20

Server Side Request Forgery in Uppy npm module

Share:

TIMELINE

sl4m-s4l3m submitted a report to Node.js third-party modules. Jan 31st (3 years ago)

Hi Team,

While we were testing our security engine at Shieldfy (https://shieldfy.io), We found a server side request forgery (SSRF) vulnerability in Uppy npm package.
It allows hacker to easily extract inside information from the server or take control of internal services.

Module

module name: Uppy
version: Latest: 1.8.0
npm page: `https://www.npmjs.com/package/uppy`

Module Description

Uppy is a sleek, modular JavaScript file uploader that integrates seamlessly with any application. It's fast, easy to use and lets you worry about more important problems than building a file uploader.

Module Stats

[1] weekly downloads : 23,153

Vulnerability

Server Side Request Forgery (SSRF)

Vulnerability Description

in the source code of the module
file: packages/@uppy/companion/src/server/controllers/url.js line: 11

You will find the express is routing the `/get` endpoint to the function `get` declared in line 43

Then it calls `downloadURL` in line`61 and pass `req.body.url` to it as argument

in the function `downloadURL` declared in line 80

It calls the url directly without any kind of sanitization or validation, opens the door to send malicious ssrf attack, allowing the hacker to extract information from any internal resource, or take control of any internal service.

Steps To Reproduce:

1. deploy the module in live server (ex: digital ocean server)
2. request 'Add More button' then click on `Link button`
3. Submit Link of DigitalOcean metadata api `http://169.254.169.254/metadata/v1/`
4. once done uploading , download the file you should see the content of the server metadata

Code 98 Bytes Wrap lines Copy Download

```
id
hostname
user-data
vendor-data
public-keys
region
interfaces/
dns/
floating_ip/
tags/
features/
```

Patch

The suggested fix.

1. use whitelist technique in the url protocol (allow only http & https), and on the port (80 & 443)
2. use blacklist technique in the host (disable IPs v4 & v6 allowing only domains, disable domains that used as internal routing if any)
3. disable redirection `followAllRedirects` to avoid bypasses

Supporting Material/References:

More info about ssrf can be found here : https://shieldfy.io/security-wiki/server-side-request-forgery/server-side-request-forgery/

Wrap up

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

Impact

- Scan local or external network
- Read files from affected server

nochnoidozor posted a comment. Jan 31st (3 years ago)

Hi @eslam-shieldfy,

Thank you for your submission. Your report is currently being reviewed and the HackerOne triage team will get back to you once there is additional information to share.

Kind regards,
@nochnoidozor

nochnoidozor changed the status to ○ Needs more info. Jan 31st (3 years ago)

Hi @eslam-shieldfy,

could you detail how to deploy the vulnerable module and provide a PoC js that imports the `uppy` module and showcase the vulnerability?

Thanks for your collaboration,
@nochnoidozor

3sl4m-s4l3m changed the status to ○ New. Feb 2nd (3 years ago)

Yea sure, The vulnerability basically in the companion server (proxy server) built for Uppy.
Here is the instruction to deploy it. Inspired by the documentation here

1. install companion server

Code 35 Bytes Wrap lines Copy Download

```
sudo npm install -g @uppy/companion
```

2. add sample config

Code 325 Bytes Wrap lines Copy Download

```
{
    "providerOptions": {
      "google": {
        "key": "***",
        "secret": "***"
      }
    },
    "server": {
      "host": "localhost:3020",
      "protocol": "http"
    },
    "filePath": "folder",
    "sendSelfEndpoint": "localhost:3020",
    "secret": "mysecret",
    "uploadUrls": [""],
    "debug": true
  }
```

3. run the server with the sample config

Code 28 Bytes Wrap lines Copy Download

```
companion --config conf.json
```

Now you can make simple html to load the plugin

Code 857 Bytes Wrap lines Copy Download

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Uppy</title>
    <link href="https://transloadit.edgly.net/releases/uppy/v1.8.0/uppy.min.css" rel="stylesheet">
  </head>
  <body>
    <div id="drag-drop-area"></div>

    <script src="https://transloadit.edgly.net/releases/uppy/v1.8.0/uppy.min.js"></script>
    <script>
      var uppy = Uppy.Core()
        .use(Uppy.Dashboard, {
          inline: true,
          target: '#drag-drop-area'
        })
        .use(Uppy.Url, {
            target: '#drag-drop-area',
            companionUrl: 'http://localhost:3020',
            locale: {}
        })
        .use(Uppy.Tus, {endpoint: 'https://master.tus.io/files/'})

      uppy.on('complete', (result) => {
```

```
  </body>
</html>
```

Note: we used `master.tus.io` just to upload the files , but the vulnerability exists in any provider you use with companion

Now just open the html and add the internal url inside the box

Now click upload, when upload is done click again on the file to download it you will find the content.

note2:
I created a dummy text file and make it available at http://127.0.0.1:3000/x.txt as a demonstration, you are free to put any url

Impact:
You can access any internal service (service meta data, Redis .... etc)

Please let me know if you need further info , or if you need me to upload it to dummy server.

1 attachment:
F703760: Screen_Shot_2020-02-02_at_12.17.46_PM.png

sl4m-s4l3m posted a comment. Feb 5th (3 years ago)
Any updates on the issue ?

asr0x01 posted a comment. Feb 5th (3 years ago)
Thanks for the input @eslam-shieldfy, I am now taking a second look and will get back to you as soon as possible.

Regards,
@nasr0x01

asr0x01 changed the status to ○ **Needs more info**. Feb 5th (3 years ago)
Hello @eslam-shieldfy,

Your time and effort in submitting this report are much appreciated, however, is it possible to share a video PoC demonstrating the vulnerability reported? for some reasons I am having issues with setting up the environment.

Your input is much appreciated.

Regards,
@nasr0x01

sl4m-s4l3m changed the status to ○ **New**. Feb 8th (3 years ago)
Hi @nasr0x01
No worries at all, our end goal here is to make the internet more secure :)

I Uploaded a POC at a dummy server, you can access it here : http://167.71.177.19:3000

- click on the open modal
- click on link
- add the url you want, the server hosted on digital ocean so you can try digital ocean meta data: http://169.254.169.254/metadata/v1/

I Also recorded a POC video here:
https://www.loom.com/share/eadfa6373a6f444b975e6fec41999ee0
password: uppy-h1

The code I used for the POC is also uploaded here in the secret gist: https://gist.github.com/netcode/1c8c28943f82ed24f77773c28f031168

Please let me know if you need any further info.

Best,

sl4m-s4l3m posted a comment. Feb 10th (3 years ago)
Any updates ??

nasr0x01 updated the severity from Critical to High (8.2). Feb 11th (3 years ago)

asr0x01 changed the status to ○ **Triaged**. Feb 11th (3 years ago)
Hello @eslam-shieldfy,

Thank you for your submission! We were able to validate your report, and have submitted it to the appropriate remediation team for review. They will let us know the final ruling on this report, and when/if a fix will be implemented. Please note that the status and severity are subject to change.

Regards,
@nasr0x01

sl4m-s4l3m posted a comment. Feb 14th (3 years ago)
Any updates ?

ifedapoolarewaju joined this report as a participant. Feb 18th (3 years ago)

ifedapoolarewaju posted a comment. Feb 18th (3 years ago)
Hello, member of the Uppy team here,

3sl4m-s4l3m posted a comment. Feb 19th (3 years ago)

Thanks for the updates, Please keep me in the loop.
If you want any help don't hesitate to ask.

marcinhoppe Node.js third-party modules staff posted a comment. Feb 19th (3 years ago)

@ifedapoolarewaju thanks for a quick update! Let us know if we can help in getting the fix tested. After the patch has been made available, we will coordinate responsible disclosure.

ifedapoolarewaju posted a comment. Feb 19th (3 years ago)

@marcinhoppe @eslam-shieldfy Update: A PR has been submitted to mitigate this issue https://github.com/transloadit/uppy/pull/2083 .

It's still awaiting review from the Uppy team, but maybe you can also test from that branch already? This would mean copying the contents of this `src` directory https://github.com/transloadit/uppy/tree/validate-url/packages/%40uppy/companion/src into your `node_modules/@uppy/companion/lib/` and running the library as you were already doing.

PS: when `debug` is set to `true` from the options listed here https://hackerone.com/reports/786956#activity-6942808, Companion will assume the library is being run in a dev environment, hence it will allow download URLs like `http://localhost:3000` or `http://127.0.0.1:3000/file-name`

3sl4m-s4l3m posted a comment. Feb 20th (3 years ago)

@ifedapoolarewaju I was about to point DNS pinning but @ Acconut already mention it on the PR.
I proposed a solution for it in the PR.
Also I'm happy to help anytime here or on the PR.

marcinhoppe Node.js third-party modules staff posted a comment. Feb 26th (3 years ago)

@eslam-shieldfy the PR has been merged. Can you confirm that this vulnerability has been fixed?

ifedapoolarewaju posted a comment. Feb 26th (3 years ago)

@marcinhoppe @eslam-shieldfy Also the patch has been released to npm
https://github.com/transloadit/uppy/blob/a88d564962a44959236206e8ea689d8d91a05279/CHANGELOG.md#193

3sl4m-s4l3m posted a comment. Feb 27th (3 years ago)

@ifedapoolarewaju
Great work, I can confirm that the vulnerability is not fixed.

@marcinhoppe
lets coordinate the disclosure for the issue, release the advisory and requesting the CVE.

Thanks

marcinhoppe Node.js third-party modules staff closed the report and changed the status to **Resolved**. Feb 28th (3 years ago)

marcinhoppe Node.js third-party modules staff requested to disclose this report. Updated Feb 28th (3 years ago)

Let' disclose it then. I will request a CVE once this goes public.

marcinhoppe Node.js third-party modules staff disclosed this report. Mar 2nd (3 years ago)

marcinhoppe Node.js third-party modules staff posted a comment. Mar 2nd (3 years ago)

I requested a CVE.

3sl4m-s4l3m posted a comment. Mar 3rd (3 years ago)

Thanks @marcinhoppe

marcinhoppe Node.js third-party modules staff changed the scope from **None** to **Uppy**. Jul 10th (2 years ago)