurity Handbook*



https://www.amazon.com/AI-Cybersecurity-Handbook-Caroline-Wong/dp/1394340869/

Podcast: The AI Security Edge

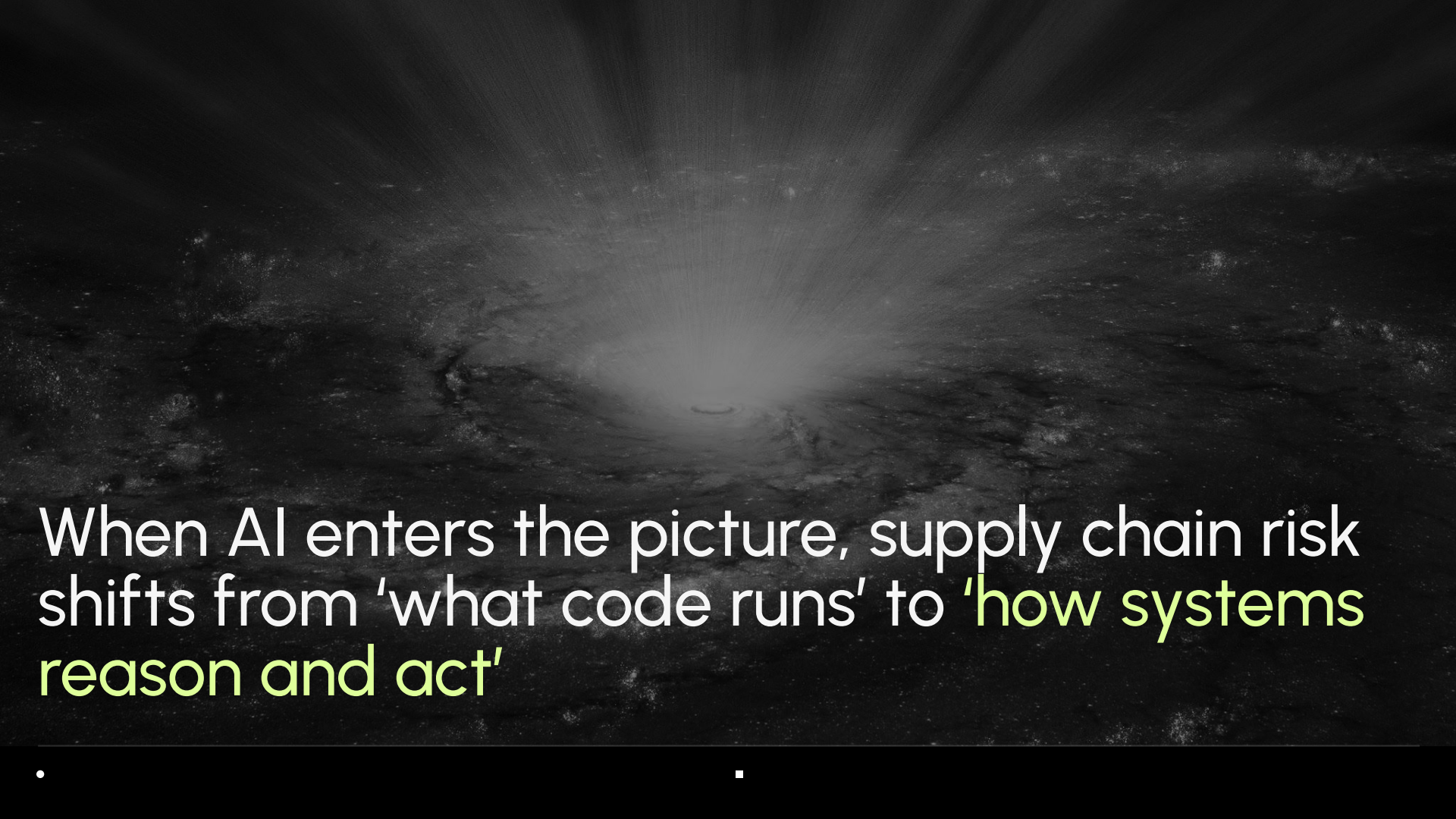
https://techstrong.tv/videos/the-ai-security-edge

Company: depthfirst

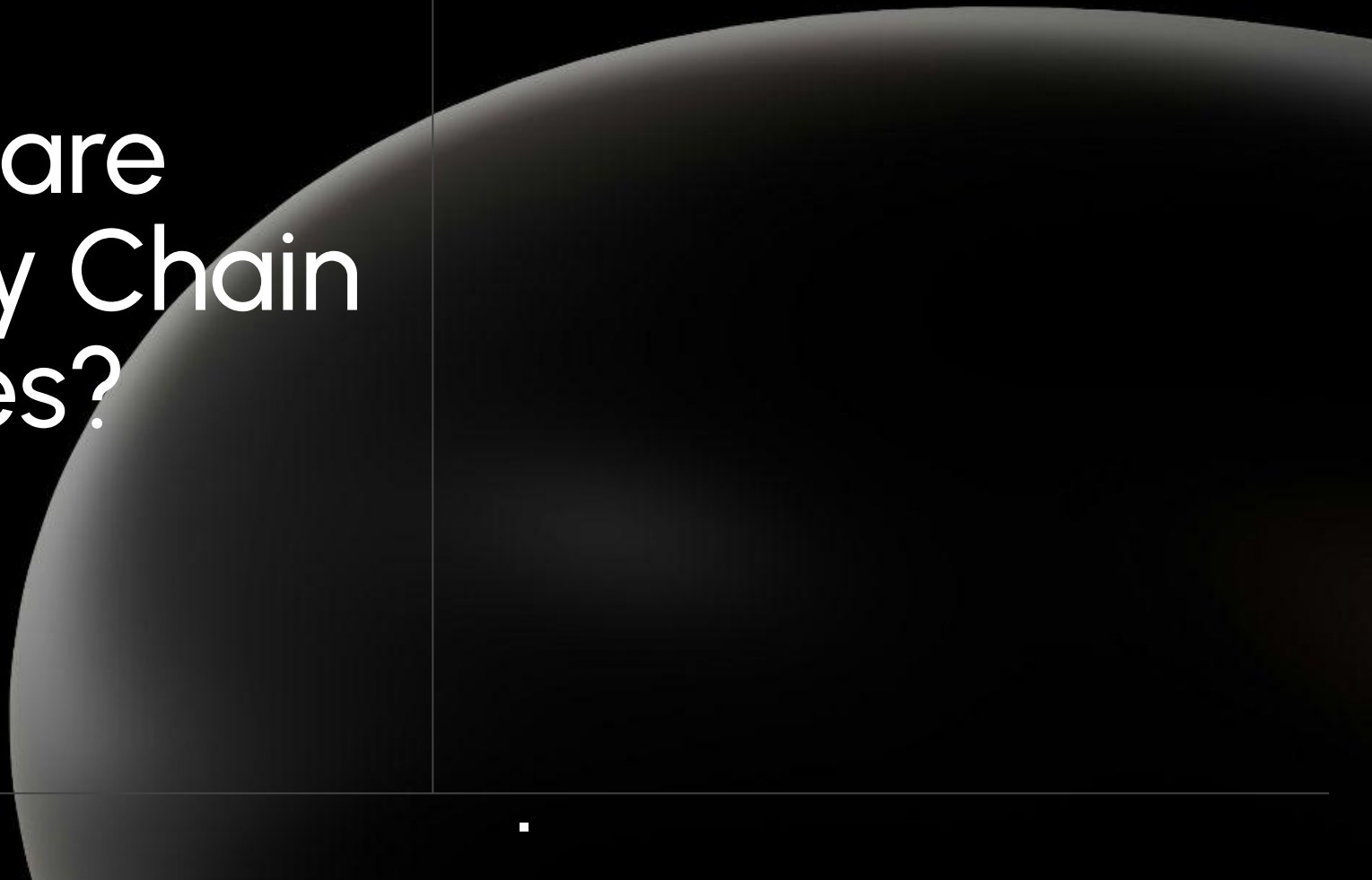
https://depthfirst.com/about
Email: cyrus@depthfirst.com

Newsletter: Unsupervised Learning

https://newsletter.danielmiessler.com/


When AI enters the picture, supply chain risk shifts from ‘what code runs’ to ‘how systems reason and act’

Why Software Supply Chain Failures?

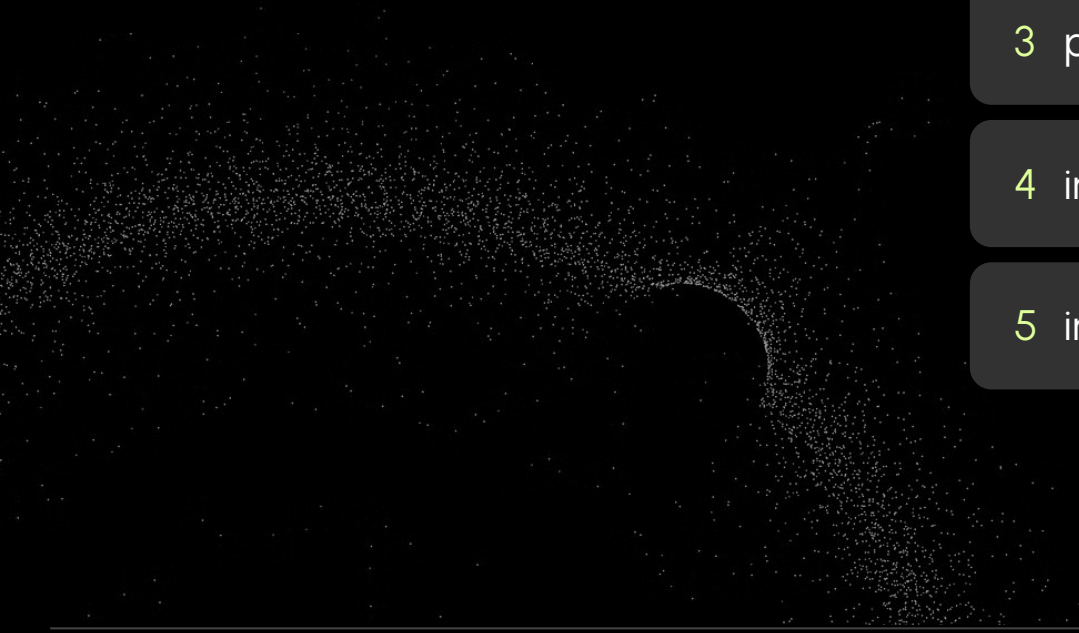


From Static Artifacts to Probabilistic Systems

Traditional software behaves in ways we're very familiar with. It's largely deterministic.

AI-enabled software behaves very differently. It's probabilistic by design.

The AI Supply Chain Is Bigger Than You Think



1 models

2 data

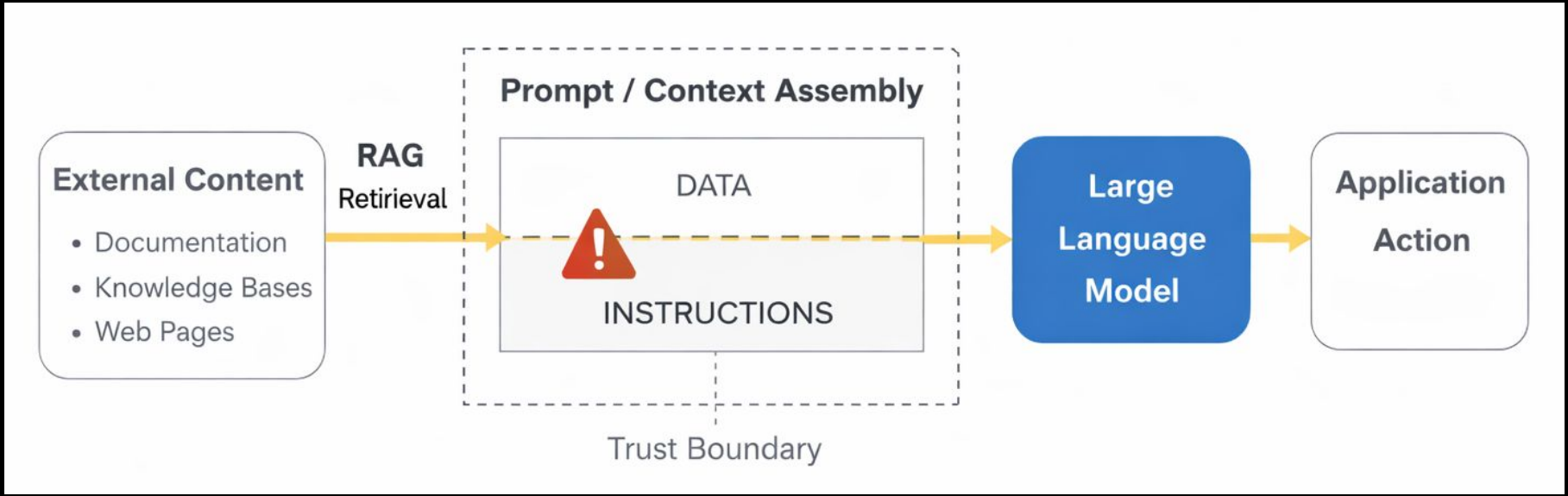
3 prompts

4 infrastructure

5 integrations

How AI Supply Chains Actually Fail

1. **Dataset poisoning**
2. Model artifact tampering
3. Malicious fine-tuning
4. Prompt supply chain attacks
5. Architectural subversion

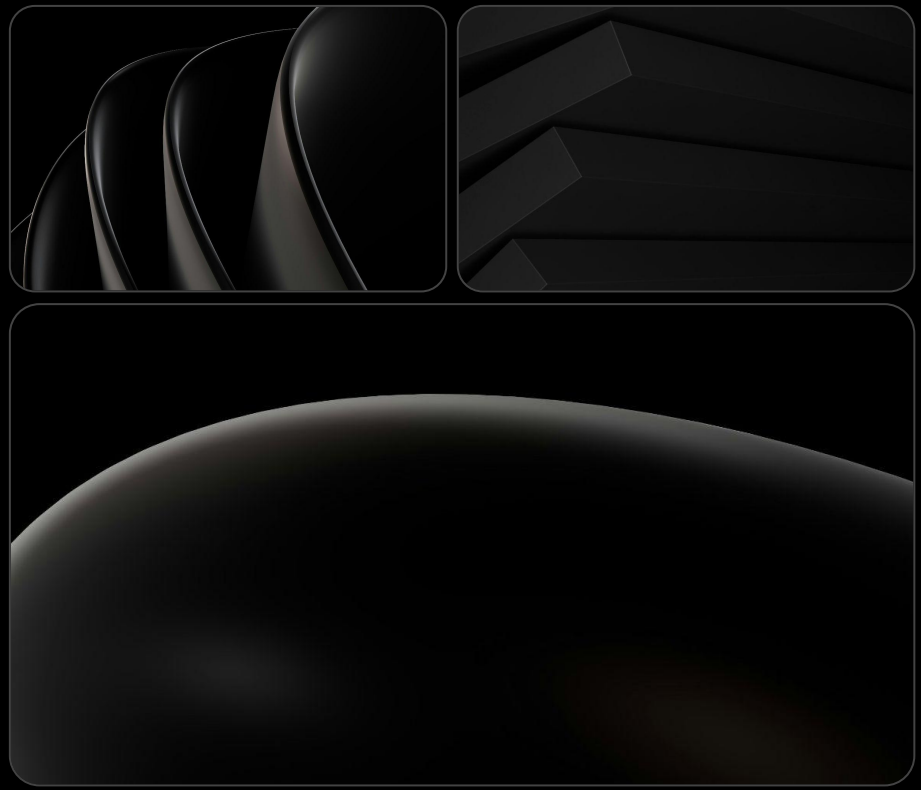

Prompt / Context Assembly
External Content
• Documentation
• Knowledge Bases
• Web Pages
RAG
Retirieval
DATA
INSTRUCTIONS
Large Language Model
Application Action
Trust Boundary

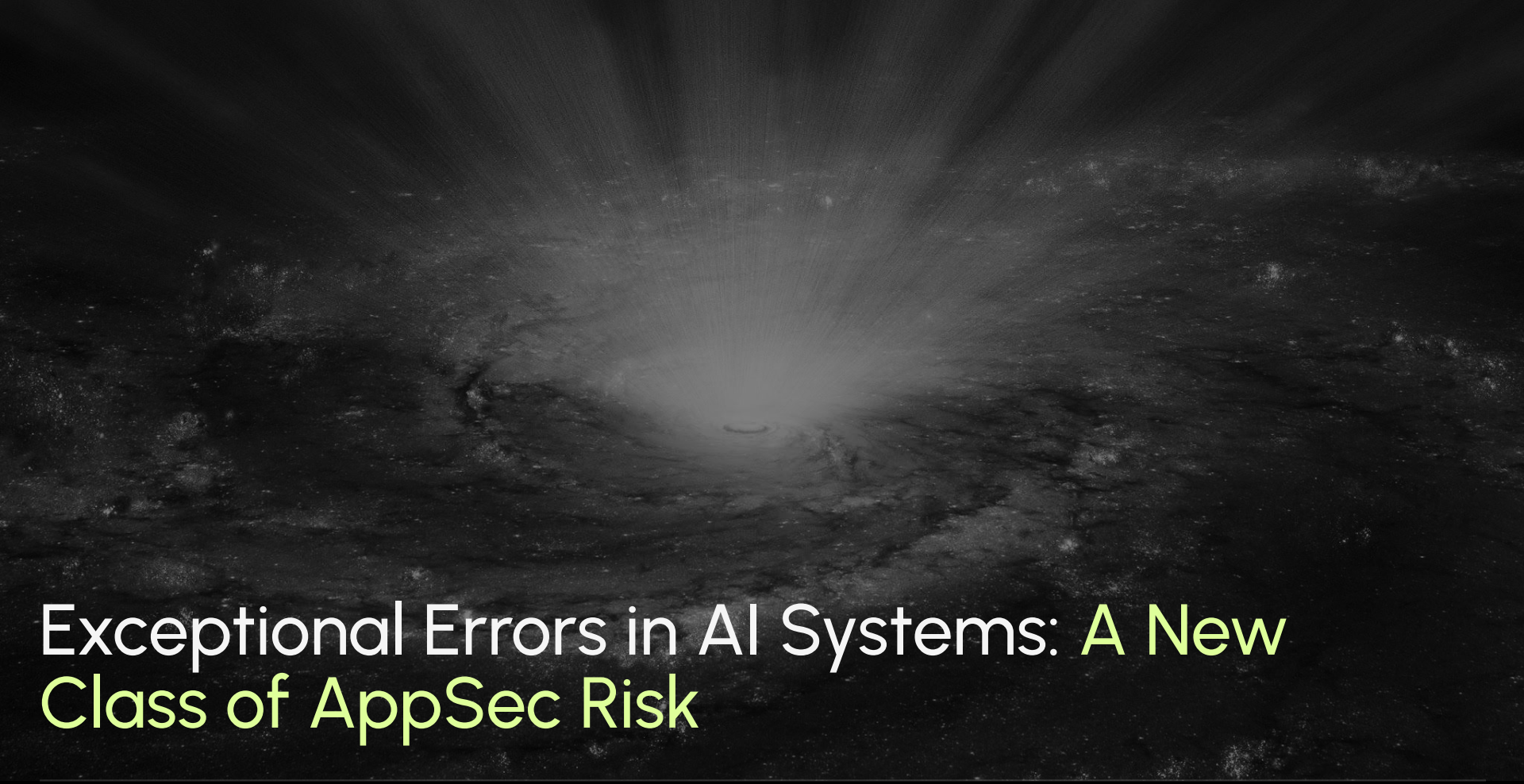
Why Detection Is Harder Than Traditional Supply Chain Attacks

Integrity failures are harder to see then availability failures.

How We Must Adapt

- Explicit trust boundaries
- Provenance tracking
- Least privilege for AI
- Change management
- Behavioral monitoring





Exceptional Errors in AI Systems: A New Class of AppSec Risk

“Exceptional Conditions”

OWASP isn’t talking about crashes or obvious failures.

It’s talking about situations where the system encounters something unexpected — and continues anyway.

How AI Changes Failure Modes

- Partial context
- Conflicting instructions
- Low confidence outputs
- Tool failure
- Permission ambiguity

Attackers Don't Break the Happy Path

1

Prompt injection that only succeeds after context truncation

2

Agents taking broader actions after tool failure

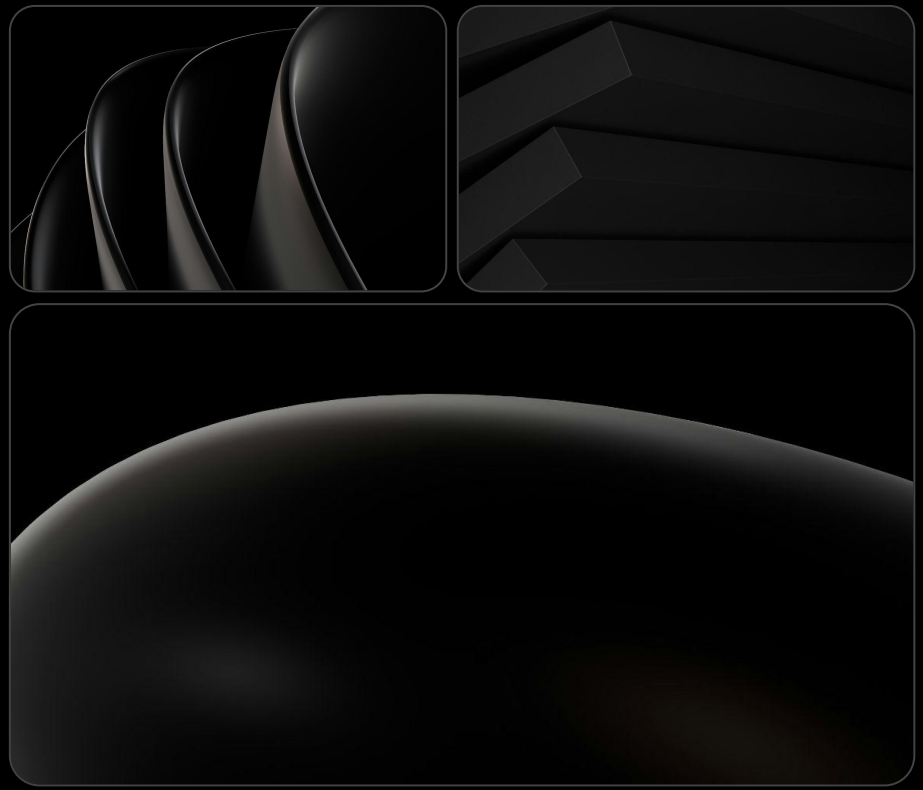
3

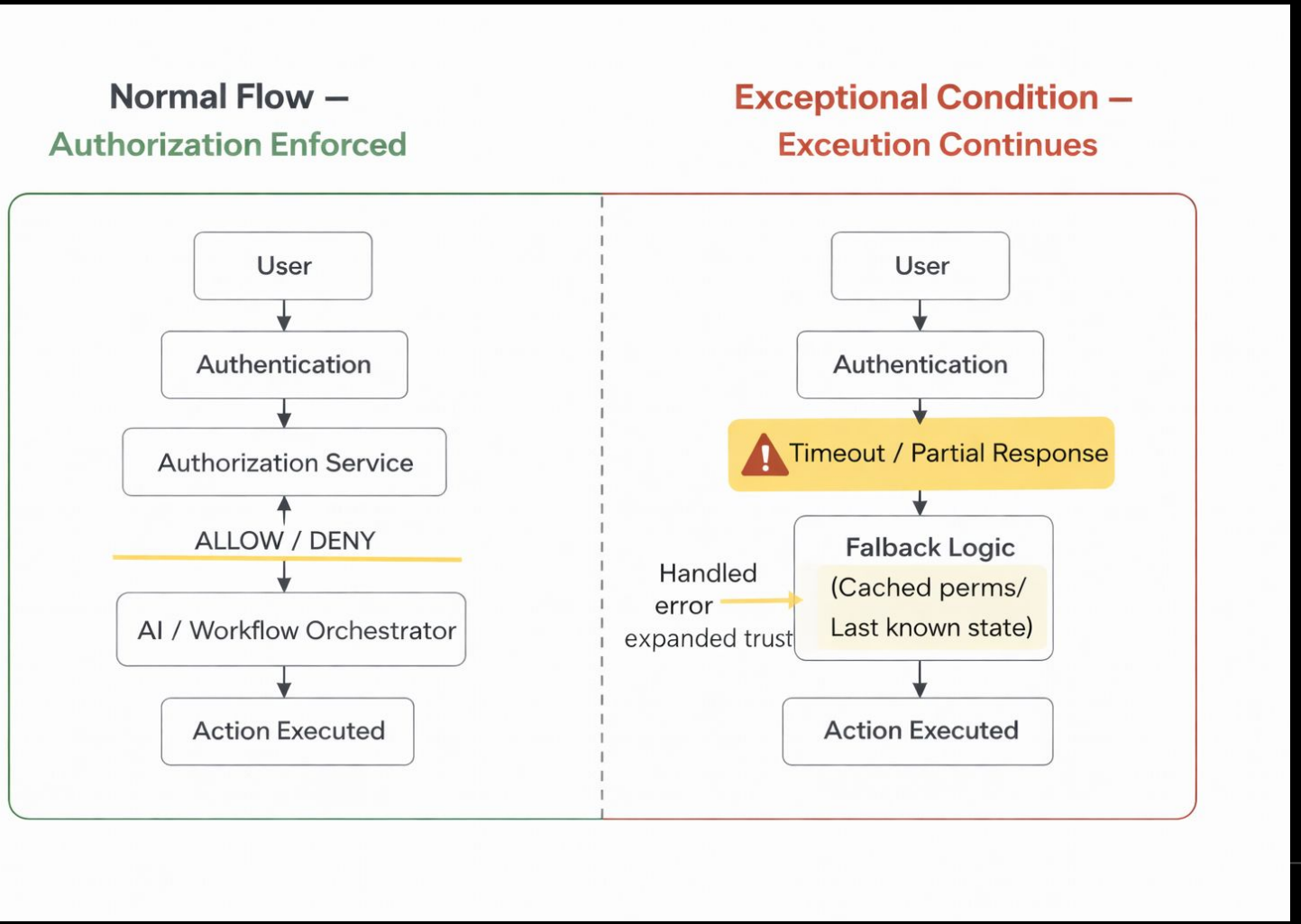
Authentication or authorization bypass via recovery logic

Why Nothing Looks "Broken"

In traditional exploitation, anomalies are often obvious. A malicious package opens a suspicious network connection. A compromised system behaves in a way that clearly violates baseline expectations.

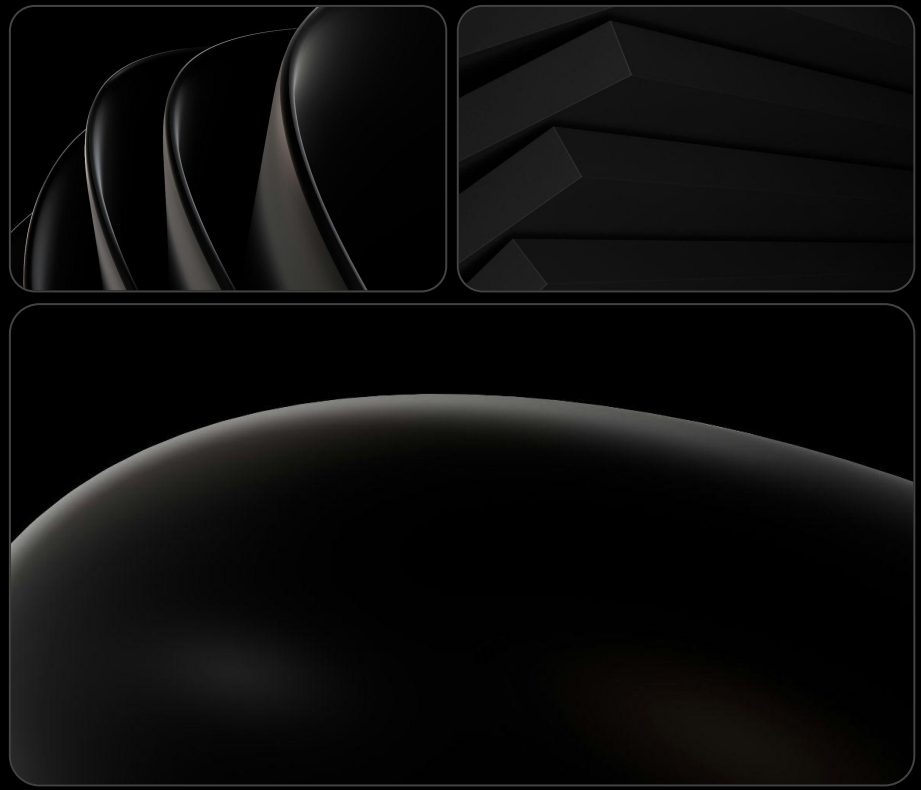
AI exploitation looks very different. It manifests as *plausible behavior*. A slightly different recommendation. A subtly broader action. A decision that still makes sense — just not the one you intended.

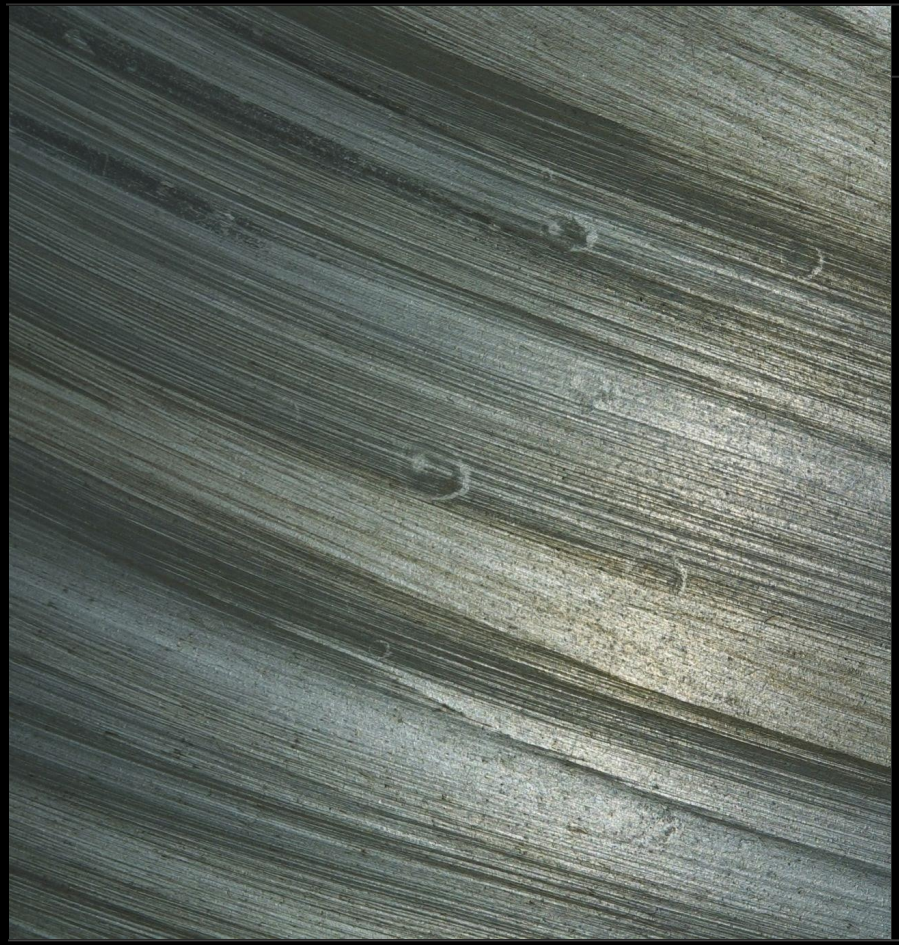



Normal Flow –
Authorization Enforced
User
Authentication
Authorization Service
ALLOW / DENY
AI / Workflow Orchestrator
Action Executed
Exceptional Condition –
Exceution Continues
User
Authentication
Timeout / Partial Response
Handled
error
expanded trust
Falback Logic
(Cached perms/
Last known state)
Action Executed

Designing for Unsafe States

- Fail closed on authority
- Explicit uncertainty handling
- Guardrails on fallback logic
- Consistent authorization checks
- Behavioral monitoring



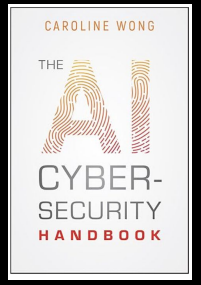


What AppSec Teams Must Change

- Threat model exceptional states
- Review fallback logic explicitly
- Include AI failure modes in design reviews
- Treat uncertainty as a risk factor
- Push error handling ownership upstream

Helpful Resources

Book: *The AI Cybersecurity Handbook*



https://www.amazon.com/AI-Cybersecurity-Handbook-Caroline-Wong/dp/1394340869/

Podcast: The AI Security Edge

https://techstrong.tv/videos/the-ai-security-edge

Company: depthfirst

https://depthfirst.com/about
Email: cyrus@depthfirst.com

Newsletter: Unsupervised Learning

https://newsletter.danielmiessler.com/