IR Protect

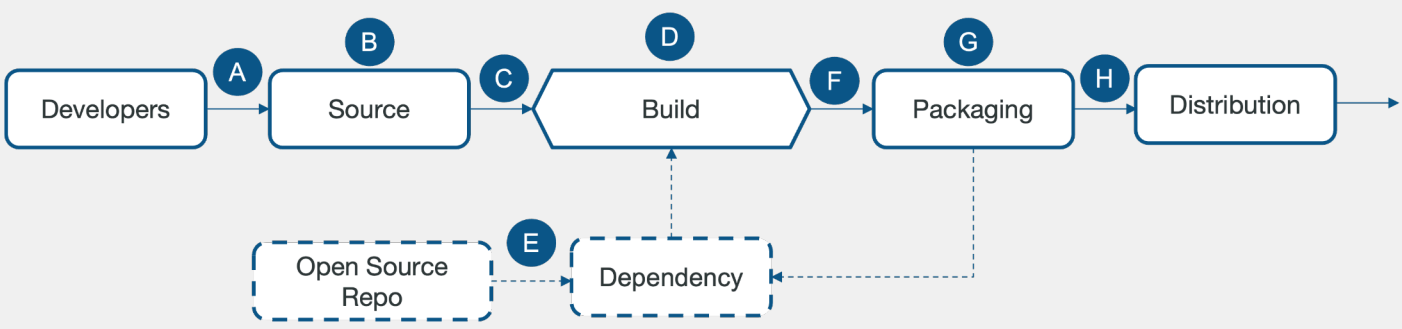
Monitor the build process, providing on-the-fly validation of components as they are pulled in. Policies are applied to issue warnings or stop the build for the most serious policy violations. The application of policy during the build ensures compliance with regulations related to data and customers safety.

- ✔ **Get Comprehensive:** Recognize all build components
- ✔ **Stay Engaged:** Track build systems and updates
- ✔ **Automate Enforcement:** Enforce IP traffic rules
- ✔ **Automate Action:** Stop builds for major policy breaches
- ✔ **Be Everywhere:** Use in cloud or on-site
- ✔ **Trust, but Verify:** Prevent or alert on risky downloads

IR Protect Features

- ✔ All the benefits of Audit
- ✔ Identifies build system components
- ✔ Monitors builds enforcing policy
- ✔ Halts build distribution for severe policy violations
- ✔ Enforces access/protects build system components from tampering or modification
- ✔ Deployable in cloud or premise build environment
- ✔ Provides clear and concise audit capability for all components on every build.

Development Process Under Attack



	Threat	Known example
A	Submit bad code to the source repository	**Color**: The open-source maintainer of the popular npm package colors, intentionally introduced an offending commit that adds an infinite loop to the source code
B	Compromise source control platform	**PHP**: Attacker compromised PHP's self-hosted git server and injected two malicious commits.
C	Build with official process but from code not matching source control	**Webmin**: Attacker modified the build infrastructure to use source files not matching source control.
D	Compromise build platform	**SolarWinds**: Attacker compromised the build platform and installed an implant that injected malicious behavior during each build.

	Threat	Known example
E	Use risky dependency (i.e. A-H, recursively)	**event-stream**: Attacker added an innocuous dependency and then later updated the dependency to add malicious behavior. The update did not match the code submitted to GitHub
F	Upload an artifact that was not built by the CI/CD system	**CodeCov**: Attacker updated the CodeCov Bash Uploader script to export secrets, credentials, and other sensitive data stored in CI systems to an attacker-controlled server.
G	Compromise package repository	**Attacks on Package Mirrors**: Researcher ran mirrors for several popular package repositories, which could have been used to serve malicious packages.
H	Trick consumer into using bad package	**Apple, Microsoft, Tesla, and Dozens of Other Companies**: The attacker uploaded a public package with the same name used by the companies tricking them into using new packages.

Solutions

	IR Audit	IR Protect	IR Attest
Advanced SBOM Build, Scan, Utilities	✔	✔	✔
Vulnerability Tracking & Risk Scores	✔	✔	✔
Unearth Hidden Vulnerabilities	✔	✔	✔
SBOM & Artifacts Exchange	✔	✔	✔
Multiple Dashboards	✔	✔	✔
TruSBOM™ Build, Scan, Utilities		✔	✔
Secrets Exfiltration Prevention		✔	✔
Pipeline Security Engine		✔	✔
Pipeline Security Enforcer		✔	✔
Protocol Aware Inspection		✔	✔
License Enforcement		✔	✔
SLA Audit Logs & Time Snapshots			✔
Automated Attestation Workflow			✔

Reach out to info@invisirisk.com for more info

Copyright © 2024. InvisiRisk, Inc.