

Node.js: validazione e uso dei comandi da shell con parametri dinamici

In questo articolo vedremo come evitare una cattiva pratica nell'uso dei comandi da shell con parametri dinamici in Node.js.

Se si utilizzano comandi da shell con inserimento di parametri dinamici, occorre sempre validare questi parametri al fine di evitare un attacco basato su shell injection.

Ad esempio:

```
'use strict';

const { exec } = require('child_process');

const ping = host => {
  return new Promise((resolve, reject) => {
    const cmd = `ping -c 1 ${host}`;
    exec(cmd, (error, stdout, stderr) => {
      if (error) {
        reject(error);
      }
      if (stderr) {
        reject(stderr);
      }
      resolve(stdout);
    });
  });
};
```

```
});  
};
```

Occorre verificare che il parametro `host` sia o un valido indirizzo IP o un valido nome a dominio. Per farlo possiamo utilizzare un modulo NPM come `validator`.

```
const validator = require('validator');  
  
const ping = host => {  
  return new Promise((resolve, reject) => {  
    if(!validator.isFQDN(host) ||  
    !validator.isIP(host)) {  
      reject('Invalid host');  
    }  
    const cmd = `ping -c 1 ${host}`;  
    exec(cmd, (error, stdout, stderr) => {  
      if (error) {  
        reject(error);  
      }  
      if (stderr) {  
        reject(stderr);  
      }  
      resolve(stdout);  
    });  
  });  
};
```