

Dependency Compromise Exposure Worksheet

Use this worksheet to prepare a bounded exercise around a compromised third-party library. Pick one application or build process, then trace where dependency code runs and what it can reach.

Fill it in with application, platform and security owners. Use general descriptions, and leave out credentials, private repository links and sensitive architecture detail.

01 Map one dependency path

FIELD	YOUR NOTES
Application or build process	
Application owner and platform owner	
Where the code runs: install, build, test or runtime	
Identity it runs as, and its permissions	
Environments, services or data it can reach	
Telemetry available, and who receives it	

Name the scenario. A malicious package, a compromised maintainer account and a compromised registry are different exercises.

02 Choose the starting position

Emulated entry path. A controlled change stands in for the malicious dependency at an agreed stage. It shows which controls that route meets.

Assumed foothold. The exercise starts after the code has run. It shows attacker reach and defensive response, but cannot show whether prevention controls would have stopped the entry.

STARTING-POSITION FIELD

YOUR NOTES

Scenario and starting position

Steps exercised, and steps assumed

03 Write the questions first

Pick two or three questions the starting position can answer:

- Which permissions does the execution identity expose?
- Which boundary should stop it reaching another environment?
- Would the activity reach someone able to investigate and contain it?
- Could the team identify affected builds within scope?

Take the completed worksheet to the relevant owners and resolve unknown permissions. Then agree the smallest scenario that answers the question.

SilentGrid runs this as a supply chain attack assumed breach, and a supply chain scenario can also run as a cycle of CAOS (Continuous Adversary Operations Service). The scenario and scope are agreed at scoping.

Discuss your testing priorities

www.silentgrid.com/contact