Securing the Software Package Supply Chain for Critical Systems using Permissioned Blockchains

Akash Ravi
Computer and Information Technology, Purdue University, West Lafayette

Problem Statement

- Critical Infrastructures increasingly make use of complex software systems to support their operations.
- These systems often make use of external/public software modules or packages to help them abstract common software functionalities.
- While this has numerous benefits, it opens up an attack surface and this software supply chain can be compromised to distribute malware masqueraded as package updates or add-ons.
- With multiple interlinks and complex workflows, any “downstream” systems that rely on these software solutions are also at risk of catastrophic failure.

Current Security Landscape

- Package distribution registries such as NPM and PyPI have implemented measures to validate package dependency chains, perform malware scanning, and audit usage history.
- Offerings such as the Snyk Intel vulnerability database, Sonatype OSS index and The Update Framework (TUF) provide ways to implement checks and balances.
- Researchers have experimented with next-gen firewalls, OS hardening, hardware checksums, and even InterPlanetary File Systems (IPFS) based methods to verify the security of software packages.

Proposed Solution

- The proposed blockchain-based architecture enforces controls by splitting the stakeholders into 4 different discrete entities: Publishers, Package Registries, Observers, and Users/Developers
- Multiple observers scan, verify and attest to the security of packages on the permissioned ledger.
- Users will be able to read/verify off the blockchain as appropriate to enforce checks.
- Observers will have a dynamic rank and combined with a Proof of Authority (PoA) consensus algorithm, the multi-party signature (MPS) for a block commit is expected to be realizable and secure.
- The advantages include a narrow distribution of trust, a zero-trust approach, a contribution to Open Source Intelligence (OSINT), and non-intrusive compatibility with existing frameworks.

Protection Against Malicious Entities:

- Deployment systems can interface with the ledger to verify any package being installed.
- For zero-day vulnerabilities discovered in used packages, post install hooks can be used to periodically scan deployments.
- In adverse attacks, an observer itself could be compromised and act maliciously. The proposed architecture mitigates this using MPS and PoA.
- All attack scenarios have been simulated and validated by integrating Corda and NPM install scripts.

Selected References: