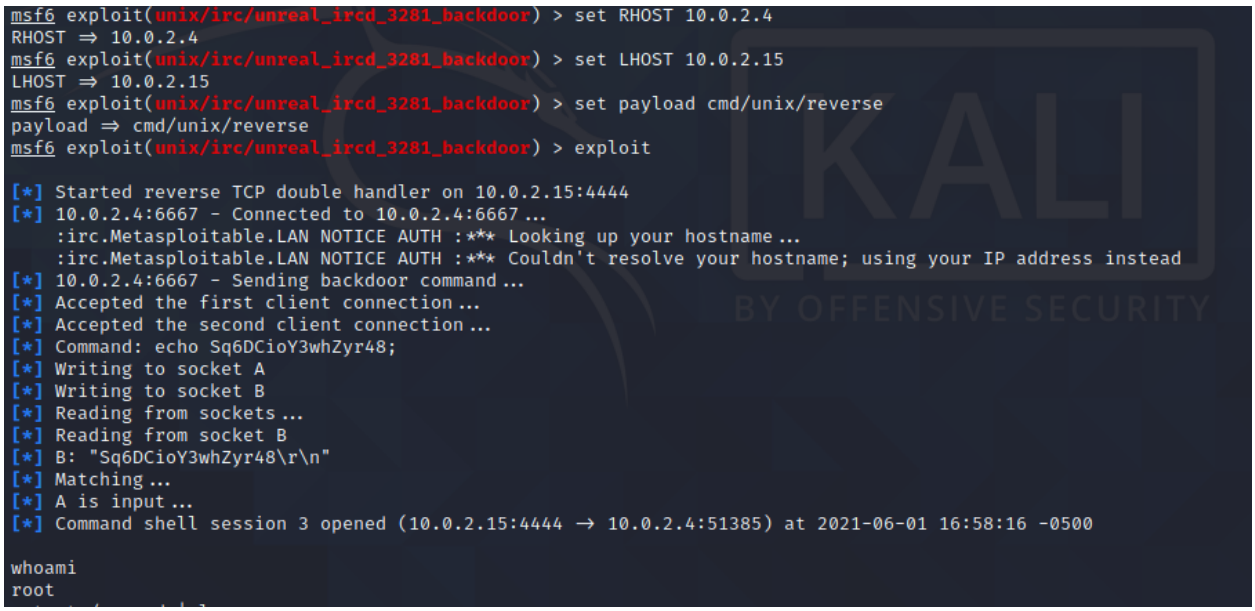
Kaeden Berg, Yasmeen Awad, Kitty Tyree

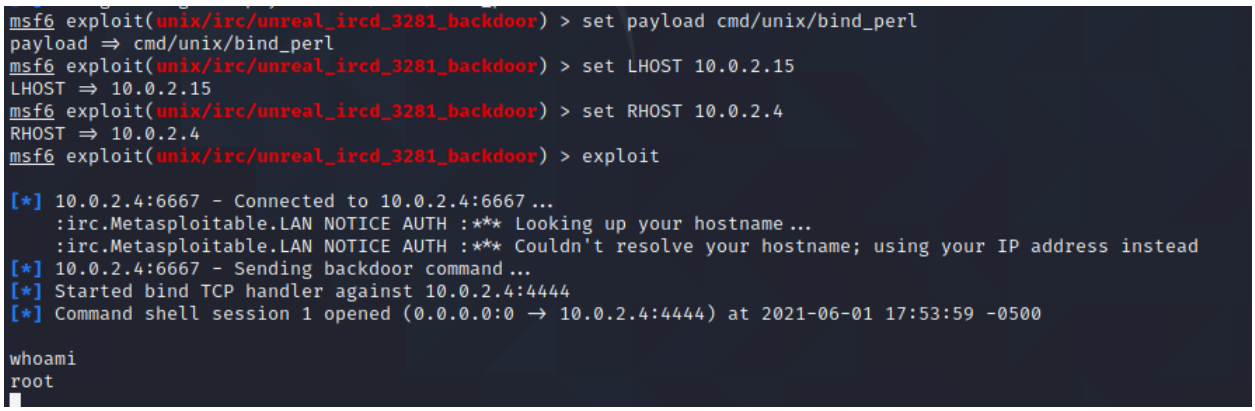
Part 2: find an exploit/two payloads

We decided to use exploit unreal_ircd_3281_backdoor

1. Simple, step-by-step instructions on how to perform the exploit with each of your chosen payloads. This might be a list of command-line commands, or a sequence of screenshots, etc. Shoot for clear, easy-to-follow instructions.
    - The first payload we used was **cmd/unix/reverse**



    - The second payload was **cmd/unix/bind_perl**



2. An explanation of how the exploit works. Not "Metasploit's X/Y/Z module does magic, and you get a shell!" Rather, you need to do the research on how the exploit in question takes advantage of some bug or misconfiguration on the target machine, and then share that research with me briefly and clearly, with citations as appropriate.
    - “This module exploits a malicious backdoor that was added to the Unreal IRCD 3.2.8.1 download archive.”

- http://www.computersecuritystudent.com/SECURITY_TOOLS/METASPLOITABLE/EXPLOIT/lesson7/
- So basically, there is already a backdoor in place and the exploit just opens it.

3. A brief description of each payload you tried out, and an explanation of how they differ.
   - **cmd/unix/reverse**
     - Sets up a listening port on the attacker's system and waits for incoming connections from the victim.
   - **cmd/unix/bind_perl**
     - Uses perl to spawn a command line on port 4444 of the target machine
   - **Differences between reverse and bind_perl**
     - In a reverse payload, the attacker waits for the victim machine to reach out in order to establish connection whereas in a bind payload, the attacker reaches out to the victim machine to open a listening port that will be used to establish connection. Therefore, the major difference is who is reaching out to who in the interaction.
4. A brief description of how you managed to transfer /etc/passwd to your attacking machine.
   - Turned on Kali Linux's SSH in order to get the file transferred and that way we could copy it through SSH using scp.

```
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::
news:*:14684:0:99999:7:::
uucp:*:14684:0:99999:7:::
proxy:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
irc:*:14684:0:99999:7:::
gnats:*:14684:0:99999:7:::
nobody:*:14684:0:99999:7:::
libuuid:!:14684:0:99999:7:::
dhcp:*:14684:0:99999:7:::
syslog:*:14684:0:99999:7:::
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
bind:*:14685:0:99999:7:::
postfix:*:14685:0:99999:7:::
ftp:*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:14685:0:99999:7:::
mysql:!:14685:0:99999:7:::
tomcat55:*:14691:0:99999:7:::
distccd:*:14698:0:99999:7:::
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:14699:0:99999:7:::
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:14715:0:99999:7:::
telnetd:*:14715:0:99999:7:::
proftpd:!:14727:0:99999:7:::
```

Part 3: how might your intrusion be detected?

- We struggled to find differences with this exploit using sp. Using ss shows us that new connections are established with port 4444 from Metasploit's IP address. This means that if someone was on metasploit, they could see that they are connected to us.
- The command ss returns a complete list of tcp sockets with established connections (listening ports). If you have connected to a machine you are not authorized on, an administrator may notice when they check all listening ports and see your unrecognized IP address.
- If you add a new user to the target machine, that will also be detectable by the machine's administrator/owner.

Part 4: something cool

- We initially tried to exploit vsftpd_234 via a backdoor (unix/ftp/vsftpd_234_backdoor), but there were no payloads, it just does its thing. It opened a backdoor the first time that we ran the exploit but did not open a session. So, we tried running exploit a second time and found a shell which opened a session for us to use. It was fairly easy. It worked like this because a backdoor was just built into Metasploit and therefore we don't need payloads.