Mc Stem Eduversity
India/USA

AGENTIC AI IN ENTERPRISE SOFTWARE ENGINEERING AND ZERO-TRUST SECURITY

SUDHAKARA REDDY PERAM

Agentic AI in Enterprise Software Engineering and Zero-Trust Security

AUTONOMOUS DELIVERY, INTELLIGENT QUALITY, AND ADAPTIVE DEFENSE FOR THE MODERN ENTERPRISE

SUDHAKARA REDDY PERAM

PRINCIPAL SOFTWARE ENGINEER & SENIOR MANAGER, ENGINEERING ILLUMIO INC, SUNNYVALE, CALIFORNIA, USA

Pub. Div. Mc Stem Eduversity, USA/India

First paperback edition printed [2025] in India

A catalogue record for this book is available from the MCS Library USA & India.
ISBN: 978-81-970269-5-9

Published by
Publication Div,
Mrs Jimmy K, CEO, Mc Stem Eduversity, USA/India
704, Mahesh Villa, Mumbai 400058

For more copies of this book, please email:
office@mcstemeduversity.us
Tel: +91 9011424678

Designed and Set by:
P V Advertiser Ltd
Mumbai/New Delhi

2025
FIRST
EDITION

ABOUT THE AUTHOR

SUDHAKARA REDDY PERAM

Sudhakara Reddy Peram is a Principal Software Engineer and Senior Manager of Engineering at Illumio Inc, where he has spent more than a decade building the zero-trust segmentation platform relied upon by some of the world's most demanding enterprise security organizations. Over twenty years he has led engineering across cybersecurity, financial systems, consumer platforms, clinical reporting, and patent intelligence — a range that gives him an unusually broad view of what enterprise software actually costs to build, secure, and keep running.

At Illumio he has architected and delivered the platform capabilities that define modern microsegmentation: network traffic visualization, risky-port exposure analytics, rule hit count intelligence, vulnerability scoring and maps, ransomware risk analytics, and the AI-assisted labeling systems that make segmentation tractable at enterprise scale. He conceived and led the CI/CD modernization program that cut pull-request build times from over three hours to under ninety minutes, recovering approximately 1,200 engineering hours per day without adding headcount.

Earlier in his career he led quality engineering for Intuit's QuickBooks and Payroll platforms, built automation for Google Play Books across web and mobile, sustained Pfizer's Clinical Data Analysis and Reporting System under regulatory scrutiny, and engineered the global patent ingestion pipelines behind Thomson Reuters' Delphion and Thomson Innovation.

He is a Senior Member of the IEEE and a member of Sigma Xi and the ACM, has filed multiple patents in applied artificial intelligence, serves on the editorial boards of several international journals, and has published extensively on machine learning for network traffic analysis, ransomware detection, and predictive quality engineering. He holds a Bachelor of Technology in Electronics and Communication Engineering from Jawaharlal Nehru Technological University.

DEDICATION

To every engineer who has watched a progress bar for three hours to learn whether a one-line change was sound — and to every security analyst who has been handed a rule set nobody dares delete from.

This book is for you. It is also an argument that neither of you should have had to.

FOREWORD

Software engineering has spent forty years learning to build faster and thirty years learning to defend what it builds. For most of that time the two disciplines have spoken past one another. Delivery teams treated security as a gate to be passed. Security teams treated delivery as a source of risk to be slowed. Both were, in their way, correct, and the result was an industry that shipped quickly and defended poorly, or defended carefully and shipped late.

What has changed is that both disciplines have run out of room. A pipeline running a hundred thousand tests cannot be reasoned about by reading logs. A network of twenty thousand workloads generating millions of flow records a day cannot be segmented by inspecting spreadsheets. In both domains the evidence has outgrown the human capacity to process it, and in both the answer has turned out to be the same: instrument everything, learn from the telemetry, and let machines make the routine judgments so that people can make the consequential ones.

This book is written from inside that transition. Its author has spent a decade building the systems it describes — not prototypes, but production platforms serving enterprises that would notice immediately if they were wrong. That provenance shows. The chapters that follow are unusually specific about what worked, what it cost, and what it measured, and they are candid about the failure modes of machine judgment in a domain where a wrong answer is a breach.

Read it as a practitioner's account rather than a survey. It will be most useful to the engineer who suspects that their pipeline is slower than it should be, and to the architect who suspects that a fifth of their firewall rules protect nothing at all. Both suspicions are usually correct, and this book explains how to prove it.

Dr P K Rajput, Sr. VP, Cadilla Pharma, India

PREFACE

0.1 Why This Book?

I have spent twenty years on both sides of a divide that should never have existed. I have built delivery pipelines and I have built the security platforms that defend what those pipelines ship, and I have come to believe that they are the same discipline practiced by people who do not talk to each other.

The evidence accumulated slowly. When I led the modernization of a pull-request pipeline that took over three hours, the fix was not a faster machine; it was telemetry — finding out which tests were flaky, which were slow, which never mattered. When I built the analytics that show an enterprise which of its firewall rules are actually used, the fix was the same: telemetry, applied to a system that had been managed on assumption for a decade. In both cases the organization had been making expensive decisions in the dark, and in both cases the remedy was to turn on the light.

0.2 The Gap This Book Fills

There are excellent books on continuous delivery and excellent books on zero-trust architecture. There are very few that treat them as a single engineering problem, and almost none written from inside the production systems where the claims can be checked.

This book is deliberately specific. Where a chapter reports that a capability reduced exposed vulnerable ports by twenty to thirty-five percent, or that containerized execution cut test runtime by seventy percent, those are measured outcomes from systems in enterprise production, not projections. Where a technique failed, or where machine judgment proved unreliable, I have said so, because the failure modes are more instructive than the successes.

0.3 How to Use This Book

Part I establishes the argument and can be read on its own. Parts II through V are practitioner's accounts — CI/CD modernization, security intelligence, AI-driven automation, and quality engineering — and each is largely self-contained; an engineer with a specific problem should

feel free to start there. Parts VI and VII address mission-critical platforms, performance, and reliability. Part VIII concerns leadership, research, and where the field is going.

Every chapter closes with a summary, and the summaries are written to be read consecutively as an abbreviated version of the whole.

0.4 A Note on Terminology

I use workload to mean any unit of compute governed by policy — a virtual machine, a container, a bare-metal host. I use label to mean a piece of structured metadata describing a workload's role, application, environment, or location, and policy to mean an intent expressed in terms of labels rather than addresses. Segmentation is the enforcement of such policy at the workload rather than at a network chokepoint.

On the delivery side, pipeline means the automated path from commit to production; flake means a test that fails non-deterministically on unchanged code; and shard means an independently executable subset of a test suite. Where an industry term is genuinely contested — agentic is the worst offender — I define it at first use and try not to lean on it.

Contents

Part I

FOUNDATIONS OF ENTERPRISE SOFTWARE ENGINEERING AND ZERO-TRUST SECURITY

Chapter 1

Evolution of Enterprise Software Delivery: From Release 1.0 to Autonomous 5.0

1.1 Introduction

Every enterprise that ships software carries an invisible tax. It is not paid in license fees or cloud spend but in waiting: engineers idle while a pipeline grinds, a release stalls behind a flaky test, a security policy is reviewed by hand because no system can say which rules are actually used. The tax compounds quietly, and because it is distributed across hundreds of people it rarely appears on any budget line [1], [2].

This book is about paying down that tax. It argues that two problems long treated as separate — how quickly an organization can deliver software, and how well it can defend what it has delivered — are in fact the same problem seen from different angles. Both are constrained by the same thing: an absence of intelligence about what the system is actually doing.

The chapters that follow trace a single thesis. Delivery pipelines and security policies were built on static assumptions in a world that is no longer static. Both are now being rebuilt on telemetry, machine learning, and increasingly on autonomous agents that observe, decide, and act. The transformation is not speculative. It has already happened in production systems serving thousands of enterprise customers, and the evidence for it is measurable in build minutes saved, attack surface removed, and incidents that never occurred.

We begin where every such argument must begin: with how the discipline arrived here.

1.2 Delivery 1.0: The Waterfall and Manual Release Era

For most of the industry's first three decades, software was delivered the way buildings were constructed. Requirements were gathered, specifications frozen, code written against them, and testing performed at the end as a discrete phase. Release was an event — scheduled months in advance, rehearsed, and attended [1].

The model was not irrational. When software shipped on physical media to customers who could not easily patch it, the cost of a defect escaping was enormous, and heavy up-front

The AI Agents Are Coming. Are You Ready to Secure Them?

Agentic AI promises to revolutionize enterprise software engineering—but only if we can guarantee its safety. This book is the essential playbook for engineering teams and security leaders navigating the collision of autonomous intelligence and Zero-Trust principles.

Learn how to build a 'Digital Immune System' where your AI agents are monitored, verified, and secured at every step, turning security into a strategic enabler rather than a bottleneck.

Secure the Autonomous Enterprise:

Build smarter.

Deploy safer.

Verify always.

INR 599 ($ 7)

9 788197 026959

ISBN: 978-81-970269-5-9

Publisher:
Mc Stem Eduversity, India & USA

www.mcstemeduversity.ac.in