

Forum:[Hack Forums](#)

Thread: [Detecting vulnerabilities in npm package chains](#)

Hackers - December 15, 2025, 8:13 am

I created a set of whitehat packages for npm, but I did so to highlight how package chains can be used for auto-update capabilities, without user approvals, notifications, etc. This makes it easier to hotfix a package, but also creates a possible vulnerability.

This package is 100% whitehat and does not take advantage of any sort of vuln or exploit. It just serves as an example of the risk of a whitehat tool being altered in the future.

Top Layers - Simple helper packages for getting the current price of Bitcoin.

Coinbase: <https://www.npmjs.com/package/cb-btc-value>

CoinGecko: <https://www.npmjs.com/package/coingecko-bitcoin>

CryptoCompare: <https://www.npmjs.com/package/cryptocompare-btc-price>

All of these have a value() function. Using is very simple:

```
Quote:import value from 'cb-btc-value';
```

```
const coinbasePrice = await value();
```

```
console.log("Coinbase Price: ", coinbasePrice);
```

These 3 packages rely on an underlying package, nodes-get-bitcoin:

```
Code "dependencies": {
```

```
  "nodes-get-bitcoin": "^1.0.0"
```

```
}
```

<https://www.npmjs.com/package/nodes-get-bitcoin>

This package uses a tool I created that can pull in all current Bitcoin prices from the public APIs and return an object. Each of the 3 packages above just extracts the single platform value from this object.

```
Code import { run } from 'nodes-url-runtime';
```

```
export default async function getBitcoin() {
```

```
  const module = await
```

```
run('https://raw.githubusercontent.com/LNodesL/public-bitcoin-prices/refs/heads/main/get-btc.js');
```

```
  if (module && typeof module.getBTCAverage === 'function') {
```

```
    return await module.getBTCAverage();
```

```
  } else {
```

```
    throw new Error('getBTCAverage is not available');
```

```
  }
```

```
}
```

As you can see, this nodes-get-bitcoin does not have the functionality built-in. Instead, it pulls code from a URL and uses that instead. This is where the vulnerability is... I could swap the code found inside get-btc.js to include any logic I want for getBTCAverage(), including running malicious code and then returning the expected data too.

```
Code "dependencies": {
```

```
  "nodes-url-runtime": "^0.0.3"
```

```
},
```

nodes-url-runtime is a helper package I created for downloading any JS and handling it properly to either run directly, or to use it for the module exports. It auto installs all detected imports/requirements for dependencies and installs before running. This lets the script (in my case, get-btc.js) to be updated with new dependencies, without forcing a re-install (npm install) or rebuild (npm build) or restart (npm start) in runtime production.

Because nodes-url-runtime can handle dynamic package installs in production, that lets nodes-get-bitcoin auto-update itself if needed. I simply need to update the source in the fetched URL, and all packages relying on nodes-get-bitcoin can benefit from the enhancements. The problem would be if someone malicious wanted to, they could update get-btc.js and inherit runtime permissions for the apps using it.

I hope this thread serves as a good, but basic example of how npm package chaining works, and how easy it is to set something up that could potentially be hacked later. For example, if someone stole my Github credentials, they could modify the source to affect anyone using my BTC price checkers, without me being responsible. Or, if I was a blackhat (I am not), I could do this after racking up a high number of installs.

Next steps for a proof of concept would be to create more packages, or repos, such as frontend Next.js builds that use my BTC price checker packages as dependencies. I am not doing that because I really just wanted to create this thread and also have a valid BTC price checker on npm. I've accomplished that, but I'm sure you can imagine how complicated this could get once a core package is used for another package, which is in another package, which is used in other builds such as starter kits for devs on Github.

Stanley - March 30, 2026, 6:03 am

I'm Stanley HF's helper bot. The OP (original poster) of this thread wants everyone to see this page and has elected to bump it to the top of the showthread listing. Since HF doesn't want a ton of LQP (Low Quality Posts) this is the best method of bumping your content. There is a small ?yte fee for each bump. Any OP can bump any of their threads on HF from the linked icon on the top of their first post.

If you haven't already read the first post of this thread we encourage you to do so.

Thanks,
Stanley

Biggy_Asian - March 30, 2026, 6:05 am

Very interesting project, could prove useful in the future.

yigliman - March 30, 2026, 9:14 am

I'd like to know how to obfuscate my .exe file or how to make it undetectable by antivirus software.