Open-Source Compliance & Security – Essentials for Non-IT Professionals
Total Duration: 24 Hours
Delivery: Online (Hands-on + Demos)

Module 1: Getting Started with Applications, Docker & Git Basics (6 Hours)

Goal: Build confidence with essential tools and concepts through guided practice.
Concepts:

- What is an application? Demo with a pre-built Python (Flask) app.
- What is Open Source? Why companies use it.
- Risks of open-source: Legal, security, operational.
- Simple intro to Software Bill of Materials (SBOM) as an "ingredient list" for software.
- Docker basics: what it is, pulling & running containers.
- Git basics: cloning a repository, checking files, simple commands.

Hands-on:

1. Run a pre-built app using Docker.
2. Pull and start a ready-made container image.
3. Clone a GitHub repository and explore files.
4. Use a simple scanning tool to identify open-source components.
5. View SBOM results and spot license types.

Module 2: Understanding Licenses Made Easy (6 Hours)

Goal: Learn how to recognize and interpret common licenses without legal jargon.
Concepts:

- Common license categories (Permissive, Copyleft, Proprietary) with real-world analogies.
- How licenses appear in software projects.
- How "hidden" dependencies affect compliance.

Hands-on:

1. Open a sample project and locate license files.
2. Identify license types with a simple scanning tool.
3. Practice matching licenses to usage rules.

Module 3: Security Awareness in Open Source (6 Hours)

Goal: Understand vulnerabilities and learn to check software safety.
Concepts:

- What is a software vulnerability? Analogy: product safety recalls.
- Where to find vulnerability alerts.
- How basic security scanning works.

Hands-on:

1. Scan a pre-built app for vulnerabilities using a simple tool.
2. Interpret scan results and identify potential risks.
3. Compare secure vs. insecure versions of software.

Module 4: End-to-End Practice Project (6 Hours)

Goal: Apply the full process from identification to reporting.
Concepts:

- Simple compliance workflow: Detect → Review → Approve.
- How to communicate findings to a team.

Hands-on:

1. Start with a pre-built Python project.
2. Generate SBOM.
3. Identify licenses.
4. Scan for vulnerabilities.
5. Prepare a basic compliance & security report.

Tools Used (Simplified): Only one easy-to-use compliance scanner, one SBOM generator, one security scanner, Docker, and Git (basic commands only).
