



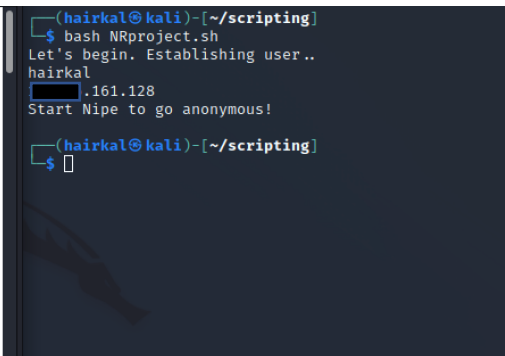
### Step 1.

I am calling my script, NRproject.sh located in /home/hairkal/scripting.

So I will be running my script in the scripting folder.

The 1<sup>st</sup> function is called 'iam'. It shows username and original ip before proceeding into going anonymous.

```
1  #!/bin/bash
2
3  # Summary.
4  # 1. establish your IP
5  # 2. start nipe
6  # 3. check if you are anonymous.
7  # 4. echo your anonymous IP and location
8  # 5. Run nmap/whois.
9  # 6. echo the answer and save result
10
11 # Step 1. establish your IP
12
13 function iam ()
14 {
15     echo "Let's begin. Establishing user.."
16     whoami
17     hostname -I
18     echo "Start Nipe to go anonymous!"
19 }
```



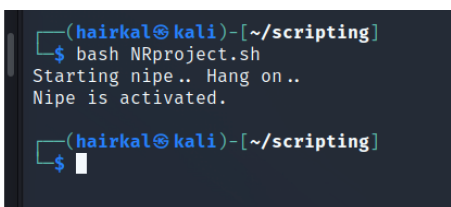
### Step 2.

After establishing the user, the script will then proceed to activate nipe. My nipe is located in /home/hairkal/nipe.

Therefore, I will need to specify the location in the script and activated nipe.

I also added on the variable \$active to be echoed to user. So that user is being prompt that nipe is activated.

```
23 # Step 2. start nipe
24
25 function startnipe ()
26 {
27     echo "Starting nipe.. Hang on.."
28     cd /home/hairkal/nipe
29     sudo perl nipe.pl start
30     active=$(sudo perl nipe.pl status | grep activated | awk '{print $3}')
31     echo "Nipe is $active"
32 }
33
```



### Step 3 & 4.

We now know that nipe is activated. But just to be sure , I have added in our anonymous checker using if conditions exercise that we previously did into use.

I added the variable \$anon which calls out our 'random/anonymous' ip to our user to confirm that we are indeed anonymous and echo new anonymous ip location

```

32 # Step 3 & 4. User wants to ensure that user is Anonymous. Echo location
33
34 function anonchecker ()
35 {
36     echo "Checking if you are Anonymous.."
37     cd /home/haikar/nipe
38     anon=$(sudo perl nipe.pl status | grep Ip | awk '{print $3}')
39     echo "You are now $anon"
40     location=$(geoipllookup $anon | awk -F: '{print $2}')
41     echo "located in: $location"
42     stat_check=$(sudo perl nipe.pl status | grep -w activated)
43
44
45     if [ ! -z "$stat_check" ]
46     then
47         echo "You are Anonymous.. Let's go!!"
48     else
49         echo "You are Expose!! Restart before you proceed.. "
50     fi
51 }

```

```

(haikal@kali)-[~/scripting]
$ bash NRproject.sh
Checking if you are Anonymous..
[sudo] password for haikar:
You are now 193.218.118.95
located in: UA, Ukraine
You are Anonymous.. Let's go!!

```

## Step 5 & 6

I will now be going into ssh after confirming that I am indeed anonymous. This function allows me to access the vps and proves that you are able to login as root for the remote server by using the whoami command.. And once I am in, it will run the case options continuously without interruption (ssh -T)

\*I also made use of my private and public key to authenticate my login without prompting password.

ssh – provides a default pseudo terminal / needs user interaction

ssh -t – Force pseudo-terminal allocation. This can be used to execute arbitrary screen-based programs on a remote machine, which can be very useful, e.g. when implementing menu services. Multiple -t options force tty allocation, even if ssh has no local tty.

ssh -T – Disable pseudo-terminal allocation.

So, I tried multiple variations into how this could work.

First solution:

1. Creating a script in the vps and run it on the vps itself. It works 😊
2. The script is now stored in the vps. So I tried calling the script and running it from my host terminal. It works 😊

Second solution:

1. Knowing I could call the script to run on the vps from my host. I decided to create the same script from my host. Scp the sh.file to the vps and execute it from my main script. It works 😊

Third solution: (it works in one script, so I went with this 😊)

1. Call out the case option before entering vps.
2. Once user has input the selection, store that variable.
3. Enter vps with said variable and run the script. Resulting, all being run on one whole script and echo and save result.

So here's how I did it.

```

32
33 # Step 5 & 6. Run nmap/whois. echo the answer
34
35 function options ()
36 {
37     read -p "Would you like to A) NMAP? or B) WHOIS? : " choose
38     case $choose in
39         A)
40             echo "IP address for NMAP: "
41             read IP
42             echo "$IP"
43
44             user="root"
45             server=
46
47             for target in "${server[@]}"
48             do
49                 ssh -T ${user}@${target}<<-END
50                 echo "You are in as.."
51                 whoami
52                 nmap "$IP" -oN nmapscan.txt
53             done
54
55             options
56             ;;
57         B)
58             echo "IP address for WHOIS: "
59             read IP
60             echo "$IP"
61
62             user="root"
63             server=
64
65             for target in "${server[@]}"
66             do
67                 ssh -T ${user}@${target}<<-END
68                 echo "You are in as.."
69                 whoami
70                 whois "$IP" > whoisresult.txt
71                 whois "$IP" | grep OrgName
72                 echo "$result"
73             done
74
75             options
76             ;;
77         C)
78             exit
79     esac
80 }

```

```

(hairkal@kali)-[~/scripting]
└─$ bash NRproject.sh
Would you like to A) NMAP? or B) WHOIS? : A
IP address for NMAP:
8.8.8.8
8.8.8.8
You are in as..
root
Starting Nmap 7.80 ( https://nmap.org ) at 2022-07-26 17:52 UTC
Nmap scan report for dns.google (8.8.8.8)
Host is up (0.0016s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
53/tcp    open  domain
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 4.51 seconds
Would you like to A) NMAP? or B) WHOIS? : B
IP address for WHOIS:
8.8.8.8
8.8.8.8
You are in as..
root
OrgName:      Level 3 Parent, LLC
OrgName:      Google LLC

Would you like to A) NMAP? or B) WHOIS? : C
(hairkal@kali)-[~/scripting]
└─$

```

```

52
53 # Step 5 & 6. Run nmap/whois. echo the answer
54
55 function options ()
56 {
57     read -p "Would you like to A) NMAP? or B) WHOIS? : " choose
58     case $choose in
59         A)
60             echo "IP address for NMAP: "
61             read IP
62             echo "$IP"
63
64             user="root"
65             server=
66
67             for target in "${server[@]}"
68             do
69                 ssh -T ${user}@${target}<<-END
70                 echo "You are in as.. "
71                 whoami
72                 nmap "$IP" -oN nmapscan.txt
73
74                 END
75             done
76
77         options
78         ;;
79
80         B)
81             echo "IP address for WHOIS: "
82             read IP
83             echo "$IP"
84
85             user="root"
86             server=
87
88             for target in "${server[@]}"
89             do
90                 ssh -T ${user}@${target}<<-END
91                 echo "You are in as.. "
92                 whoami
93                 whois "$IP" > whoisresult.txt
94                 whois "$IP" | grep OrgName
95                 echo "$wresult"
96
97                 END
98             done
99
100         options
101         ;;
102
103         C)
104             exit
105         esac
106     }
107 }
108

```

That basically summarize all of the step. Put it all together and run. →

Running Result: script is running on kali. Result stored in txt in vps.

The screenshot shows a Kali Linux terminal window on the left and a VPS terminal window on the right. The Kali terminal shows the execution of a script named NRproject.sh. The script sets user="root" and server="8.8.8.8", then loops through targets. It uses ssh to connect to the target, runs whois, and stores the results in whoisresult.txt. The VPS terminal shows the output of the script, including the IP address 8.8.8.8, the location RO, Romania, and the Nmap scan results for 8.8.8.8. The VPS terminal also shows the output of the whois command for 8.8.8.8, identifying it as Level 3 Parent, LLC and Google LLC. The Kali terminal also shows the output of the ls command, listing nmapscan.txt, snap, and whoisresult.txt.

```
(hairkal@kali)-[~/scripting]
$ bash NRproject.sh
Let's begin. Establishing user..
hairkal
161.128
Start Niipe to go anonymous!
Starting niipe.. Hang on..
[sudo] password for hairkal:
Niipe is activated.
Checking if you are Anonymous..
You are now 83.97.20.88
located in: RO, Romania
You are Anonymous.. Let's go!!
Would you like to A) NMAP? or B) WHOIS? : A
IP address for NMAP:
8.8.8.8
8.8.8.8
You are in as..
root
Starting Nmap 7.80 ( https://nmap.org ) at 2022-07-26 17:58 UTC
Nmap scan report for dns.google (8.8.8.8)
Host is up (0.0017s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
53/tcp    open  domain
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 4.60 seconds
Would you like to A) NMAP? or B) WHOIS? : B
IP address for WHOIS:
8.8.8.8
8.8.8.8
You are in as..
root
OrgName:      Level 3 Parent, LLC
OrgName:      Google LLC
Would you like to A) NMAP? or B) WHOIS? : C

(hairkal@kali)-[~/scripting]
$

root@Ubuntu-vps-1:~# ls
nmapscan.txt  snap  whoisresult.txt
root@Ubuntu-vps-1:~#
```

Anonymous ssh result ☺

The screenshot shows a Kali Linux terminal window with the output of the netstat -tapn command. The output shows active Internet connections (servers and established). The table below represents the data shown in the terminal:

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	127.0.0.1:9051	0.0.0.0:*	LISTEN	-
tcp	0	0	127.0.0.1:9050	0.0.0.0:*	LISTEN	-
tcp	0	0	172.16.161.128:44978	172.106.200.150:443	ESTABLISHED	-
tcp	0	0	172.16.161.128:34754	159.223.4.52:22	TIME_WAIT	-
tcp	0	0	172.16.161.128:42962	51.159.177.222:443	ESTABLISHED	-
tcp	0	0	127.0.0.1:48864	127.0.0.1:9050	ESTABLISHED	117107/ssh
tcp	0	0	127.0.0.1:9051	172.16.161.128:34746	ESTABLISHED	-
tcp	0	0	127.0.0.1:9050	127.0.0.1:48864	ESTABLISHED	-

Credits :

Torify ssh: <https://youtu.be/B9kogZO0oml>

Ssh private / public key : <https://www.cyberciti.biz/faq/how-to-set-up-ssh-keys-on-linux-unix/>

Executing scripts while in ssh: <https://youtu.be/o9H303Z9ukc>

```
#!/bin/bash

# Summary.
# 1. establish your IP
# 2. start nipe
# 3. check if you are anonymous.
# 4. echo your anonymous IP and location
# 5. Run nmap/whois.
# 6. echo the answer and save result

# Step 1. establish your IP

function iam ()
{
    echo "Let's begin. Establishing user.."
    whoami
    hostname -I
    echo "Start Nipe to go anonymous!"
}

# Step 2. start nipe

function startnipe ()
{
    echo "Starting nipe.. Hang on.."
    cd /home/hairkal/nipe
    sudo perl nipe.pl start
    active=$(sudo perl nipe.pl status | grep activated | awk '{print $3}')
    echo "Nipe is Sactive"
}

# Step 3 & 4. User wants to ensure that user is Anonymous. Echo location

function anonchecker ()
{
    echo "Checking if you are Anonymous.."
    cd /home/hairkal/nipe
    anon=$(sudo perl nipe.pl status | grep Ip | awk '{print $3}')
    echo "You are now $anon"
    location=$(geoplookup $anon | awk -F: '{print $2}')
    echo "located in: $location"
    stat_check=$(sudo perl nipe.pl status | grep -w activated)

if [ ! -z "$stat_check" ]
then
    else
        echo "You are Anonymous.. Let's go!!"
        echo "You are Expose!! Restart before you proceed.."
fi
}

# Step 5 & 6. Run nmap/whois. echo the answer

function options ()
{
    read -p "Would you like to A) NMAP? or B) WHOIS?: " choose
case $choose in
A)
    echo "IP address for NMAP: "
    read IP
    echo "$IP"

    user="root"
    server=""

    for target in "${server[@]}"
    do
        ssh -T ${user}@${target}<<-END
        echo "You are in as.."
    done

    whoami

    nmap "$IP" -oN nmapscan.txt

    END
done

options
;;

B)
    echo "IP address for WHOIS: "
    read IP
    echo "$IP"

    user="root"
    server=""
```

```

        whoami
        whois "$IP" | grep OrgName
        echo "$wresult"

        for target in "${server[@]}"
        do
            ssh -T ${user}@${target}<<-END
            echo "You are in as.."
            whois "$IP" > whoisresult.txt
        END
    done

options
;;

C)
    exit
esac
}

iam
startnipe
anonchecker
options

```