Security Issues of Package Managers	Attack Methodology	Mitigation	Successful Attacks	Compatibility with BI (Y/N)	Comments
Typo-Squatting	Attacker masquerades as developer, uploads lexically similar package name, but with malicious content.	Double checking the package name before installing	On NPM's package "cross-env", with a misspelled package "crossenv"	Y	Packages with typos can be verified for security issues, but advisable to avoid usage.
Man-In-The-Middle (MITM) Attack	Attacker masquerades as an official mirror site, forwarding his own tampered files to "answer" the clients' requests.	Use https sources if possible. However, depends on security of connection between client and PM server.	No known history of MITM attacks regarding Package Managers. Feasibility is confirmed through several research papers.	N	MITM attacks may not always occur during Build Inspector testing but may install during actual installation.
Metadata Replay (Subset of MITM)	Attacker signs metadata of outdated packages/ vulnerable packages and replays them to client.	Constantly check and update repository metadata from the package manager distribution.	No known history of MITM attacks regarding Package Managers. Feasibility is confirmed through several research papers.	Y	No repercussions as signed packages will be inherently trusted.

Mirror Control (Requires compromised mirror)	Attacker with resources sets up and officialises his own mirror site. With root control attacker can upload malicious/ outdated packages for exploitation on a later date.	Using official or trusted sites would ensure safety from compromised mirrors. Using trusted external providers such as TheUpdateFramework also prevents usage of malicious mirrors.	No known history of mirror control attacks regarding Package Managers. Feasibility is confirmed through several research papers.	Y	
Endless Data (Requires compromised mirror)	An attacker returns an endless stream of data instead, when requested for a package. Client is denied package and data stream consumes large amounts of data in local disk.	Using a separate filesystem for package installation/ package managers may mitigate this attack specifically. One can also take other measures such as using Transport Layer Security (TLS) on connections to repositories, and avoid repositories served over plain text HTTP. Validating repository communication, e.g. checking file sizes and data rates will help too.	No known history of endless data attacks regarding Package Managers. Feasibility is confirmed through several research papers.	N	Attack exploits flaw in PM design – Occurs when files are requested

Extraneous Dependencies (Requires PMs with unsigned metadata)	False dependency information calling other packages are included. These dependencies can then call any package the attacker decides, e.g. one listed on his own server.	Use PMs that are known to sign their metadata. Clients or package managers should utilize TLS to access or host their repositories.	No known history of extraneous dependency attacks regarding Package Managers. Feasibility is confirmed through several research papers.	Y	Manually checking and verifying signatures on metadata would work too but defeats the convenience that Package Managers provide.
Unsatisfiable Dependencies (Requires PMs with unsigned metadata)	Attacker includes metadata pointing to non-existent dependencies. Prevents PM from installing the package.	Use PMs that are known to sign their metadata. Alternatively, download packages from source and install them from an offline basis.	No known history of unsatisfiable dependency attacks regarding Package Managers. Feasibility is confirmed through several research papers.	N	Attack exploits flaw in PM design – Occurs as an error to the user, which reduces suspicion of attack.

Provides Everything (Requires PMs with unsigned metadata)	Attacker changes metadata to falsely indicate that the attacker's package resolves a large number of dependencies. This causes the package to be installed, believing that it resolves dependencies that other packages require. The attacker's package can contain malicious codes, or call on mirror sites holding malicious packages.	Use PMs that are known to sign their metadata.	No known history of provides everything attacks regarding Package Managers. Feasibility is confirmed through several research papers.	Y	Network activity and dependencies to resolve can be monitored by build inspector.