Supply Chains

- Supply chain attack:
  - Nefarious alteration of trusted software before delivery
- Open-source supply chain vulnerability:
  - Exploitable weakness in trusted software caused by an open source component
  - The software itself doesn't have to be open source

Open Source Supply Chain Security at Google

- Understand, strengthen, and monitor the software supply chain
- Understand: understand where things are coming from and where attacks could target
  - Look at:
  - Build graph (all of the libraries being built to make a program)
  - Server graph (all of the servers and what they are communicating with each other)
  - Dependency graph (which open source components are being used to build a program)

Defending Supply Chain

- Most powerful technique: cryptography signatures
  - Removes downloading infrastructure concerns
  - Go checksum database
    - Holds SHA256 of every publically available go module
    - Signed by Google's private key
    - Public key is hard-coded in Go distribution
    - Database is updated whenever a module is published for the first time
      - Guarantees that everyone sees the same module and that no one person is getting a different version
- Computer system security:
  - Dedicated build systems should be isolated
  - Reproducible builds
    - Builds should work the exact same everywhere so anyone can verify them / reproduce them
- Two-person approvals
  - Code author and code reviewer (both of whom are Google employees) must look over code
- Capslock:
  - Static analyzer that looks at what functionalities each program can use
  - I.e. looking at log4j, why can this do remote downloads?

Finding Supply Chain Attacks

- OSS fuzz: automated continuous fuzzing for open source software
  - Automatically tries to find vulnerabilities
- Other tools to analyze kernel / system calls
- Language vulnerabilities:
  - C/C++: buffer overflow leads to remote code execution
  - Java: misuse of reflection / code loading leads to RCE
  - Go / Rust: memory safe, issue is just large / malformed inputs can lead to DOS attacks

xz attack

- Goal: take over sshd
  - Can't take over OpenSSH, instead take over xz-utils which is used as a part to build it

- Attackers targeted this because it was a single burnt out maintainer
- They spent years taking over maintenance of this by submitting real patches / doing maintenance
- Eventually hacked the build script to add this in