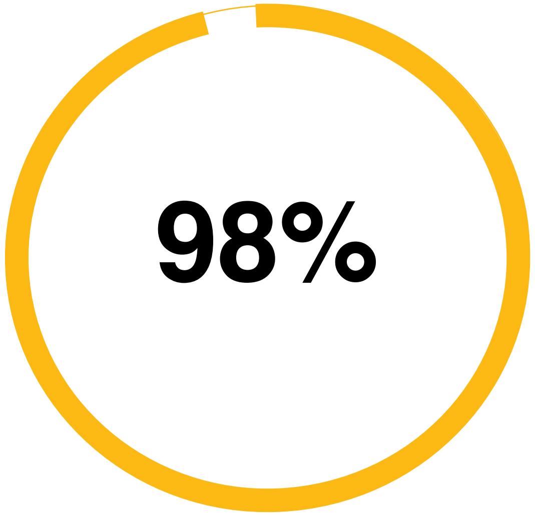
SLSA SPECIFICATION

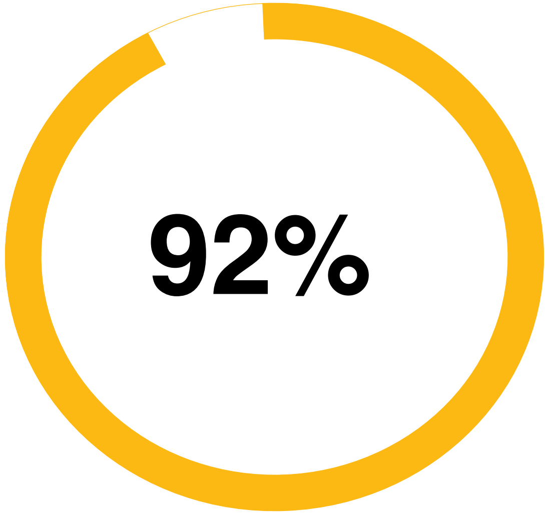
TECHNICAL PATH TO OPEN-SOURCE SOFTWARE SUPPLY CHAIN SECURITY

Harimohan Rajamohanan | Lead OSS Security COE | Wipro Limited

Open Source Software Security Landscape

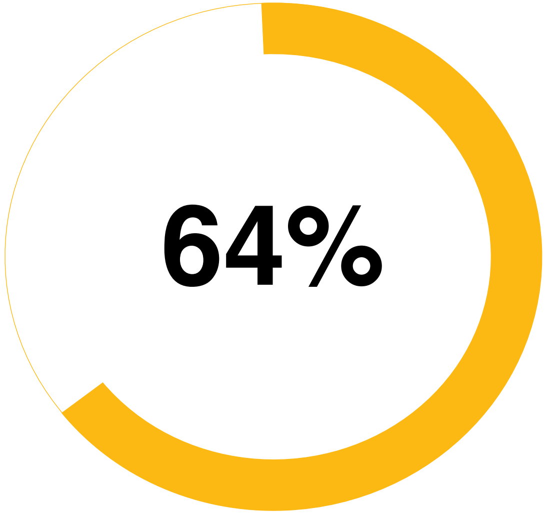
# of codebases use open source software





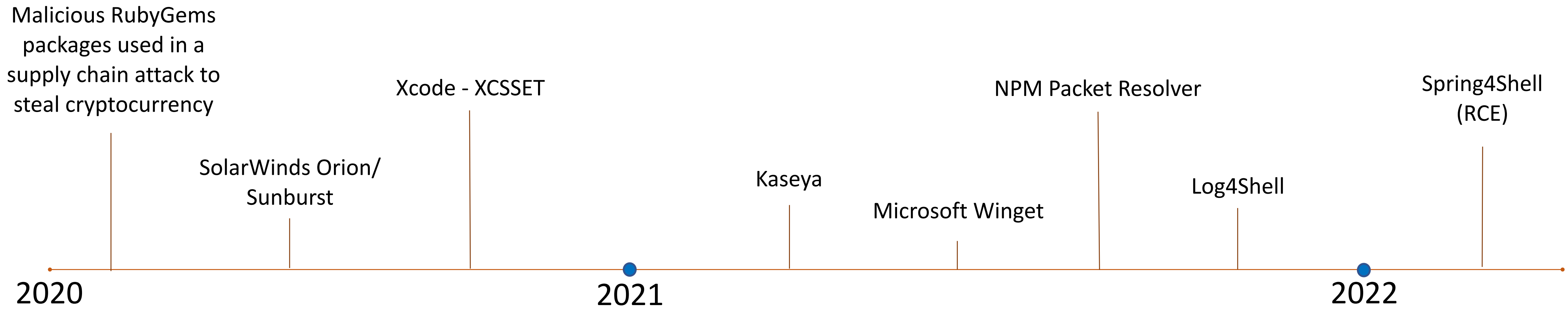
# of projects containing at least one outdated and vulnerable dependency

# of organizations affected by software supply chain attacks



[Source: Sonatype, Anchore 2021 Report on Supply Chain Security]

Notable Attacks



Malicious RubyGems packages used in a supply chain attack to steal cryptocurrency

SolarWinds Orion/ Sunburst

Xcode - XCSSET

Kaseya

Microsoft Winget

NPM Packet Resolver

Log4Shell

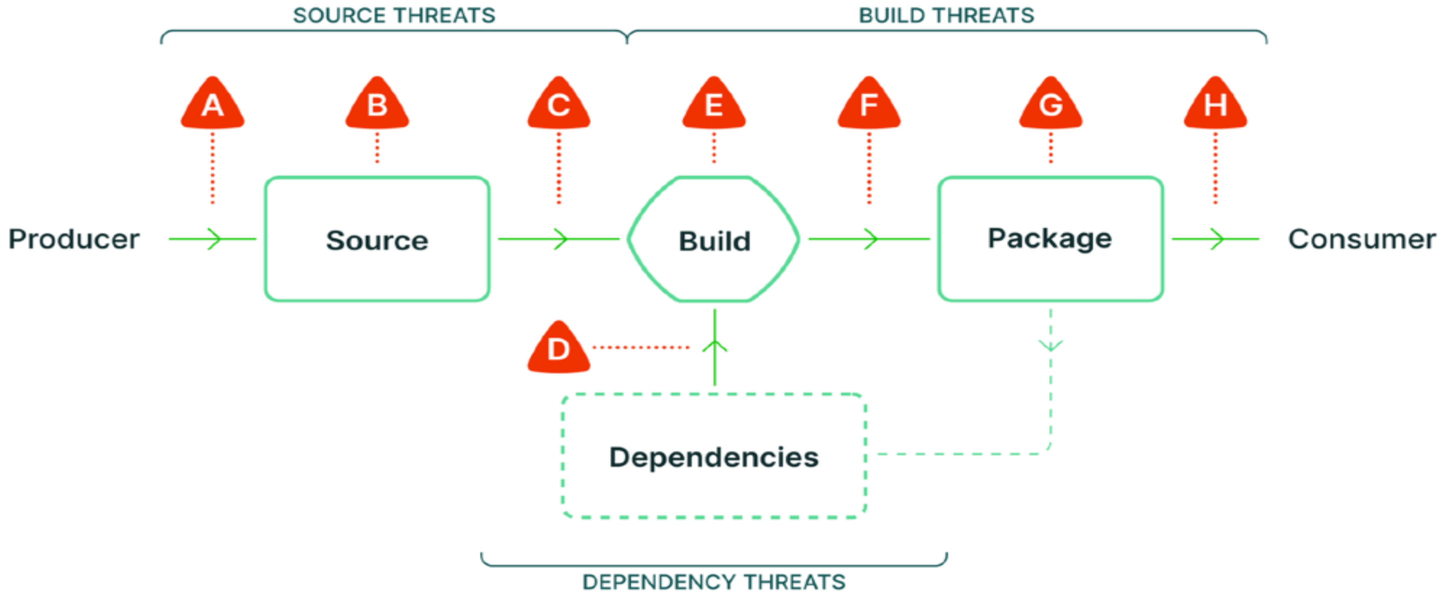
Spring4Shell (RCE)

Top 4 open source ecosystems represent 37 million+ component versions

2.2 trillion 3rd party OS packages borrowed by Developers

Source: Sonatype, 2021 SSCS Report, *Anchore 2021 Report on Supply Chain Security, Atlantic Council Breaking Trust Dataset*

The Supply Chain Threat



SOURCE THREATS

A Submit unauthorized change

B Compromise source repo

C Build from modified source

DEPENDENCY THREATS

D Use compromised dependency

BUILD THREATS

E Compromise build process

F Upload modified package

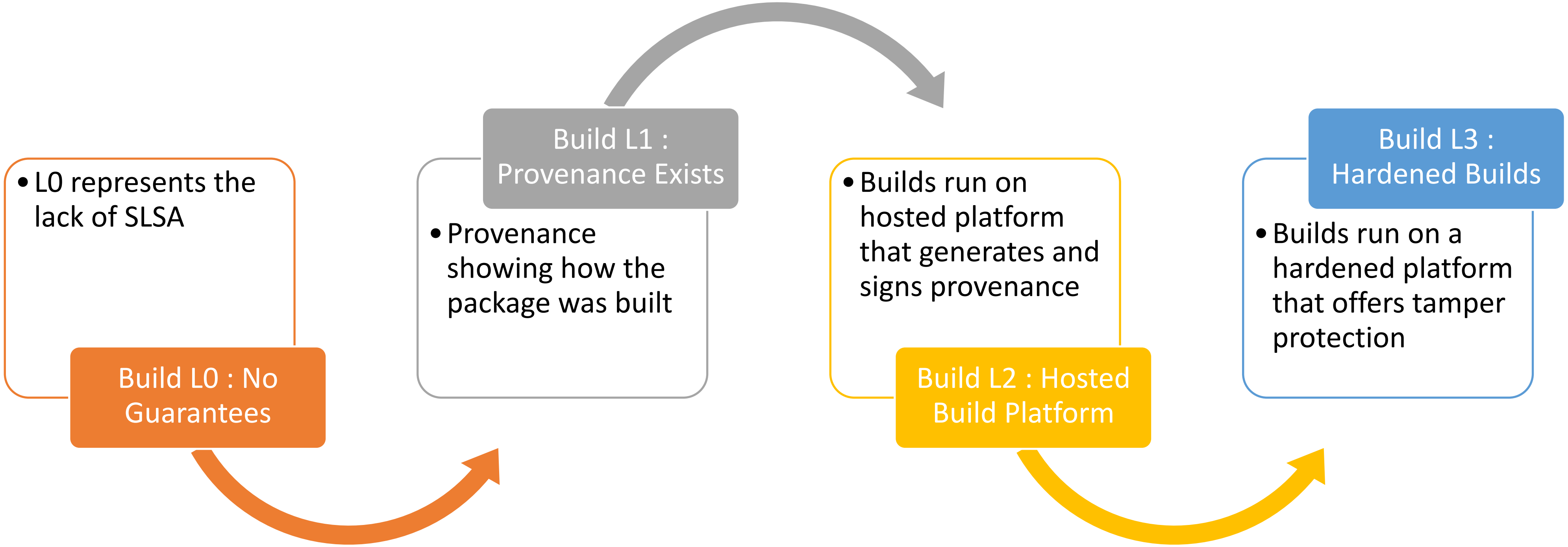
G Compromise package repo

H Use compromised package

	Known Examples of Attacks
A	SushiSwap: Contractor with repository access pushed a malicious commit redirecting cryptocurrency to themself.
B	PHP: Attacker compromised PHP's self-hosted git server and injected two malicious commits.
C	Webmin: Attacker modified the build infrastructure to use source files not matching source control.
D	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 (i.e. attack F).
E	SolarWinds: Attacker compromised the build platform and installed an implant that injected malicious behavior during each build.
F	CodeCov: Attacker used leaked credentials to upload a malicious artifact to a GCS bucket, from which users download directly.
G	Attacks on Package Mirrors: Researcher ran mirrors for several popular package repositories, which could have been used to serve malicious packages.
H	Browserify typosquatting: Attacker uploaded a malicious package with a similar name as the original.

SLSA Specification

SLSA (Supply Chain Levels for Software Artifacts) is a specification for describing and incrementally improving supply chain security, established by industry consensus. It is organized into a series of levels that describe increasing security guarantees.



Build L0 : No Guarantees

- L0 represents the lack of SLSA

Build L1 : Provenance Exists

- Provenance showing how the package was built

Build L2 : Hosted Build Platform

- Builds run on hosted platform that generates and signs provenance

Build L3 : Hardened Builds

- Builds run on a hardened platform that offers tamper protection

Public