Thought process of the project

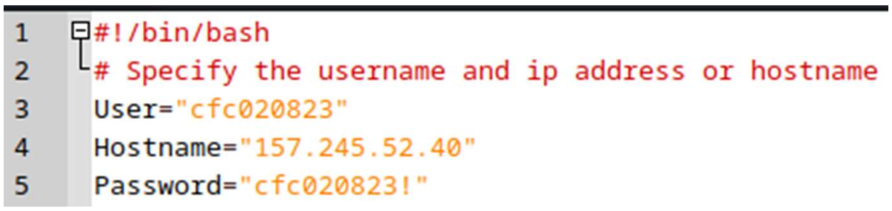
There should be two machines, host and remote server(target/victim)

1. Host machine will bash the script to check if the needed applications such as “sshpass, tor, Geoip” installed on the machine. Will install missing applications.
2. Add auto start nipe
3. Check network connection is anonymous or not, else exit with notifications.
4. Display the spoofed information. (IP and Country)
5. Allow users to type address and store as variable (to be used later)
6. SSH into remote server (using sshpass)
7. Display the Uptime of the remote server, IP address & country (geoiplookup)
8. Use the input from point 5, Whois to get the registrar/information.
9. Use the input from point 5, Nmap to scan for the ports.
10. Save the result to the host machine
11. Create a nr.log

Objective to do create bash file with

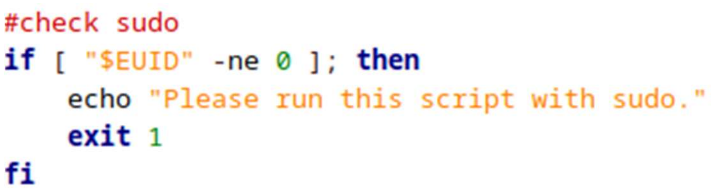
1. Function to check applications – will Install if not installed
2. Function to Ensure nipe.pl is running
3. Function to check if host is anonymous else exit
4. Function to store Input of the ip/domain for whois & nmap
5. Function to Auto ssh by using sshpass
6. Function to run whois & nmap, then save the files on the host machine
7. Function to create nr.log.

Before running nr.sh, ensure that your information is set correctly in the nr.sh in the first six line of the nr.sh



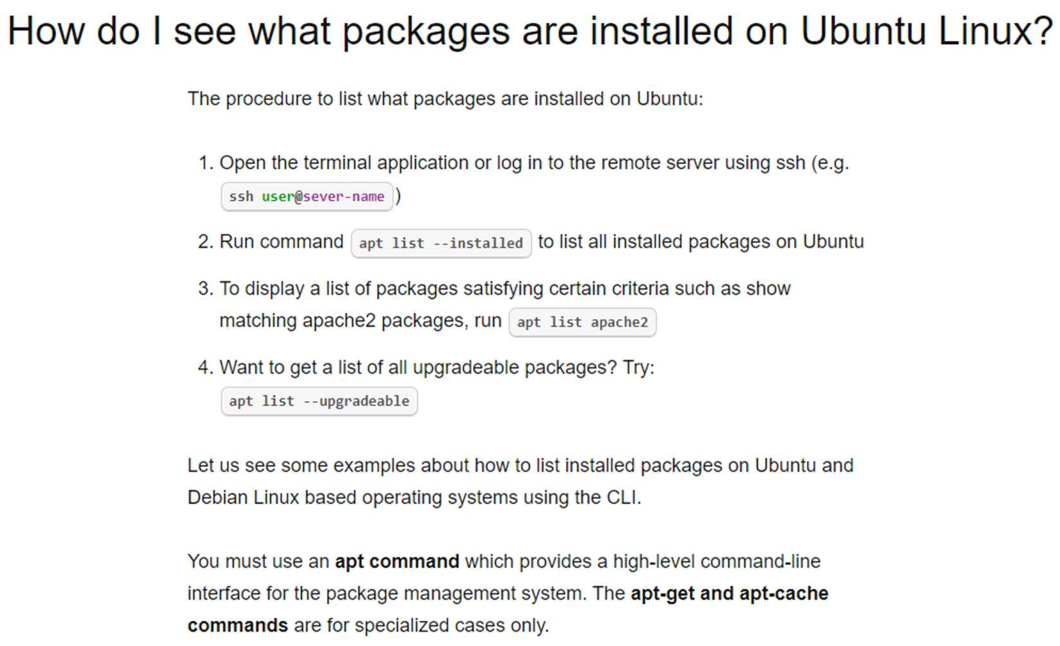
```
#!/bin/bash
# Specify the username and ip address or hostname
User="cfc020823"
Hostname="157.245.52.40"
Password="cfc020823!"
```

Be ROOT or else many error appear, script will not run.



```
#check sudo
if [ "$EUID" -ne 0 ]; then
    echo "Please run this script with sudo."
    exit 1
fi
```

Command to check if applications installed.



How do I see what packages are installed on Ubuntu Linux?

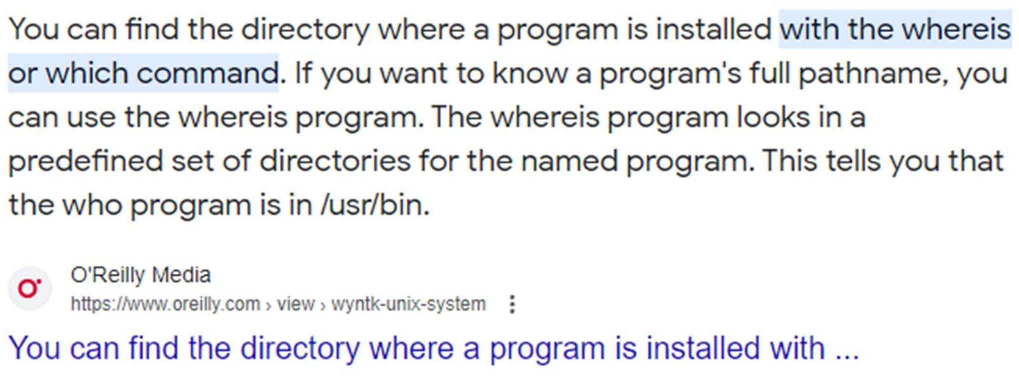
The procedure to list what packages are installed on Ubuntu:

1. Open the terminal application or log in to the remote server using ssh (e.g. `ssh user@sever-name`)
2. Run command `apt list --installed` to list all installed packages on Ubuntu
3. To display a list of packages satisfying certain criteria such as show matching apache2 packages, run `apt list apache2`
4. Want to get a list of all upgradeable packages? Try: `apt list --upgradeable`

Let us see some examples about how to list installed packages on Ubuntu and Debian Linux based operating systems using the CLI.

You must use an **apt command** which provides a high-level command-line interface for the package management system. The **apt-get and apt-cache commands** are for specialized cases only.

https://www.cyberciti.biz/faq/apt-get-list-packages-are-installed-on-ubuntu-linux/
could also use whereis or which

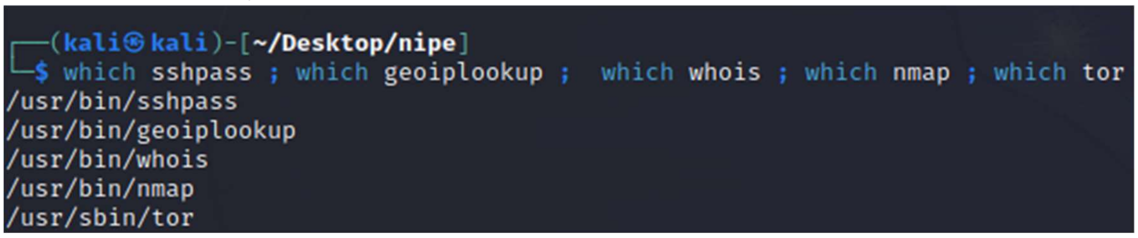


You can find the directory where a program is installed with the whereis or which command. If you want to know a program's full pathname, you can use the whereis program. The whereis program looks in a predefined set of directories for the named program. This tells you that the who program is in /usr/bin.

O'Reilly Media
https://www.oreilly.com › view › wyntk-unix-system

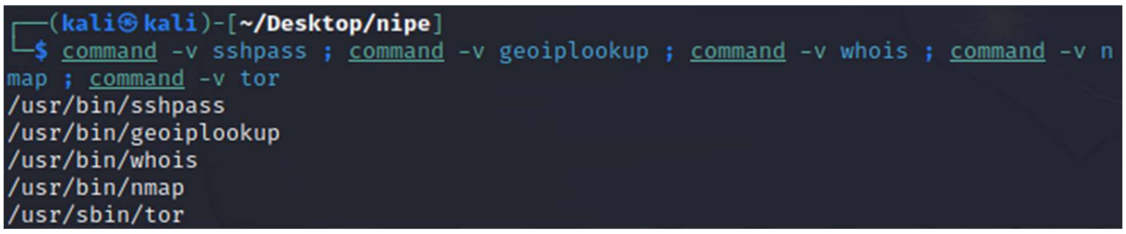
You can find the directory where a program is installed with ...

which will show the path if it is installed.



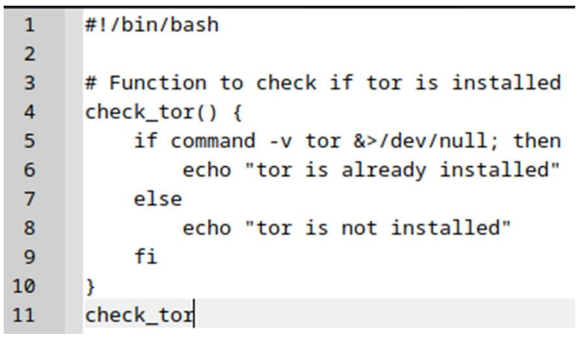
```
┌──(kali㉿kali)-[~/Desktop/nipe]
└─$ which sshpass ; which geoiplookup ;  which whois ; which nmap ; which tor
/usr/bin/sshpass
/usr/bin/geoiplookup
/usr/bin/whois
/usr/bin/nmap
/usr/sbin/tor
```

command -v has the same effect.

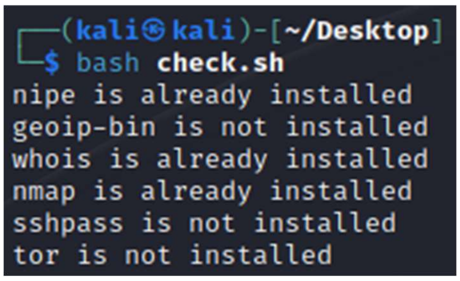


```
┌──(kali㉿kali)-[~/Desktop/nipe]
└─$ command -v sshpass ; command -v geoiplookup ; command -v whois ; command -v n
map ; command -v tor
/usr/bin/sshpass
/usr/bin/geoiplookup
/usr/bin/whois
/usr/bin/nmap
/usr/sbin/tor
```

So I used the command -v to compare if it is installed. (copy and paste the rest)
is like telling it to command -v “tor” then hide text, if installed echo tor is already installed.

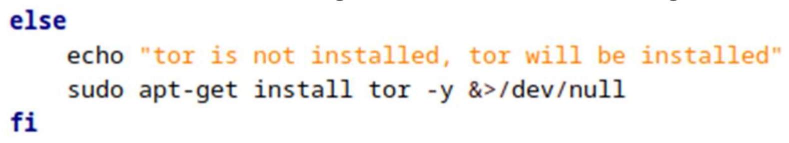


```
#!/bin/bash

# Function to check if tor is installed
check_tor() {
    if command -v tor &>/dev/null; then
        echo "tor is already installed"
    else
        echo "tor is not installed"
    fi
}
check_tor
```



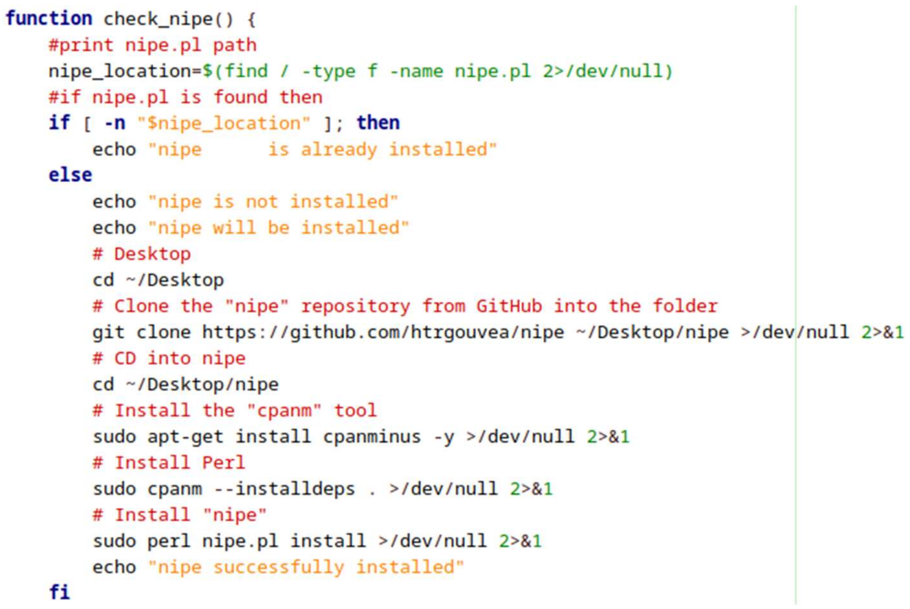
```
┌──(kali㉿kali)-[~/Desktop]
└─$ bash check.sh
nipe is already installed
geoip-bin is not installed
whois is already installed
nmap is already installed
sshpass is not installed
tor is not installed
```

else will install of the missing ones and hide the installing text.



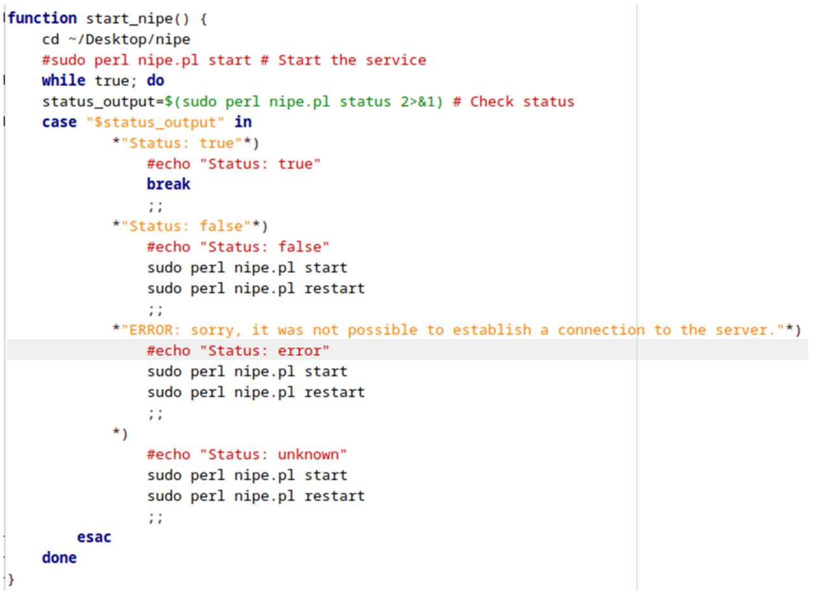
```
else
    echo "tor is not installed, tor will be installed"
    sudo apt-get install tor -y &>/dev/null
fi
```

For nipe.pl, I used “find” to search for nipe.pl. if unable to locate it will install



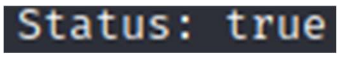
```
function check_nipe() {
    #print nipe.pl path
    nipe_location=$(find / -type f -name nipe.pl 2>/dev/null)
    #if nipe.pl is found then
    if [ -n "$nipe_location" ]; then
        echo "nipe      is already installed"
    else
        echo "nipe is not installed"
        echo "nipe will be installed"
        # Desktop
        cd ~/Desktop
        # Clone the "nipe" repository from GitHub into the folder
        git clone https://github.com/htrgouvea/nipe ~/Desktop/nipe >/dev/null 2>&1
        # CD into nipe
        cd ~/Desktop/nipe
        # Install the "cpanm" tool
        sudo apt-get install cpanminus -y >/dev/null 2>&1
        # Install Perl
        sudo cpanm --installdeps . >/dev/null 2>&1
        # Install "nipe"
        sudo perl nipe.pl install >/dev/null 2>&1
        echo "nipe successfully installed"
    fi
```

Making nipe run auto (auto start nipe)

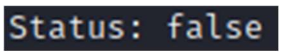


```
function start_nipe() {
    cd ~/Desktop/nipe
    #sudo perl nipe.pl start # Start the service
    while true; do
    status_output=$(sudo perl nipe.pl status 2>&1) # Check status
    case "$status_output" in
            *"Status: true"*)
                #echo "Status: true"
                break
                ;;
            *"Status: false"*)
                #echo "Status: false"
                sudo perl nipe.pl start
                sudo perl nipe.pl restart
                ;;
            *"ERROR: sorry, it was not possible to establish a connection to the server."*)
                #echo "Status: error"
                sudo perl nipe.pl start
                sudo perl nipe.pl restart
                ;;
            *)
                #echo "Status: unknown"
                sudo perl nipe.pl start
                sudo perl nipe.pl restart
                ;;
        esac
    done
}
```

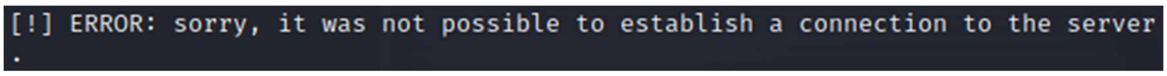
To check the status, run sudo perl nipe.pl status and status_output will store the result
there will be 4 result
true, false, ERROR or unknown



```
Status: true
```



```
Status: false
```



```
[!] ERROR: sorry, it was not possible to establish a connection to the server
.
```

in while loop, if true it will break out of the loop (will comment out the echo)

If false or error, it will start, restart nipe.pl then run start_nipe to check again. (will comment out the echo)

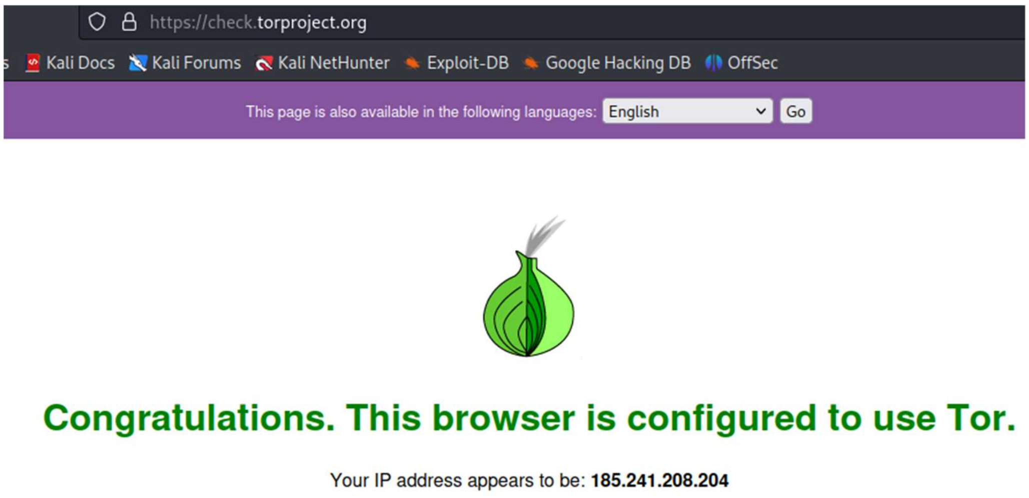
in other words, other then true, it will just loop.

Checking if anonymous.

Use curl on torproject.org and grep for the keyword Congratulation.

```
#congratulation = on tor network
result=$(curl -s https://check.torproject.org/ | grep -o "Congratulations")
if [[ -n "$result" ]]; then
```

The spoofed IP address also appears on the page.


https://check.torproject.org
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
This page is also available in the following languages:
English
Go

Congratulations. This browser is configured to use Tor.

Your IP address appears to be: **185.241.208.204**

Grep for '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' and there is 2 set of IP addresses.

```
spoofed_ip=$(curl -s https://check.torproject.org/ | grep -E '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}')
echo "$spoofed ip"
```

```
   <p>Your IP address appears to be:  <strong>185.220.101.18</strong></p>
       Please refer to the <a href="https://www.torproject.org/">Tor website</
 a> for further information about using Tor safely.  You are now free to brows
 e the Internet anonymously. For more information about this exit relay, see:
 <a href="https://metrics.torproject.org/rs.html#search/185.220.101.18">Relay
 Search</a>.
```

I choose the first because it contains the "Your IP address appears to be : "

```
spoofed_ip=$(curl -s https://check.torproject.org/ | grep -E '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' | head -n 1)
echo "$spoofed_ip"
```

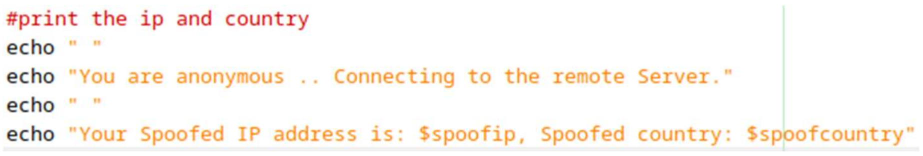
```
   <p>Your IP address appears to be:  <strong>162.247.74.216</strong></p>
 You are anonymous .. Connecting to the remote Server.
```

Changed from "grep -E" to "grep -oE" and captured the address.

```
spoofip=$(curl -s https://check.torproject.org/ | grep -oE '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}'| head -n 1)
echo "$spoofed_ip"
```

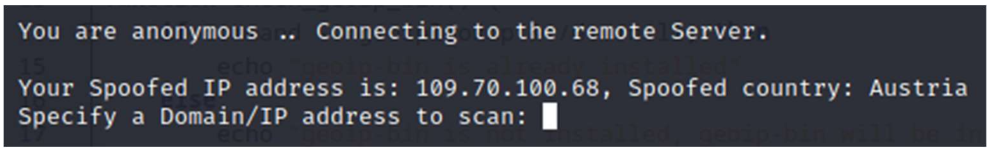
Use the ip found and use geoiplookup

```
spoofcountry=$(geoiplookup "$spoofip" | awk '{print $5, $6 ,$7}')
```

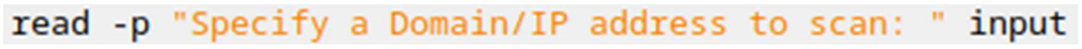


```
#print the ip and country
echo " "
echo "You are anonymous .. Connecting to the remote Server."
echo " "
echo "Your Spoofed IP address is: $spoofip, Spoofed country: $spoofcountry"
```

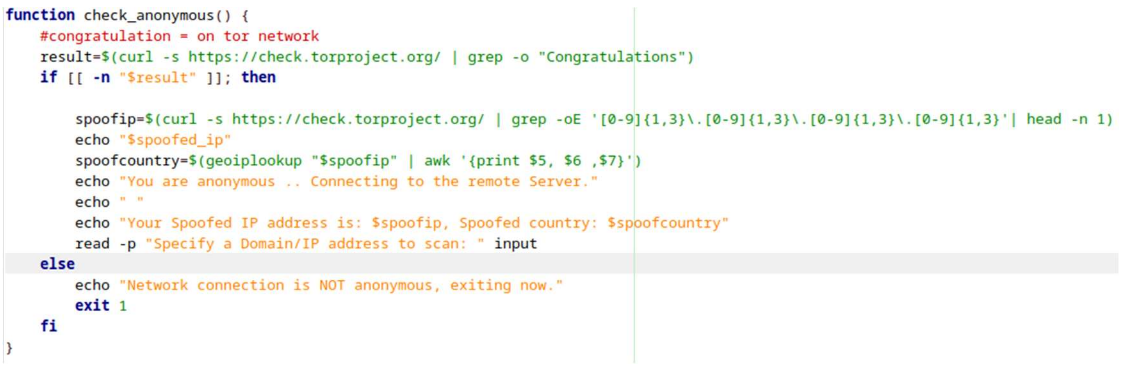
This will allow user to type in the IP address or domain and will be stored as input to be used at the remote server.



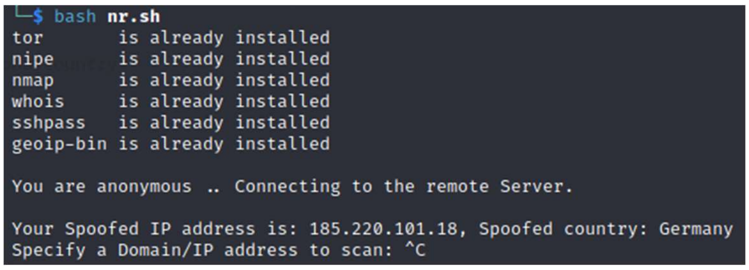
```
You are anonymous .. Connecting to the remote Server.

Your Spoofed IP address is: 109.70.100.68, Spoofed country: Austria
Specify a Domain/IP address to scan: 
```



```
read -p "Specify a Domain/IP address to scan: " input
```

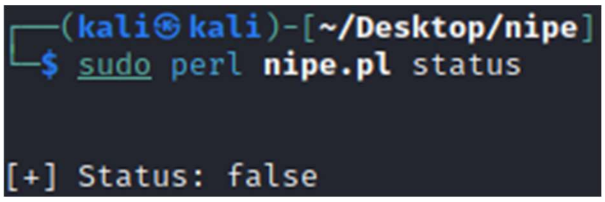


```
function check_anonymous() {
    #congratulation = on tor network
    result=$(curl -s https://check.torproject.org/ | grep -o "Congratulations")
    if [[ -n "$result" ]]; then

        spoofip=$(curl -s https://check.torproject.org/ | grep -oE '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}'| head -n 1)
        echo "$spoofed_ip"
        spoofcountry=$(geoiplookup "$spoofip" | awk '{print $5, $6 ,$7}')
        echo "You are anonymous .. Connecting to the remote Server."
        echo " "
        echo "Your Spoofed IP address is: $spoofip, Spoofed country: $spoofcountry"
        read -p "Specify a Domain/IP address to scan: " input
    else
        echo "Network connection is NOT anonymous, exiting now."
        exit 1
    fi
}
```



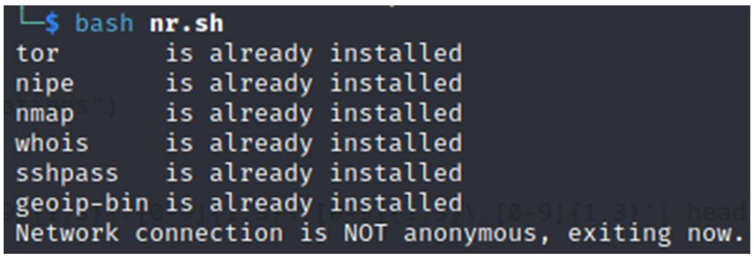
```
└─$ bash nr.sh
tor       is already installed
nipe      is already installed
nmap      is already installed
whois     is already installed
sshpass   is already installed
geoip-bin is already installed

You are anonymous .. Connecting to the remote Server.

Your Spoofed IP address is: 185.220.101.18, Spoofed country: Germany
Specify a Domain/IP address to scan: ^C
```

I used “sudo perl nipple.pl stop” to test if the nr.sh is able to detect or not.



```
┌──(kali㉿kali)-[~/Desktop/nipe]
└─$ sudo perl nipe.pl status


[+] Status: false
```

It will not proceed if status is false.

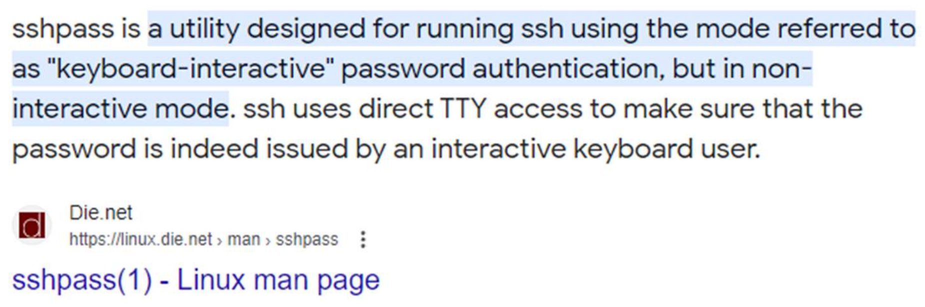


```
└─$ bash nr.sh
tor       is already installed
nipe      is already installed
nmap      is already installed
whois     is already installed
sshpass   is already installed
geoip-bin is already installed
Network connection is NOT anonymous, exiting now.
```

Sometime needed to “sudo perl nipple.pl restart” a couple of time

Network connection is NOT anonymous, exiting now.

What is sshpass?

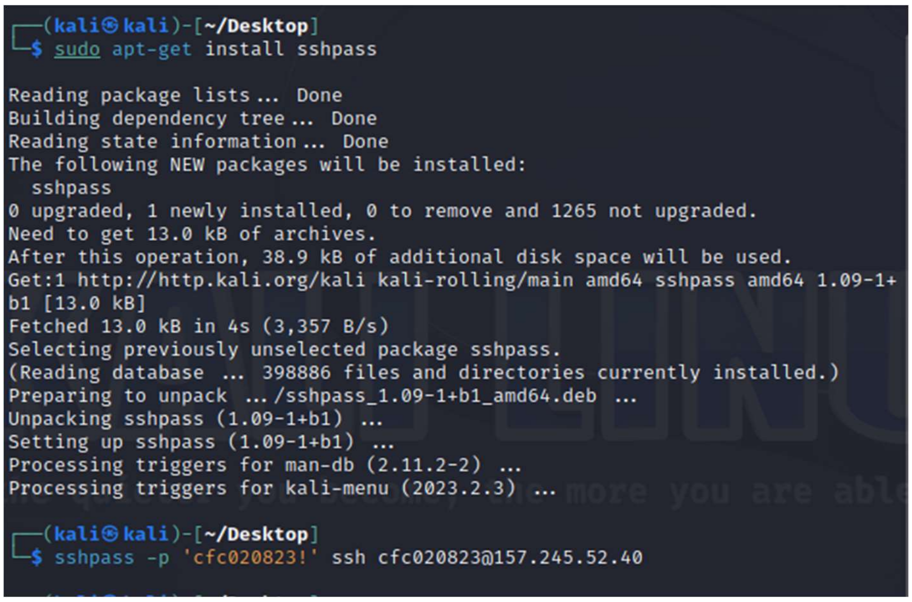


sshpass is a utility designed for running ssh using the mode referred to as "keyboard-interactive" password authentication, but in non-interactive mode. ssh uses direct TTY access to make sure that the password is indeed issued by an interactive keyboard user.

Die.net
https://linux.die.net › man › sshpass

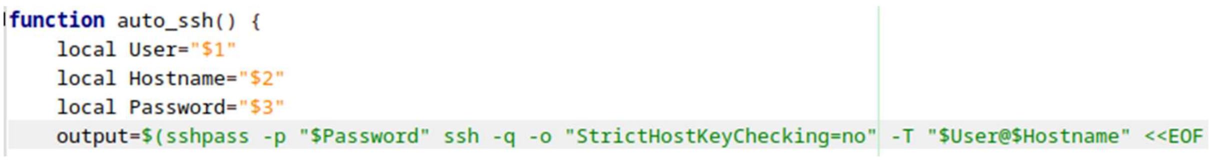
sshpass(1) - Linux man page

Installing sshpass



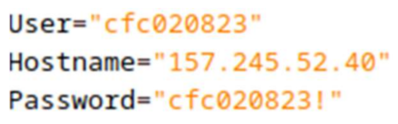
```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo apt-get install sshpass

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  sshpass
0 upgraded, 1 newly installed, 0 to remove and 1265 not upgraded.
Need to get 13.0 kB of archives.
After this operation, 38.9 kB of additional disk space will be used.
Get:1 http://http.kali.org/kali kali-rolling/main amd64 sshpass amd64 1.09-1+
b1 [13.0 kB]
Fetched 13.0 kB in 4s (3,357 B/s)
Selecting previously unselected package sshpass.
(Reading database ... 398886 files and directories currently installed.)
Preparing to unpack .../sshpass_1.09-1+b1_amd64.deb ...
Unpacking sshpass (1.09-1+b1) ...
Setting up sshpass (1.09-1+b1) ...
Processing triggers for man-db (2.11.2-2) ...
Processing triggers for kali-menu (2023.2.3) ...

┌──(kali㉿kali)-[~/Desktop]
└─$ sshpass -p 'cfc020823!' ssh cfc020823@157.245.52.40
```

sshpass -p 'Password' ssh Username@serveripaddress

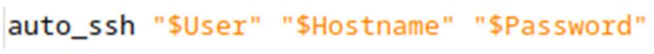


```
function auto_ssh() {
    local User="$1"
    local Hostname="$2"
    local Password="$3"
    output=$(sshpass -p "$Password" ssh -q -o "StrictHostKeyChecking=no" -T "$User@$Hostname" <<EOF
```

Set the User, Hostname and password.



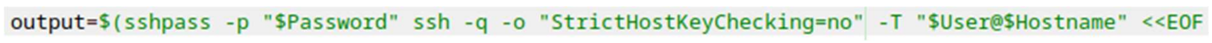
```
User="cfc020823"
Hostname="157.245.52.40"
Password="cfc020823!"
```



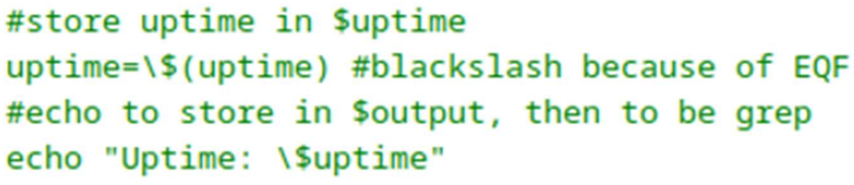
```
auto_ssh "$User" "$Hostname" "$Password"
```

StrictHostKeyChecking must be set to no.

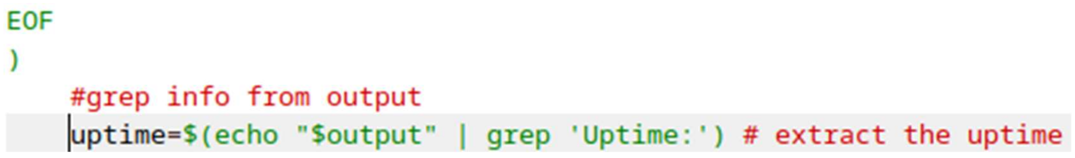
$output will capture the all the information in remote server.


```
output=$(sshpass -p "$Password" ssh -q -o "StrictHostKeyChecking=no" -T "$User@$Hostname" <<EOF
```

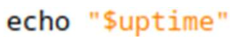
Run uptime at the remote server, then store $uptime


```
#store uptime in $uptime
uptime=\$(uptime) #blackslash because of EQF
#echo to store in $output, then to be grep
echo "Uptime: \$uptime"
```

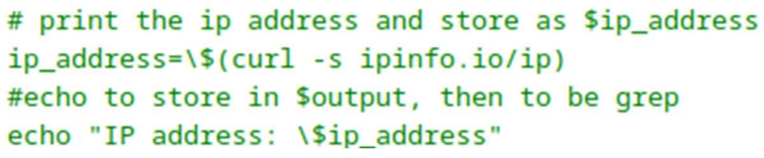
Storing the result to $uptime


```
EOF
)
    #grep info from output
    uptime=$(echo "$output" | grep 'Uptime:') # extract the uptime
```

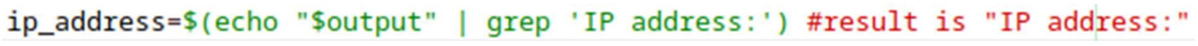
Then echo it to the host screen.


```
echo "$uptime"
```

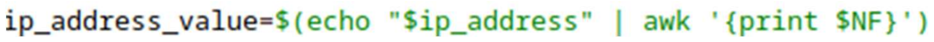
Use ipinfo.io/ip to check the obtain the IP address of remote server & echo IP address: \$ipaddress so it could be grep


```
# print the ip address and store as $ip_address
ip_address=\$(curl -s ipinfo.io/ip)
#echo to store in $output, then to be grep
echo "IP address: \$ip_address"
```

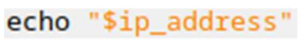
Extract “IP address: \$ip_address” and store it as $ip_address from the output


```
ip_address=$(echo "$output" | grep 'IP address:') #result is "IP address:"
```

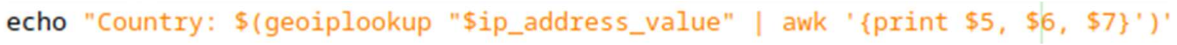
Get IP address and exclude the “IP address: ”


```
ip_address_value=$(echo "$ip_address" | awk '{print $NF}')
```

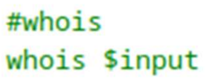
Will echo the result from before


```
echo "$ip_address"
```

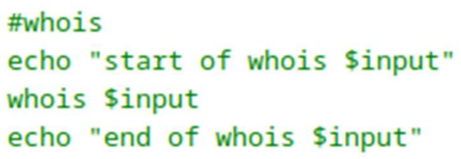
Will geriplookup with the ip address, then awk the behide (country name)


```
echo "Country: $(geoiplookup "$ip_address_value" | awk '{print $5, $6, $7}')'
```

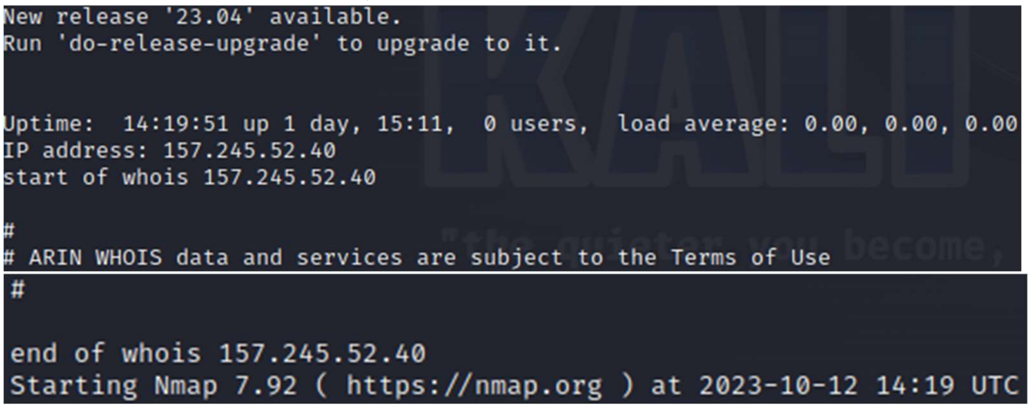
Adding whois


```
#whois
whois $input
```

So I add marker to know where to grep from, such as start point and end point.

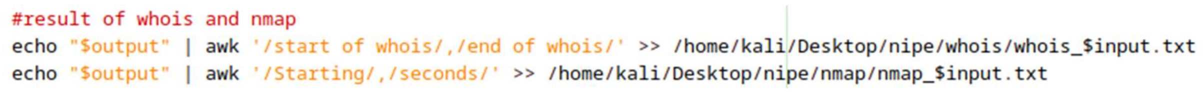

```
#whois
echo "start of whois $input"
whois $input
echo "end of whois $input"
```

Run nr.sh and checked the output



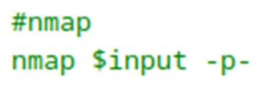
```
New release '23.04' available.
Run 'do-release-upgrade' to upgrade to it.


Uptime:  14:19:51 up 1 day, 15:11,  0 users,  load average: 0.00, 0.00, 0.00
IP address: 157.245.52.40
start of whois 157.245.52.40

#
# ARIN WHOIS data and services are subject to the Terms of Use
#

end of whois 157.245.52.40
Starting Nmap 7.92 ( https://nmap.org ) at 2023-10-12 14:19 UTC
```

Extract information from $output using awk from the start and end



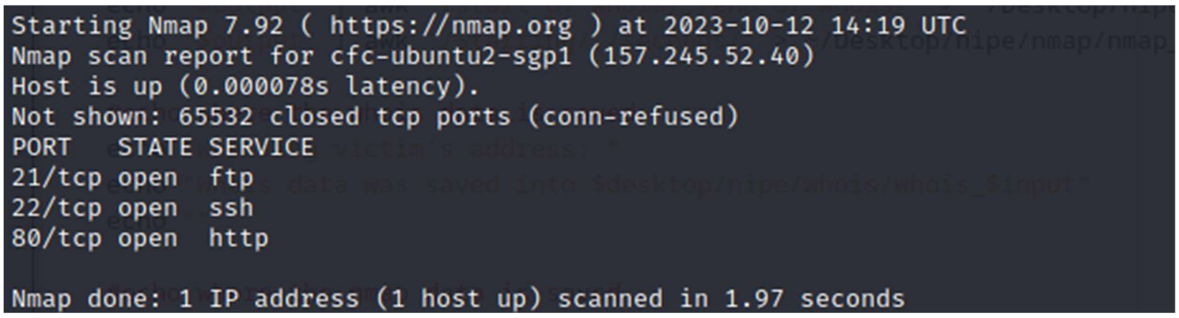
```
#result of whois and nmap
echo "$output" | awk '/start of whois/,/end of whois/' >> /home/kali/Desktop/nipe/whois/whois_$input.txt
echo "$output" | awk '/Starting/,/seconds/' >> /home/kali/Desktop/nipe/nmap/nmap_$input.txt
```

Nmap the input with the flag -p-



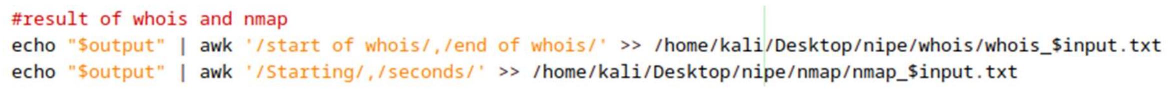
```
#nmap
nmap $input -p-
```

So the nmap scan result is also in the $output.



```
Starting Nmap 7.92 ( https://nmap.org ) at 2023-10-12 14:19 UTC
Nmap scan report for cfc-ubuntu2-sgp1 (157.245.52.40)
Host is up (0.000078s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 1.97 seconds
```

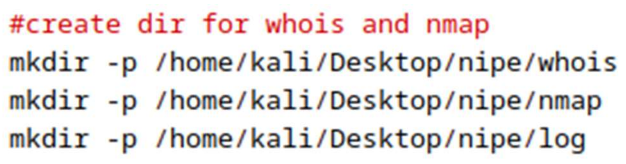
Extract information from $output using awk from Starting to seconds



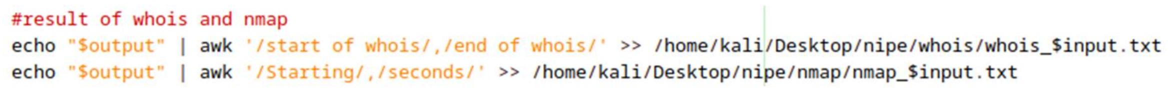
```
#result of whois and nmap
echo "$output" | awk '/start of whois/,/end of whois/' >> /home/kali/Desktop/nipe/whois/whois_$input.txt
echo "$output" | awk '/Starting/,/seconds/' >> /home/kali/Desktop/nipe/nmap/nmap_$input.txt
```

Saving the datas files on the host machine

Create the dir for the data

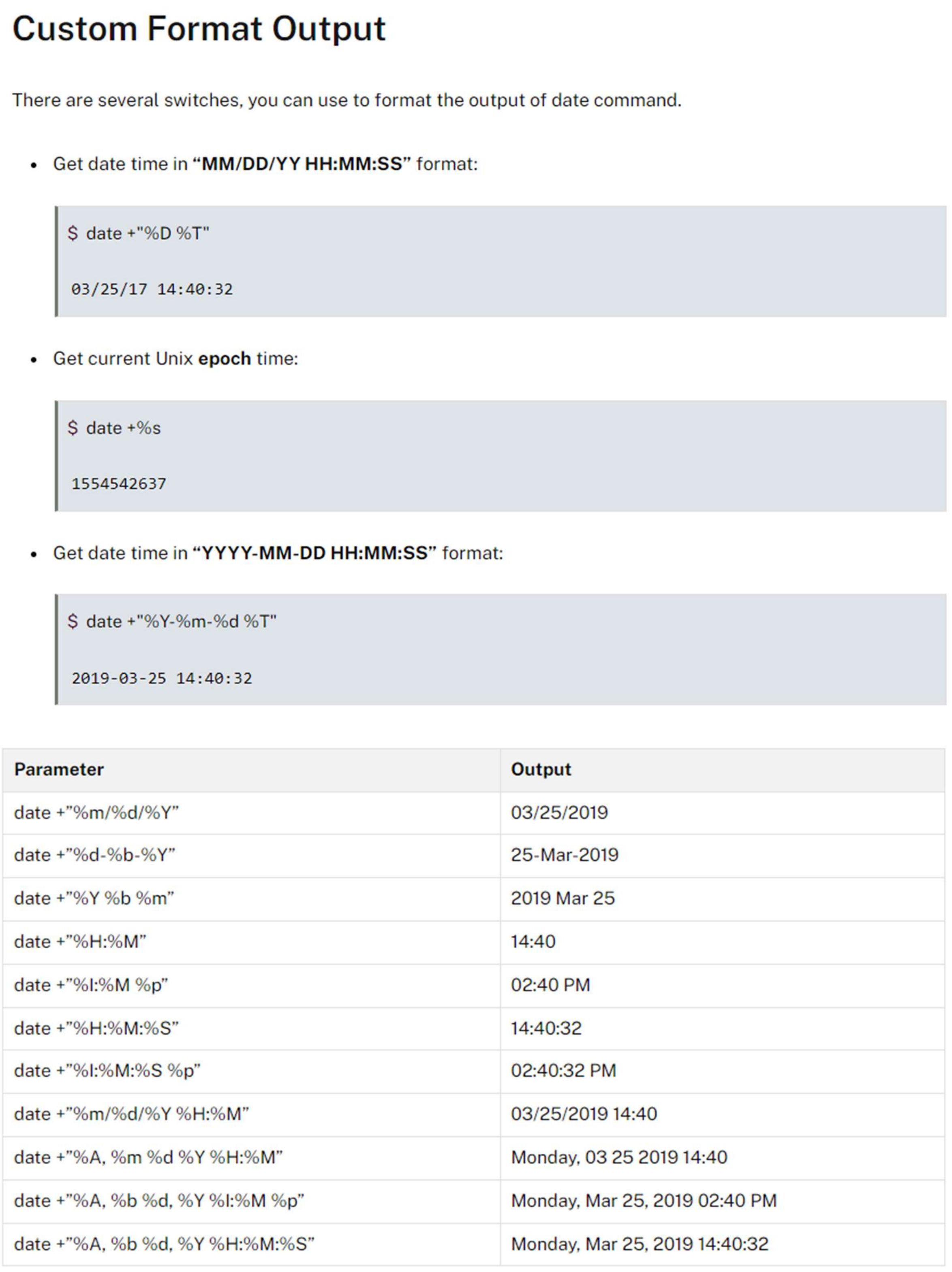


```
#create dir for whois and nmap
mkdir -p /home/kali/Desktop/nipe/whois
mkdir -p /home/kali/Desktop/nipe/nmap
mkdir -p /home/kali/Desktop/nipe/log
```



```
#result of whois and nmap
echo "$output" | awk '/start of whois/,/end of whois/' >> /home/kali/Desktop/nipe/whois/whois_$input.txt
echo "$output" | awk '/Starting/,/seconds/' >> /home/kali/Desktop/nipe/nmap/nmap_$input.txt
```

Trying to add dates into the nr.log, google for advice and found:



Custom Format Output

There are several switches, you can use to format the output of date command.

- Get date time in **"MM/DD/YY HH:MM:SS"** format:

```
$ date +"%D %T"

03/25/17 14:40:32
```

- Get current Unix **epoch** time:

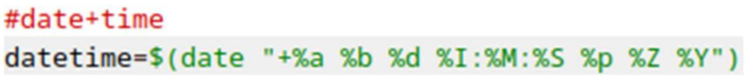
```
$ date +%s

1554542637
```

- Get date time in **"YYYY-MM-DD HH:MM:SS"** format:

```
$ date +"%Y-%m-%d %T"

2019-03-25 14:40:32
```

Parameter	**Output**
date +"%m/%d/%Y"	03/25/2019
date +"%d-%b-%Y"	25-Mar-2019
date +"%Y %b %m"	2019 Mar 25
date +"%H:%M"	14:40
date +"%I:%M %p"	02:40 PM
date +"%H:%M:%S"	14:40:32
date +"%I:%M:%S %p"	02:40:32 PM
date +"%m/%d/%Y %H:%M"	03/25/2019 14:40
date +"%A, %m %d %Y %H:%M"	Monday, 03 25 2019 14:40
date +"%A, %b %d, %Y %I:%M %p"	Monday, Mar 25, 2019 02:40 PM
date +"%A, %b %d, %Y %H:%M:%S"	Monday, Mar 25, 2019 14:40:32

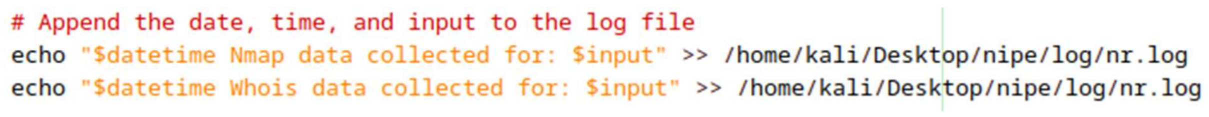
https://tecadmin.net/get-current-date-and-time-in-bash/



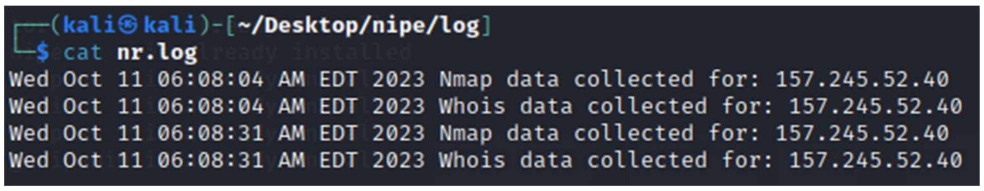
```
#date+time
datetime=$(date "+%a %b %d %I:%M:%S %p %Z %Y")
```

Create the logfile and point at the dir created.

Use >> to apend information to the nr.log

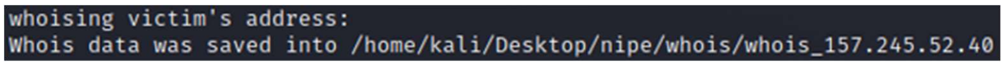


```
# Append the date, time, and input to the log file
echo "$datetime Nmap data collected for: $input" >> /home/kali/Desktop/nipe/log/nr.log
echo "$datetime Whois data collected for: $input" >> /home/kali/Desktop/nipe/log/nr.log
```

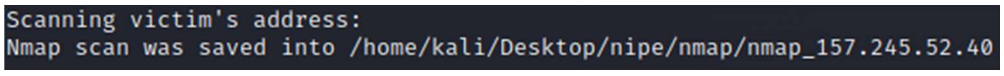


```
┌──(kali㉿kali)-[~/Desktop/nipe/log]
└─$ cat nr.log
Wed Oct 11 06:08:04 AM EDT 2023 Nmap data collected for: 157.245.52.40
Wed Oct 11 06:08:04 AM EDT 2023 Whois data collected for: 157.245.52.40
Wed Oct 11 06:08:31 AM EDT 2023 Nmap data collected for: 157.245.52.40
Wed Oct 11 06:08:31 AM EDT 2023 Whois data collected for: 157.245.52.40
```

Manage to append the information to nr.log



```
whoising victim's address:
Whois data was saved into /home/kali/Desktop/nipe/whois/whois_157.245.52.40
```



```
Scanning victim's address:
Nmap scan was saved into /home/kali/Desktop/nipe/nmap/nmap_157.245.52.40
```

Challenges faced during the project

1) When trying to check if nipe.pl is installed
    1) unable to use which, command -v
    2) unable to use dpkg, type, apt list
    3) installing nipe.pl in /usr/bin/ or symtemctl
2) Checking if anonymous
    1) netstat -tuln | grep 9050 – not accurate
    2) sudo perl nipe.pl status – trying to not sudo
    3) curl ipinfo.io/ip or curl ifconfig.me unable to get IP address
3) Sshpass
    1) Need a lot of research on sshpass, output -p and -e took awhile to understand
    2) StrictHostKeyChecking=no
    3) Using local and backslash
4) SUDO
    1) Have to rename everything beacause ~ becomes ROOT (added check sudo)
    2) Changes to nipe install from ~/Desktop to /home/kali
    3) Renaming dir for whois, nmap