Software/Hardware Supply Chain Security

Laurie Williams, North Carolina State University

Akond Rahman, Tennessee Tech University

Navid Asadi, University of Florida

Many thanks to: Tamzidul Hoque, University of Kansas

Goal

This breakout group focused on identifying **similarities** and **differences** between software and hardware supply chain security so the security community ***can share a common vocabulary*** *between government (NSF, DoD, DHS, NIST), industry, academia*, leverage the research in overlapping areas, and consider the convergence of software/hardware supply chain into a joint threat model.

Insight

Software supply chain attacks historically have been attackers finding and exploiting unintentionally-injected vulnerabilities and is increasingly including intentionally-injecting and exploiting vulnerabilities.

Hardware supply chain attacks are moving to increasingly include finding and exploiting unintentionally-injected vulnerabilities (a.k.a Spectre/Meltdown).

Common definitions are hard!

- … starting from “supply chain”
- spent many minutes on each term

More similar

Build infrastructure: In hardware, electronics production involves tools and methods used to design, fabricate, and test electronics components and systems. In software, build infrastructure includes tools and scripts to compile, build, and deploy a product. Nefarious instructions can be injected into these tools/methods/scripts to result in a malicious artifact.

Less similar

Counterfeiting: A counterfeit piece of hardware may be known as an explicit substitution for the desired/authentic product made for monetary benefits. In software, the term counterfeit is not generally used (see malicious clone)

Malicious clones: In hardware, the clone will have duplicate functionality through unauthorized access to the “design”; the designer loses their intellectual property but the user is not harmed. In software, clones are duplicated packages often copied and re-deployed via typosquatting, forking, etc.; and the user is deceived and does not receive an authentic, supported product/package and may receive a version containing intentionally-injected vulnerabilities.

Proposed future work

Workshop to fully explore these common (or dissimilar) definitions … to achieve our original workshop goal