Fredi Arro

CVE-2017-5638

Essay
## Contents

Contents ............................................................................................................................................ 2  
Introduction ...................................................................................................................................... 3  
Description of the vulnerability ........................................................................................................ 4  
Solutions ........................................................................................................................................ 6  
Summary ......................................................................................................................................... 7  
Used sources ................................................................................................................................... 8
Introduction

On the 6-th of March, in 2017, a new RCE (remote code execution) vulnerability in Apache Struts was made public. (Imperva, 2017). Apache Struts is an open source project of the Apache foundation Jakarta project team. It is used by Java-based Web applications (Qualys, 2017). The vulnerability affects the versions of Apache Struts 2 2.3.x before 2.3.32 and 2.5.x before 2.5.10.1 (CVE, 2017).

The vulnerability very wide-spread, because fixing the issue isn’t as easy as just installing a patch, although it is the main recommended way to fix the issue. The problem is that if you have made any custom changes on the Struts source code, it could take a long time, days or weeks, to resolve the vulnerability (eWEEK, 2017). The purpose of this essay is to give an overview of the vulnerability and how to resolve the issue.
Description of the vulnerability

CVE-2017-5638 is an Apache Struts 2 vulnerability which exists in the Jakarta Multipart parser. It exists because of improper handling of the Content-Type header\(^1\) (Qualys, 2017). The attacker can use malicious OGNL\(^2\) in Content-Type header to trigger this vulnerability and then execute the system command. (Qualys, 2017). This can result in an unauthenticated, remote attack.

The main point was that the attacker could send request with a custom made Custom-Type header. The Struts would give an error message to the user, when the upload, using multipart HTTP request\(^3\) is not successful. To provide the right error message, Struts used function “findText” to parse the error key messages and get the defined error message for it. (Trend Micro, 2017)

```java
public static String findText(Class aClass, String aTextName,
Locale locale, String defaultMessage, Object[] args )
```


The function would find a locally stored message for the given error key passed in variable “aTextName”. The parameters are:

- *aClass*: *FileUploadInterceptor\(^4\)* passes the error class to use as the start point of the error message search
- *aTextName*: the error key used to find the error message for it
- *Locale*: the locale where the message is saved for the respective error key
- *defaultMessage*: the message to be returned if no text message can be found in any resource bundle
- *args*: the error arguments’ resource bundle

(Trend Micro, 2017)

---

\(^1\) Content-Type is an HTTP header field, which is the header section of request and response messages (Wikipedia, 2018).

\(^2\) OGNL is Object-graph Navigation Language. It’s an open source Expression Language for Java. It doesn’t have the wide range of commands that are supported by Java, but it allows getting and setting properties, and execution of Java Methods. (Wikipedia, 2018)

\(^3\) A HTTP mutlipart request is a request that HTTP clients construct to send files and data over to a HTTP server. (Stack Overflow, 2013)

\(^4\) *FileUploadInterceptor* is the file where the *findText* function was used, before the patch
If the message is found, then those arguments will be treated as an OGNL expression. So if the attacker can use malicious OGNL in Content-Type header to trigger this and then execute commands on the server. (Qualys, 2017)

Using this vulnerability, the attacker could access different data on the server. For example the big Equifax breach was due to this. The attackers gained access to people’s social security number, birth dates and even credit card information. About 140 million people were affected. The data breach was so big because the vulnerability gives attackers full access to the server(s).
Solutions

Apache released the patch for this vulnerability in the same day that it was made public. The official way to fix the issue is to update Apache Struts then we are going to take a closer look, what changes were made.

The function that was the main point of the vulnerability “findText” was removed with the usage of Class “Localized TextUtil” form FileUploadInterceptor.java. (TrendMicro 2017)

```java
if (multiWrapper.hasErrors()) {
  if (multiWrapper.hasErrors() && validation != null) {
    TextProvider textProvider = getTestProvider(action);
    for (LocalizedMessage error : multiWrapper.getErrors()) {
      if (validation != null) {
        validation.addActionError(LocalizedTextUtil.findText(error.getClassName(), error.getKey(), actionContext.getMessage().getLocale(), error.getArgs()));
      } else {
        errorMessage = textProvider.getText(error.getClassName(), actionContext.getMessage().getLocale(), error.getArgs());
      }
    }
    validation.addActionError(errorMessage);
  }
}
```

FileUploadInterceptor.java file differences before and after the patch (red are removed lines, and green are added lines) (TrendMicro 2017)

Also is removed “java.io.File” which could be used to output the executed code for the attacker. (TrendMicro, 2017) As it is seen from the difference picture they wrote new error message receiving function. The loop doesn’t use function findText from LocalizedTextUtil class, instead they use getText function, which gets the errors from certain array without parsing the Content-Type header. Or in else case they get the default message.
Summary

The Apache Struts 2 vulnerability was a big threat and it was quickly patched, but probably there are still Web applications which still haven’t received an update. It was probably one of the most wide-spread vulnerabilities related to Java. Attackers can gain remotely execute their code using custom Content-Type header in HTTP request. They can add OGNL command in there and it would be executed in the Web application server. This gives them access to almost everything in the server.

Easiest way to fix the problem is to install the proper update from Apache. Since there are so many Web applications using Struts, and many have changed the source code, there are still applications with the vulnerability, because implementing the update takes a long time. Hopefully there won’t be any big issues from this vulnerability.
Used sources

  Used: 30.04.2018

  Used: 30.04.2018

  Used: 30.04.2018

  Used: 30.04.2018

  Used: 30.04.2018

  Used: 30.04.2018

  Used: 30.04.2018

  Used: 30.04.2018