ET4028 Host & Network Security Host Hardening and PEN Testing Assignment

Group 3:

Kevin McCarthy 19237219

Elton Babela 18196497

Fionn Murray 18223451

Contents

April 2023

Department of Electronics & Computer Engineering University of Limerick

1. The Apache Struts vulnerability (CVE-2017-5638)

Struts is vulnerable to remote command injection attacks because of incorrect interpretation of an improper Content-Type HTTP header sent by an attacker. These instructions may be run with full Web server capabilities because to a flaw in Struts. Since the vulnerability was made public, it has been used in the wild to execute full remote commands.

The Multipart parser for Jakarta contains the exploitable code. If the value of Content-Type does not correspond to a suitable type, an exception is thrown and the user is presented with an appropriate error message. **$(#_='multipart/form-data')** is the content type.

Why did this security hole appear in Apache Struts?

This flaw arises because **LocalizedTextUtil.findText** uses the Content-Type unescaped when an error occurs while constructing the error message. Everything within $... is treated as an Object Graph Navigation Library (OGNL) expression and evaluated by this function. Under these circumstances, the offender may run OGNL expressions, which in turn run commands on the system.

Furthermore, OGNL stands on its own as a very expressive and comprehensive language. The weapon is a trustworthy and efficient tool for the attacker. Many of the essential JAVA APIs, such as **java.core**, are available for **use.The** use of **ProcessBuilder()** allows a user-supplied software to be run on the computer.

Feedback from the server on the progress and output of instructions performed by the web server facilitates exploitation as well. Since no extra line of communication is needed, the risk of being seen is reduced, and inbound firewall restrictions may be avoided.

The above curl programme shows server vulnerability by making an HTTP request with an OGNL expression inserted in the Content-Type header and waiting for a response.

Each member of the **OgnlContext** JAVA object, which represents the execution context of the OGNL expression, is granted read/write access by default. It unloads unused code to make use of all available libraries and classes.

To show the secure manipulation of fundamental servlet parameters and to give a string to return to the user through the system echo command, the String representation of the container object is assigned to the variable "#eps" using its **toString()** function.

#eps=#container.toString()

"#cmds=("/bin/echo", "#eps")"

When you use the echo command, a JAVA ProcessBuilder object is created and used to print the specified String. Additional JAVA capability is used to reroute the input stream of this process to the output stream of the servlets response. The servlet is then able to counter the

attacker, as seen by the following:
com.opensymphony.xwork2.inject.ContainerImpl@d0d0b00

This command is an example of an exfiltration channel, Java capability, and remote command execution.

Since exploits and proofs of concept for this vulnerability are widely disseminated, the skill level necessary to launch an attack has dropped dramatically. The widespread use of Struts makes it vulnerable to both targeted and non-targeted assaults.

When should I use a patch for CVE-2017-5638?

This vulnerability may be reduced if Web application firewalls like **mod_security** were set up to permit just legitimate content types or to forbid OGNL expressions. Instead of updating Struts, you may use Jason Pells' multipart parser. The jar file for this plugin should be placed in the **/WEB-INF/lib** directory of your application, which will replace the vulnerable Struts component. The library must also be part of your application.

2. System Hardening

In order to create a more hardened machine, the following steps were taken to create a more protected system.

First, the password setting of the system were adjusted, to make them stricter on password choices, and enforcing frequent changes. The changes made can be seen in the figure below, where the victim account "vboxuser" must now change its password every thirty days, being warned of this action seven days in advance. The following commands were used to do this:

- chage -M 30 -m 7 -W 7 vboxuser

As another level of protection from file access on the victim machine, USB access was blocked on the machine to increase its level of physical protection. The following commands were used to accomplish this.

- sudo vim /etc/modprobe.d/blacklist.conf
- blacklist usb-storage

Logwatch was also installed on our victim machine to automatically read suspicious files or and logs that may appear on the system. This can be configured to email the user notifying them of the existence of any of these files. Logwatch can be easily installed and configured with the following commands in linux:

- sudo apt-get install logwatch
- sudo -u logcheck logcheck

The log file can then be viewed with

- sudo cat /var/mail/root

Finally, the system was also checked to ensure no Non-Root users have a UID of 0, granting them admin access. If any users other than the root appear this needs to be changed. This can be assessed with the following command.

- awk -F: '($3 == "0") {print}' /etc/passwd

3. Exploit Step-by-Step Instructions

To first set up the exploit, the following software was installed, and a web server was created to store the vulnerable file and allow for the exploit to take place.

First Java JDK version 8 was installed to the victim server machine. As well as this Apache tomcat and maven were to allow for the web server hosting. The software was configured on the victim to allow hosting from the machine

Next, the old version of struts 2 was installed, to deploy the vulnerable struts web app to the server.

Once setup, the web server is launched and the vulnerable file is deployed to allow for the exploit.

With this setup, we can easily make use of Metasploit in our attacking machine to exploit the vulnerability.

There are two options that must be set for this exploit as follows:

RHOSTS = Victim IP: 192.168.0.101

TARGETURI = /struts_2_2.3.15.1-showcase/

This sets the target for the exploit to the victim machine IP, as well as the vulnerable target file for the exploit.

After this you can run the exploit, which will then grant access to the admin terminal of the victim machine via the Metasploit meterpreter.

In order to confirm operation multiple files were created, downloaded, and run on the victim macine from the Metasploit meterpreter on the attacking machine.

This exploit was then demonstrated in the lab with success.

4. Proposed Solutions & Justifications

Patching or upgrading your Apache Struts framework to a version that contains the vulnerability fix is the first and most crucial step. It is advised to install the fix made available for this issue by the Apache Software Foundation right away.

Disable the susceptible functionality: You should disable the vulnerable functionality if you can't update or fix your Apache Struts framework. The component that is susceptible to this attack, the Struts REST plugin, can be disabled to achieve this.

Adopt network security measures: To reduce the likelihood of an attack on your server, you should adopt network security measures. To identify and stop malicious traffic, this includes establishing firewalls and intrusion detection systems.

Keep an eye on your system logs: Keep an eye out for any indications of shady activities. This involves keeping an eye out for odd HTTP requests, failures, or other potential attack signs.

Undertake routine security audits: To find any weaknesses that an attacker could exploit, you should routinely undertake security audits of your web apps and infrastructure. Conducting vulnerability scanning and penetration testing are examples of this.

In general, patching or updating your Apache Struts framework to a version that contains the fix for this issue is the most efficient way to mitigate this risk.