Project

Network research

Israel Gatterer

Objective

Create a script that communicates with a remote server and executes tasks anonymously.

1. Install relevant applications on the local computer.

2. Check if the connection is from your origin country.

3. Once the connection is anonymous, communicate via SSH and execute nmap scans and whois queries.

Functions - function is a technique for grouping reusable bits of code under one name for later use, and comes with two benefits:
1. A function is read directly into the shell's memory and stored for later use. Since computer memory is not an issue nowadays, using functions is faster than repeating code.

2. Functions help organize long shell scripts into modular and reusable code blocks. The chunks are easier to develop and maintain.

The commands between the curly braces **{ <commands> }** are called the function's body. The body can contain any number of declarations, variables, loops, or conditional statements.

- One way to write a bash function is using the reserved word '**function**' (see below colored blue).

- Using descriptive names (see below: INSTL, ANON, VPS) for functions aren't necrssary for testing commands but help in settings where other developers look at the code.

Function 1 - Installing relevant applications

1. Installing nipe

- Protecting our user in brute force attack. Once the nipe service is started, the ip address representing the vm - within the tor network/Internet - is not associated with the wan network adapter of the customer's router but instead is identified by the 3'rd tor node within the tor network that serves as an exit.

using the following command, we must clone this repository from GitHub: git clone https://github.com/htrgouvea/nipe

if statements in Bash are used to execute code based on a certain condition.

- **if** - The if statement starts with the if keyword followed by the conditional expression and the then keyword. The statement ends with the fi keyword.

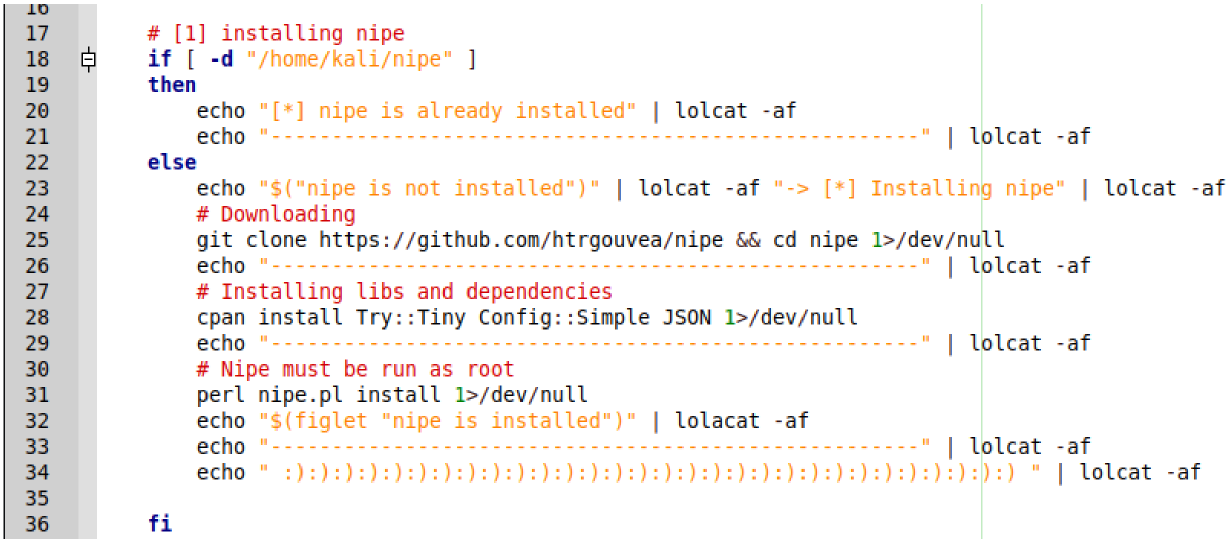
If the performs of a particular set of actions if a statement is true and other statement is false. if the test-command evaluates to True, the statements gets executed. If test-command returns False, nothing happens; the statements get ignored.

To do this, we use the '**else**' statement, which has the following screenshot.

(We can add more arguments to the statements with the help of 'elif' statement).

- Using 'echo' I can print a variable.
- Foloowing to the 'echo' - we are using single quote (') to use the veriable. But when I'm using a variable with something other then I need to use double quote (").

- "**#**" - note. Keeps the following line out of the output of the bash script.



```
16
17      # [1] installing nipe
18      if [ -d "/home/kali/nipe" ]
19      then
20          echo "[*] nipe is already installed" | lolcat -af
21          echo "------------------------------------------------------" | lolcat -af
22      else
23          echo "$("nipe is not installed")" | lolcat -af "-> [*] Installing nipe" | lolcat -af
24          # Downloading
25          git clone https://github.com/htrgouvea/nipe && cd nipe 1>/dev/null
26          echo "------------------------------------------------------" | lolcat -af
27          # Installing libs and dependencies
28          cpan install Try::Tiny Config::Simple JSON 1>/dev/null
29          echo "------------------------------------------------------" | lolcat -af
30          # Nipe must be run as root
31          perl nipe.pl install 1>/dev/null
32          echo "$(figlet "nipe is installed")" | lolacat -af
33          echo "------------------------------------------------------" | lolcat -af
34          echo " :):):):):):):):):):):):):):):):):):):):):):):):):):):):):) " | lolcat -af
35
36      fi
```

- Line 18: '-d' = Directory check. If it does exist in the following path (Can use also '-f' for example, for “file”).

- Credit for installing commands github - https://github.com/htrgouvea/nipe

2. Installing sshpass

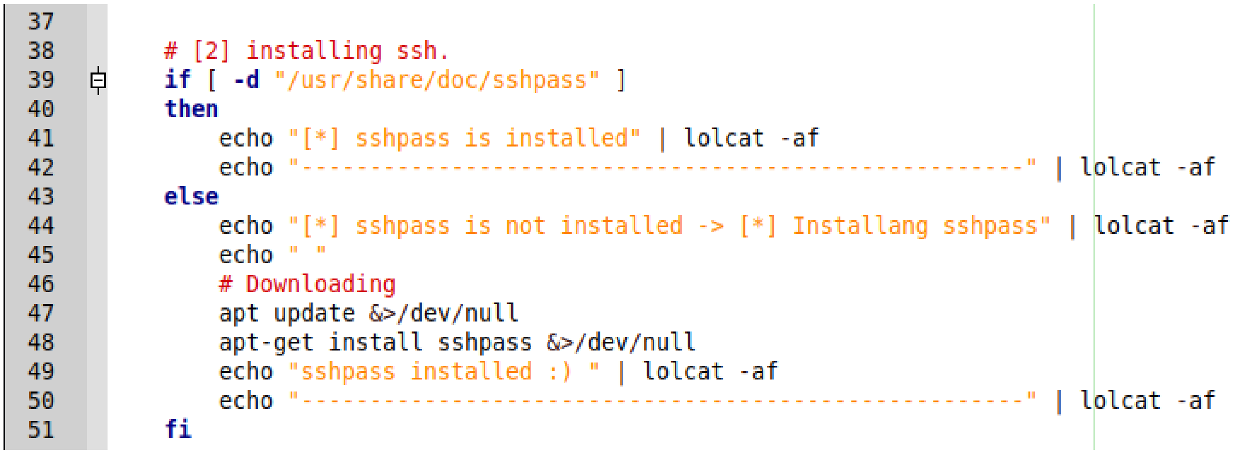
- Downloading sshpass to avoid in ssh command the fingerprint / password request.

SSH’s (secure shell) most common authentication mode is called “interactive keyboard password authentication”, so called both because it is typically done via keyboard, and because openssh takes active measures to make sure that the password is, indeed, typed interactively by the keyboard. Sometimes, however, it is necessary to fool ssh into accepting an interactive password non-interactively. This is where sshpass comes in.

Installing sshpass Using apt-get - Update apt database with apt-get using the following command: sudo apt-get update.

After updating apt database, We can install sshpass using apt-get by running the following command:

sudo apt-get install sshpass



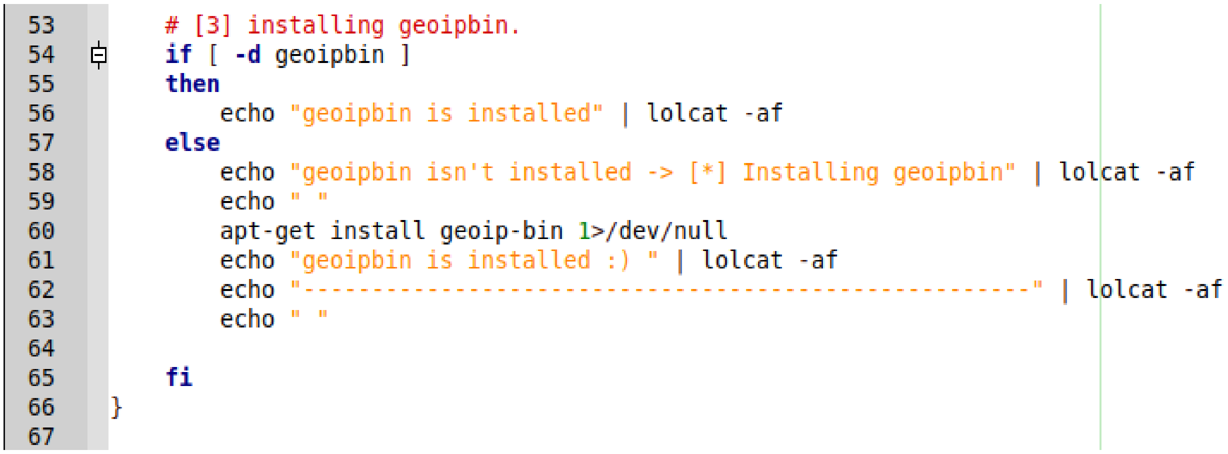
```
    # [2] installing ssh.
    if [ -d "/usr/share/doc/sshpass" ]
    then
        echo "[*] sshpass is installed" | lolcat -af
        echo "----------------------------------------------------" | lolcat -af
    else
        echo "[*] sshpass is not installed -> [*] Installang sshpass" | lolcat -af
        echo " "
        # Downloading
        apt update &>/dev/null
        apt-get install sshpass &>/dev/null
        echo "sshpass installed :) " | lolcat -af
        echo "----------------------------------------------------" | lolcat -af
    fi
```

3. <u>Installing geoipbin</u>

- GeoIP is a C library that enables the user to find the country that any IP address or hostname originates from. It uses a file based database.

This database simply contains IP blocks as keys, and countries as values and it should be more complete and accurate than using reverse DNS lookups.

This package contains the command line utilities to resolve the IP numbers using the GeoIP library. Using geoipbin application because “whois” database doesn’t recognize all Tor commands in nipe mode - that’s why it doesn’t works all the time.

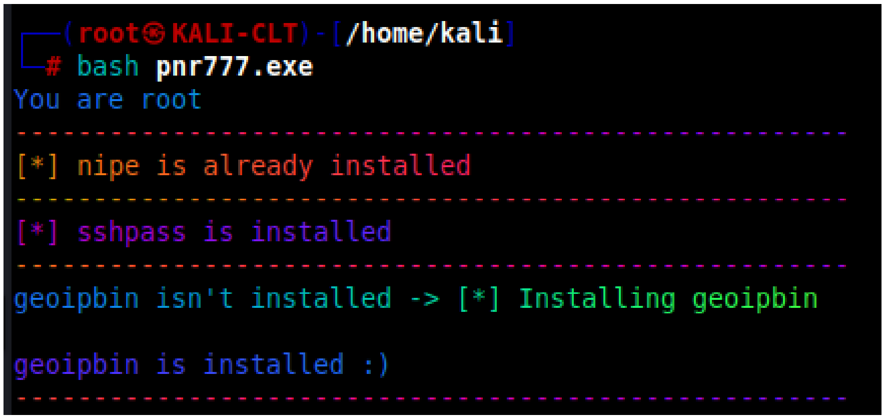


```
    # [3] installing geoipbin.
    if [ -d geoipbin ]
    then
        echo "geoipbin is installed" | lolcat -af
    else
        echo "geoipbin isn't installed -> [*] Installing geoipbin" | lolcat -af
        echo " "
        apt-get install geoip-bin 1>/dev/null
        echo "geoipbin is installed :) " | lolcat -af
        echo "-----------------------------------------------------" | lolcat -af
        echo " "

    fi
}

```

* S.P -lolcat command

Lolcat is an utility for Linux, BSD and OSX which concatenates like similar to cat command and adds rainbow coloring to it. Lolcat is primarily used for rainbow coloring of text in Linux Terminal (No obligatory. I used it only for this specific fecorative work)

Output displayed:



```
┌──(root㉿KALI-CLT)-[/home/kali]
└─# bash pnr777.exe
You are root
-------------------------------------------------------
[*] nipe is already installed
-------------------------------------------------------
[*] sshpass is installed
-------------------------------------------------------
geoipbin isn't installed -> [*] Installing geoipbin

geoipbin is installed :)
-------------------------------------------------------
```

- Screenshot display that the applications installed successfully.

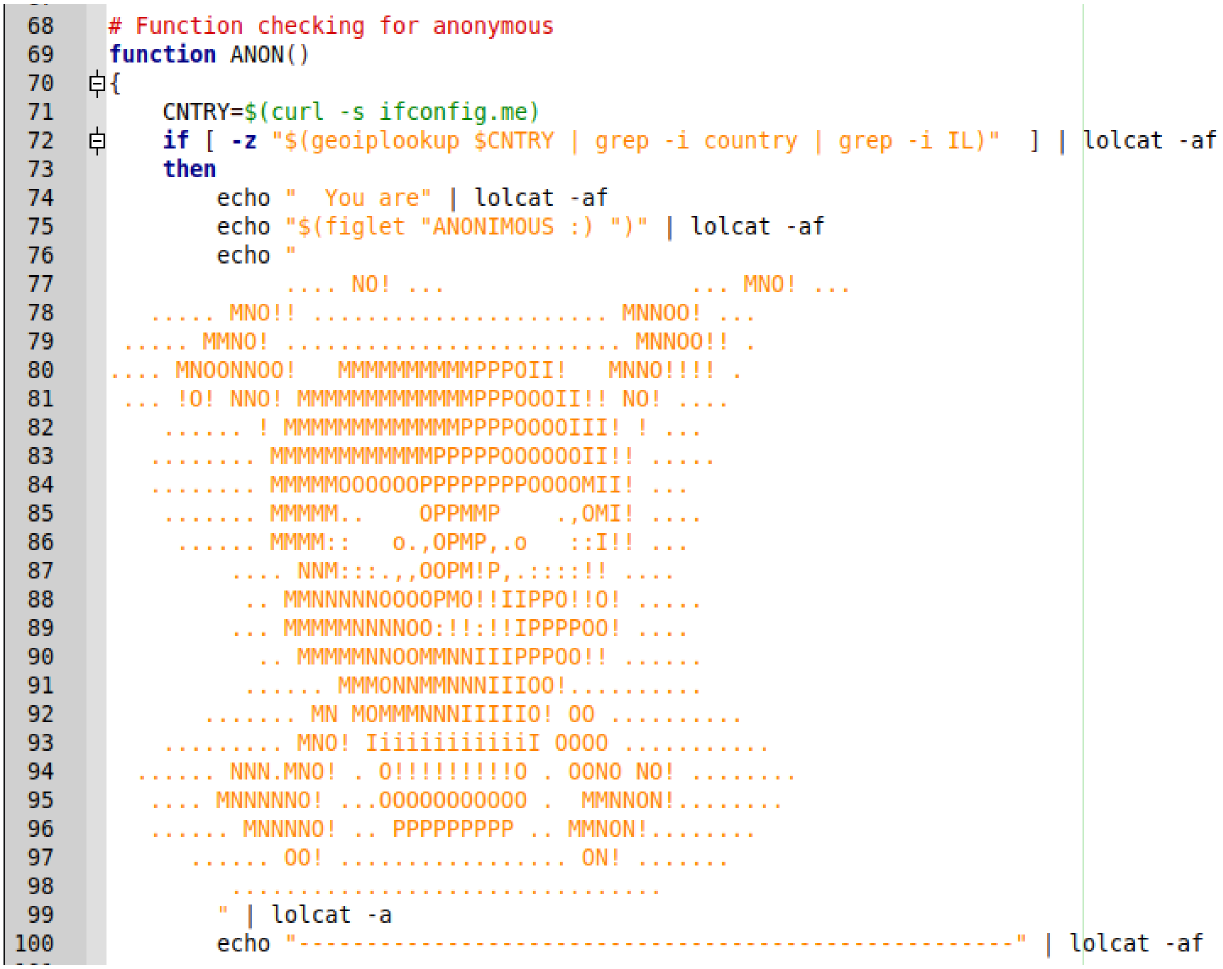
Function 2 - Starting anonymous

In order to start connection with remote server and executing automatic tasks the master must be hiding using nipe (look at Installing nipe folder). Nipe must be run as root.

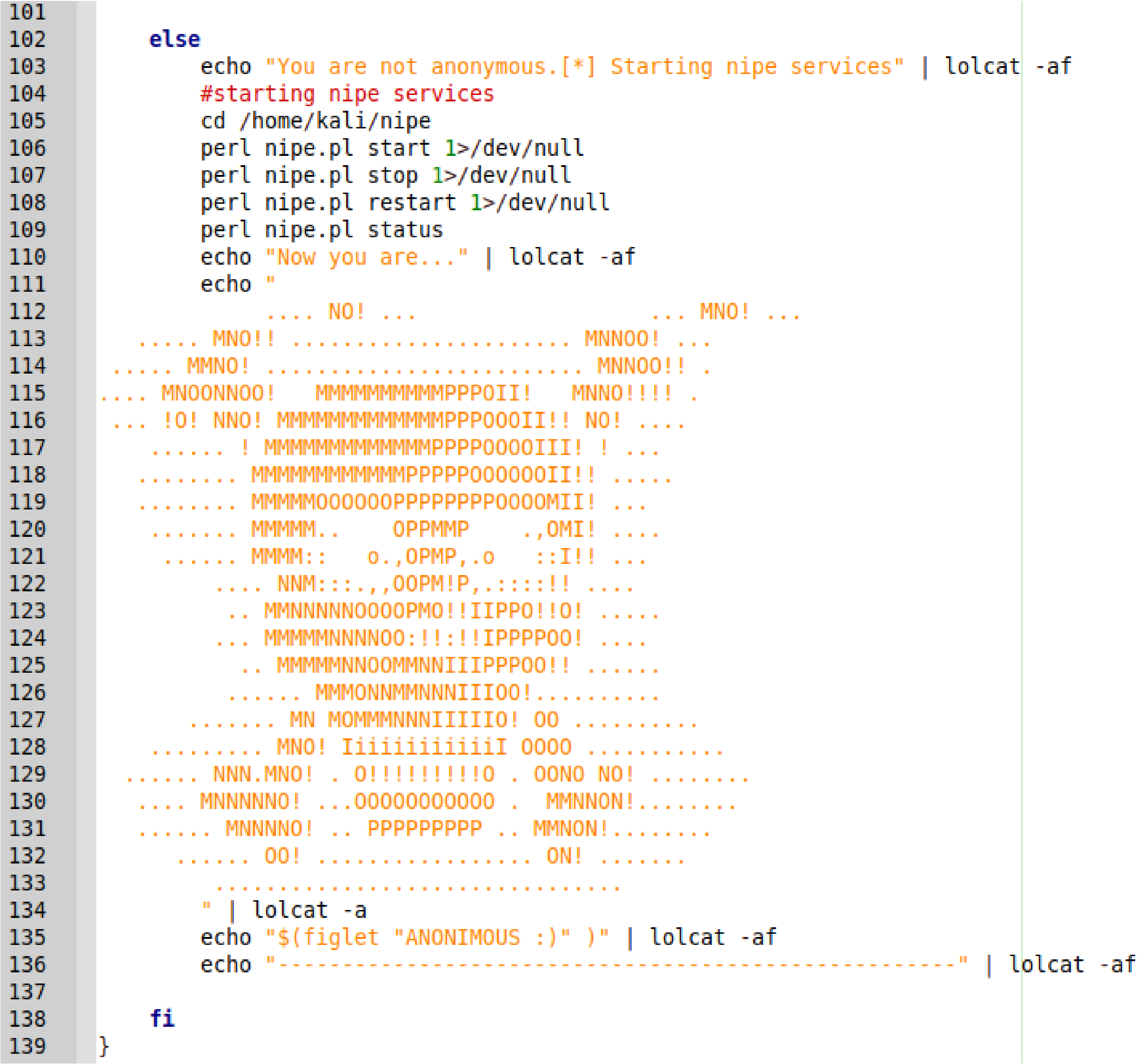
- The following script displays examples of terminal variables. Terminal variable content always start with $(). This syntax $() is used to execute the command and the result of executing the command will become the variable content.

- To use/reference properly a variable we precede the variable with '**$**' character. The bash replaces the variable name with it's value before executing the command (see screenshot bellow CNTRY=$).
- "()" Allows to take the outpot of a command (who's going to be printed to the screen) and have it saved as a value of a variable by placing the variable name in parentheses "()"' preceded by a "$" character.
- CNTRY=$(curl -s ifconfig.me) # The result of executing curl -s ifconfig.me command will display the public ip

- Here I assigned the output of the 'curl -s' command the CNTRY variable. Then I displayed it's value by echo. In the following screenshot we can see the output of the above command.

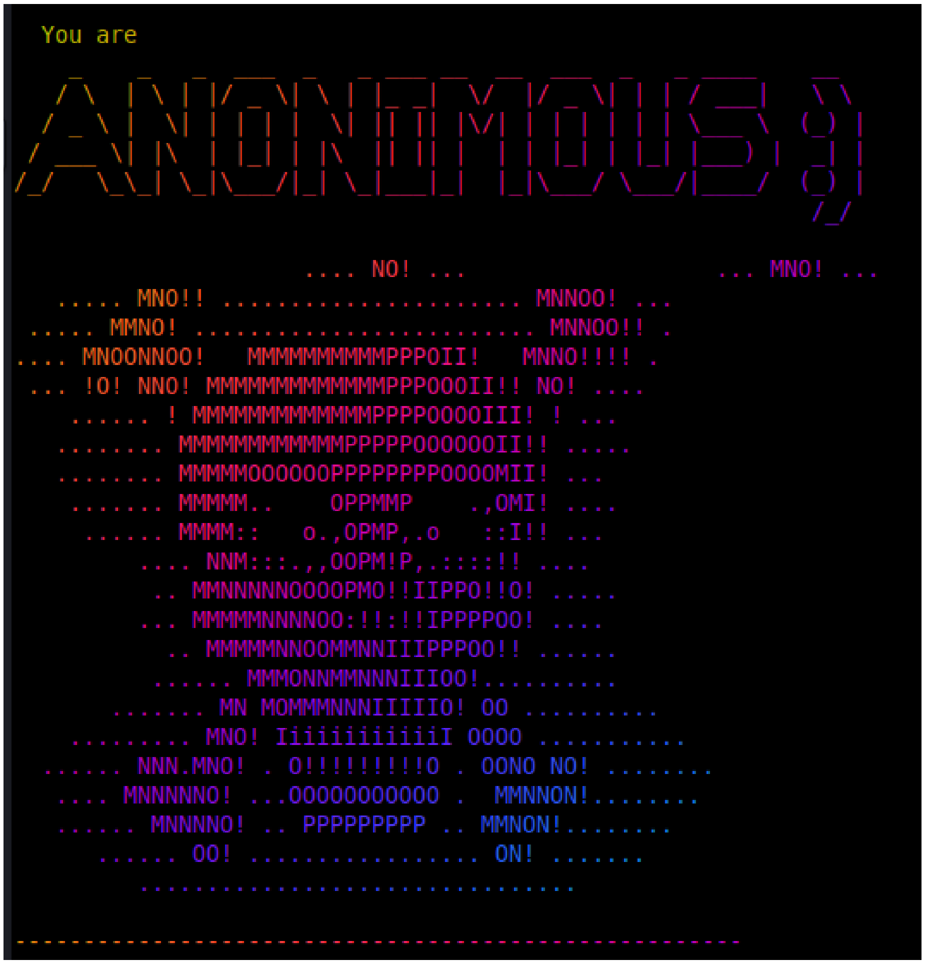


```
# Function checking for anonymous
function ANON()
{
    CNTRY=$(curl -s ifconfig.me)
    if [ -z "$(geoiplookup $CNTRY | grep -i country | grep -i IL)"  ] | lolcat -af
    then
        echo "  You are" | lolcat -af
        echo "$(figlet "ANONIMOUS :) ")" | lolcat -af
        echo "
             .... NO! ...                  ... MNO! ...
   ..... MNO!! ...................... MNNOO! ...
 ..... MMNO! ......................... MNNOO!! .
.... MNOONNOO!   MMMMMMMMMMPPPOII!   MNNO!!!! .
 ... !O! NNO! MMMMMMMMMMMMMPPPOOOII!! NO! ....
    ...... ! MMMMMMMMMMMMMPPPPOOOOIII! ! ...
   ........ MMMMMMMMMMMMPPPPPOOOOOOII!! .....
   ........ MMMMMOOOOOOPPPPPPPPOOOOMII! ...
    ....... MMMMM..    OPPMMP    .,OMI! ....
     ...... MMMM::   o.,OPMP,.o   ::I!! ...
         .... NNM:::.,,OOPM!P,.::::!! ....
          .. MMNNNNNOOOOPMO!!IIPPO!!O! .....
         ... MMMMMNNNNOO:!!:!!IPPPPOO! ....
           .. MMMMMNNOOMMNNIIIPPPOO!! ......
          ...... MMMONNMMNNNIIIOO!.........
       ....... MN MOMMMNNNIIIIIO! OO ..........
    ......... MNO! IiiiiiiiiiiI OOOO ...........
  ...... NNN.MNO! . O!!!!!!!!!O . OONO NO! ........
   .... MNNNNNO! ...OOOOOOOOOO .  MMNNON!........
   ...... MNNNNO! .. PPPPPPPPP .. MMNON!........
      ...... OO! ................. ON! .......
         ................................
        " | lolcat -a
        echo "------------------------------------------------------" | lolcat -af
```



```

    else
        echo "You are not anonymous.[*] Starting nipe services" | lolcat -af
        #starting nipe services
        cd /home/kali/nipe
        perl nipe.pl start 1>/dev/null
        perl nipe.pl stop 1>/dev/null
        perl nipe.pl restart 1>/dev/null
        perl nipe.pl status
        echo "Now you are..." | lolcat -af
        echo "
             .... NO! ...                  ... MNO! ...
   ..... MNO!! ...................... MNNOO! ...
 ..... MMNO! ......................... MNNOO!! .
.... MNOONNOO!   MMMMMMMMMMPPPOII!   MNNO!!!! .
 ... !O! NNO! MMMMMMMMMMMMMPPPOOOII!! NO! ....
    ...... ! MMMMMMMMMMMMMPPPPOOOOIII! ! ...
   ........ MMMMMMMMMMMMPPPPPOOOOOOII!! .....
   ........ MMMMMOOOOOOPPPPPPPPOOOOMII! ...
    ....... MMMMM..    OPPMMP    .,OMI! ....
     ...... MMMM::   o.,OPMP,.o   ::I!! ...
         .... NNM:::.,,OOPM!P,.::::!! ....
          .. MMNNNNNOOOOPMO!!IIPPO!!O! .....
         ... MMMMMNNNNOO:!!:!!IPPPPOO! ....
           .. MMMMMNNOOMMNNIIIPPPOO!! ......
          ...... MMMONNMMNNNIIIOO!..........
       ....... MN MOMMMNNNIIIIIO! OO ..........
    ......... MNO! IiiiiiiiiiiI OOOO ...........
  ...... NNN.MNO! . O!!!!!!!!!O . OONO NO! ........
   .... MNNNNNO! ...OOOOOOOOOOO .  MMNNON!........
   ...... MNNNNO! .. PPPPPPPPP .. MMNON!........
      ...... OO! ................. ON! .......
         ................................
        " | lolcat -a
        echo "$(figlet "ANONIMOUS :)" )" | lolcat -af
        echo "-----------------------------------------------------" | lolcat -af

    fi
}
```

Output displayed:



```
  You are
    _    _   _  ___  _   _ ___ __  __  ___  _   _ ____    __
   / \  | \ | |/ _ \| \ | |_ _|  \/  |/ _ \| | | / ___|  _\ \
  / _ \ |  \| | | | |  \| || || |\/| | | | | | | \___ \ (_) |
 / ___ \| |\  | |_| | |\  || || |  | | |_| | |_| |___) | _| |
/_/   \_\_| \_|\___/|_| \_|___|_|  |_|\___/ \___/|____/ ( ) |
                                                         /_/

                    .... NO! ...                  ... MNO! ...
   ..... MNO!! ...................... MNNOO! ...
 ..... MMNO! ......................... MNNOO!! .
.... MNOONNOO!   MMMMMMMMMMPPPOII!   MNNO!!!! .
 ... !O! NNO! MMMMMMMMMMMMMPPPOOOII!! NO! ....
    ...... ! MMMMMMMMMMMMMPPPPOOOOIII! ! ...
   ........ MMMMMMMMMMMMPPPPPOOOOOOII!! .....
   ........ MMMMMOOOOOOPPPPPPPPOOOOMII! ...
    ....... MMMMM..    OPPMMP    .,OMI! ....
     ...... MMMM::   o.,OPMP,.o   ::I!! ...
         .... NNM:::.,,OOPM!P,.::::!! ....
          .. MMNNNNNOOOOPMO!!IIPPO!!O! .....
         ... MMMMMNNNNOO:!!:!!IPPPPOO! ....
           .. MMMMMNNOOMMNNIIIPPPOO!! ......
          ...... MMMONNMMNNNIIIOO!..........
       ....... MN MOMMMNNNIIIIIO! OO ..........
    ......... MNO! IiiiiiiiiiiiI OOOO ...........
  ...... NNN.MNO! . O!!!!!!!!!O . OONO NO! ........
   .... MNNNNNO! ...OOOOOOOOOOO .  MMNNON!........
   ...... MNNNNO! .. PPPPPPPPP .. MMNON!........
      ...... OO! ................. ON! .......
         ................................

-----------------------------------------------------
```

- Line 72: “-z” means the result command is empty (“! -z” -> contrary/not empty).

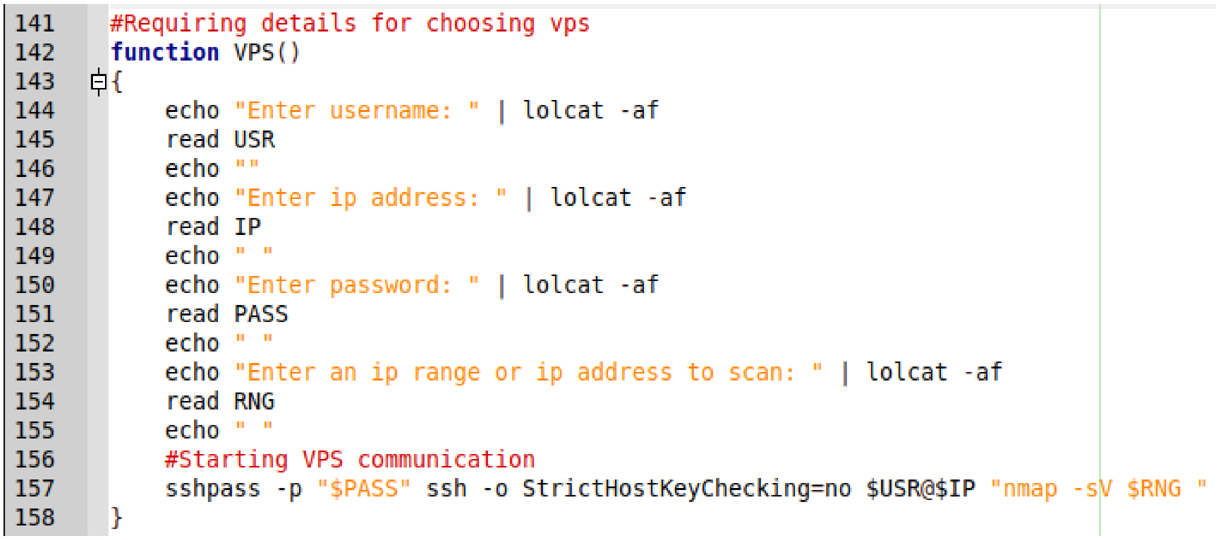
s.p - figlet command isn’t necessary. Utility for creating ascii text banners or large letters out of ordinary text.

Function 3 - Communicate via ssh and execute commands

- The following screenshot specify a connection with ssh protocol ignoring fingerprint or any authorizations by using sshpass (using internal variables).

The bellow script shows an Internal Variable script that requires user input. When creating a script the read command is used by specifying a variable name and the variable content is updated within the Internal Variable once the user enters data while executing the script.

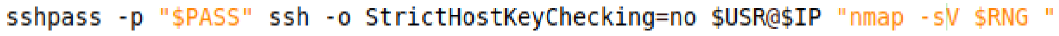
- * read it's a command capture interactive user input during a script is running.
- On the kali-srv starting the sshpass service.



```
#Requiring details for choosing vps
function VPS()
{
    echo "Enter username: " | lolcat -af
    read USR
    echo ""
    echo "Enter ip address: " | lolcat -af
    read IP
    echo " "
    echo "Enter password: " | lolcat -af
    read PASS
    echo " "
    echo "Enter an ip range or ip address to scan: " | lolcat -af
    read RNG
    echo " "
    #Starting VPS communication
    sshpass -p "$PASS" ssh -o StrictHostKeyChecking=no $USR@$IP "nmap -sV $RNG "
}
```

- This function allows individual users to use the system according to their own variables data.

- In order to use ssh service as a master to his “agent/s” and get into the server to login, we need 4 elements: **User name, ip address** (both of the server), **password** & **fingerprint**.



```
sshpass -p "$PASS" ssh -o StrictHostKeyChecking=no $USR@$IP "nmap -sV $RNG "
```

can use as well hostname and scanme

```
'hostname'
```

```
'nmap scanme.nmap.org'
```

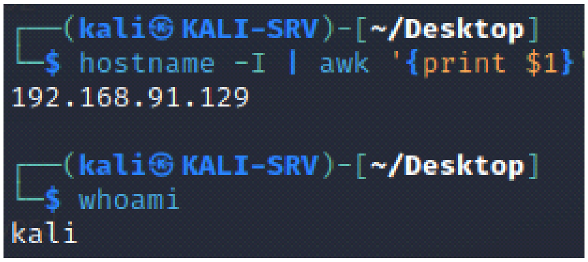
- sshpass -p (for password) “$pass”: allows using the password

- StrictHostKeyChecking=no: Ignore fingerprint.

- username@ip address : Connect to a particular “agent”. For example, my other kali-Vm . After then the master is connected to his server and he’s ready for brute force attack without risking expose himself (protected under nipe).

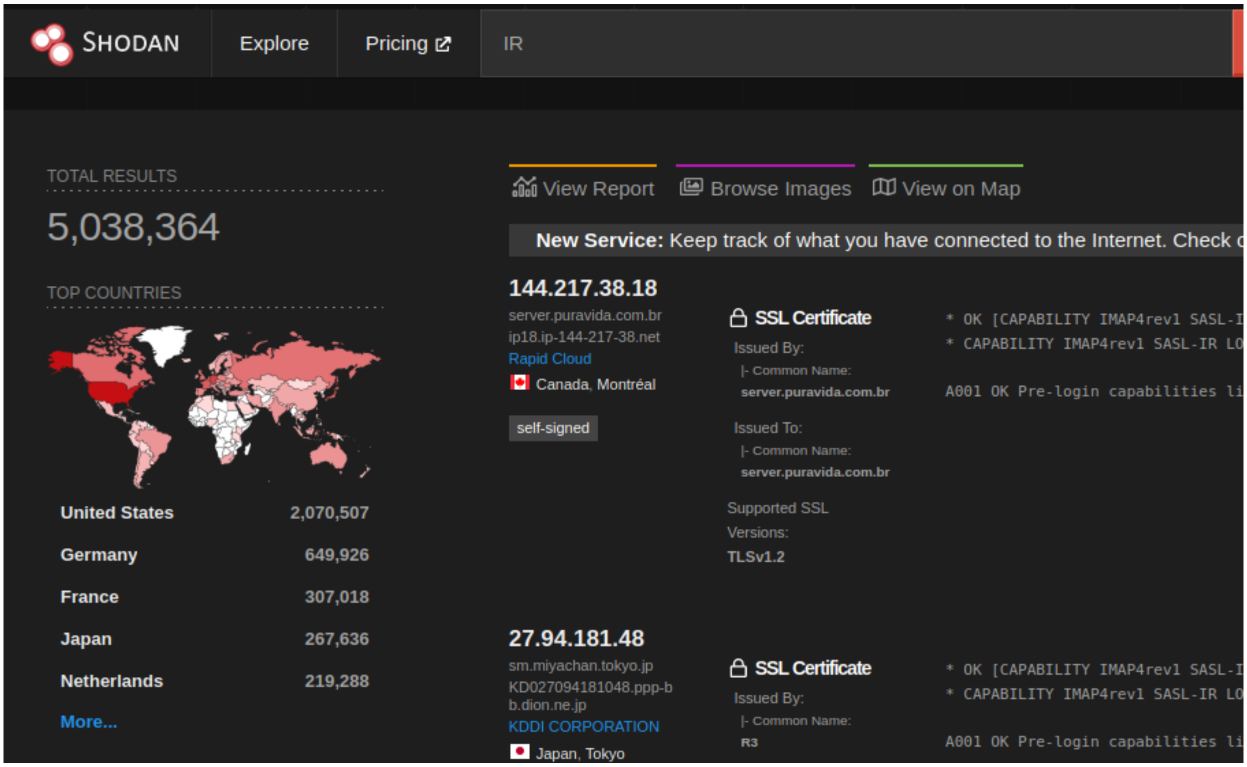
- nmap discover host (if pc’s exist or not) and services. “-Pn” – Don’t discover pc’s out of the range ($RNG). “-sv” – version, using a single port (22).

- RNG scans ip addresses range & port.
* Kali - Srv vm use as a service. Password & User name = kali.



```
┌──(kali㉿KALI-SRV)-[~/Desktop]
└─$ hostname -I | awk '{print $1}'
192.168.91.129

┌──(kali㉿KALI-SRV)-[~/Desktop]
└─$ whoami
kali
```

Target IP range took from Shodan (Iran for example - using country code IR).



IR

TOTAL RESULTS

5,038,364

TOP COUNTRIES

United States	2,070,507
Germany	649,926
France	307,018
Japan	267,636
Netherlands	219,288
More...	

New Service: Keep track of what you have connected to the Internet. Check c

144.217.38.18
server.puravida.com.br
ip18.ip-144-217-38.net
Rapid Cloud
Canada, Montréal
self-signed

SSL Certificate
Issued By:
|- Common Name:
server.puravida.com.br
Issued To:
|- Common Name:
server.puravida.com.br
Supported SSL Versions:
TLSv1.2

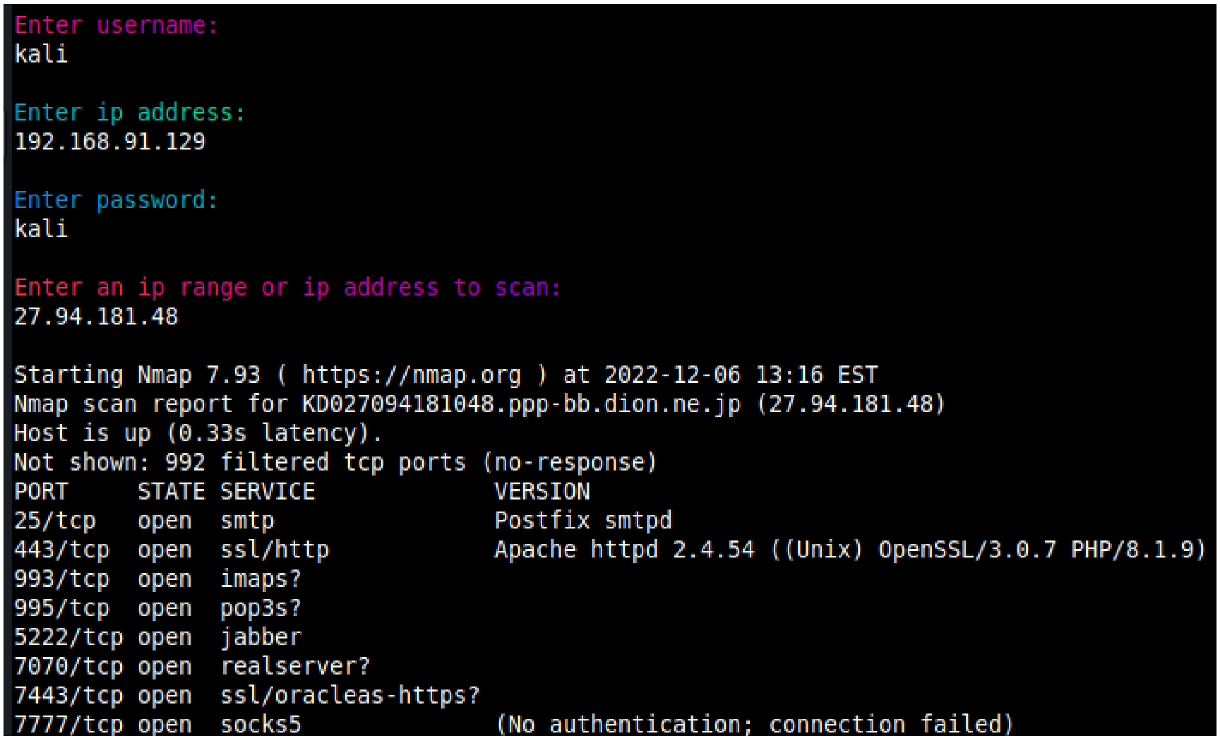
```
* OK [CAPABILITY IMAP4rev1 SASL-I
* CAPABILITY IMAP4rev1 SASL-IR LO

A001 OK Pre-login capabilities li
```

27.94.181.48
sm.miyachan.tokyo.jp
KD027094181048.ppp-bb.dion.ne.jp
KDDI CORPORATION
Japan, Tokyo

SSL Certificate
Issued By:
|- Common Name:
R3

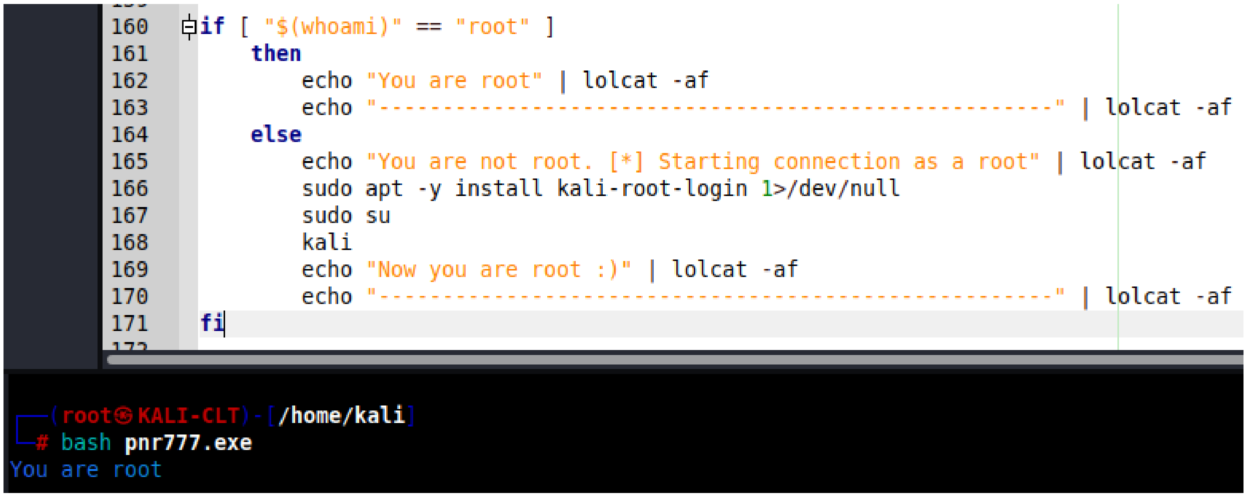
```
* OK [CAPABILITY IMAP4rev1 SASL-I
* CAPABILITY IMAP4rev1 SASL-IR LO

A001 OK Pre-login capabilities li
```

Outprint displayed:



```
Enter username:
kali

Enter ip address:
192.168.91.129

Enter password:
kali

Enter an ip range or ip address to scan:
27.94.181.48

Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-06 13:16 EST
Nmap scan report for KD027094181048.ppp-bb.dion.ne.jp (27.94.181.48)
Host is up (0.33s latency).
Not shown: 992 filtered tcp ports (no-response)
PORT     STATE SERVICE            VERSION
25/tcp   open  smtp               Postfix smtpd
443/tcp  open  ssl/http           Apache httpd 2.4.54 ((Unix) OpenSSL/3.0.7 PHP/8.1.9)
993/tcp  open  imaps?
995/tcp  open  pop3s?
5222/tcp open  jabber
7070/tcp open  realserver?
7443/tcp open  ssl/oracleas-https?
7777/tcp open  socks5             (No authentication; connection failed)
```

* In order to use and perform all commands including special permissions we’ll performe/active as a root.



```
160  if [ "$(whoami)" == "root" ]
161      then
162          echo "You are root" | lolcat -af
163          echo "------------------------------------------------------" | lolcat -af
164      else
165          echo "You are not root. [*] Starting connection as a root" | lolcat -af
166          sudo apt -y install kali-root-login 1>/dev/null
167          sudo su
168          kali
169          echo "Now you are root :)" | lolcat -af
170          echo "------------------------------------------------------" | lolcat -af
171  fi
```

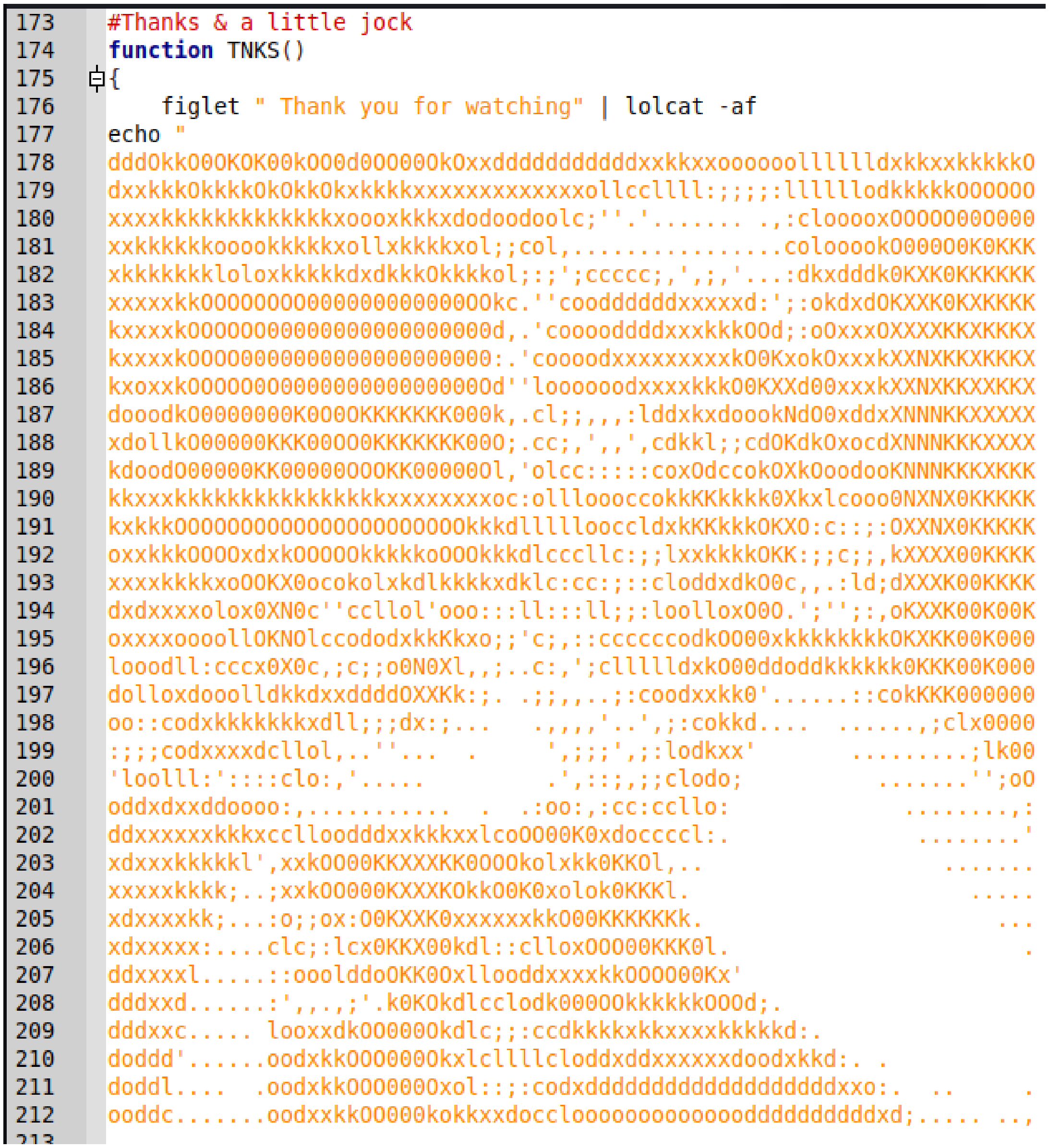
```
┌──(root㉿KALI-CLT)-[/home/kali]
└─# bash pnr777.exe
You are root
```

- line166: I’ve been looking for another way to get in the root without being asking for password. The command < sudo apt -y install kali-root-login > allows this direct connection as you can see at the following screenshot.

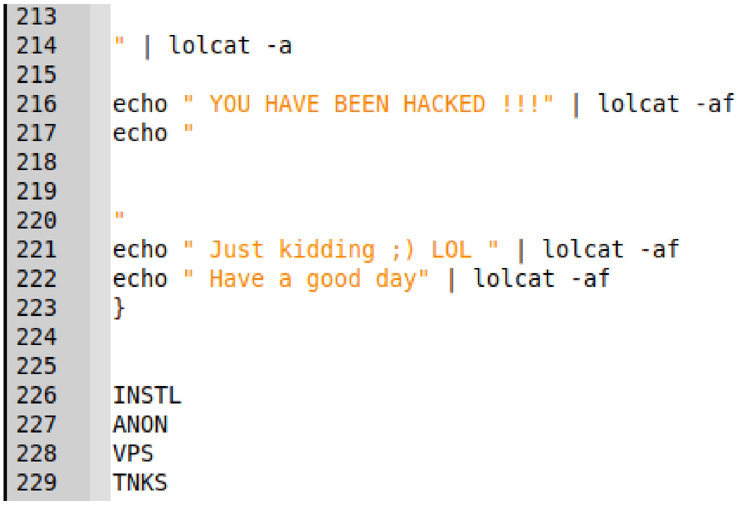
Credit enabling root's command - https://www.kali.org/docs/general-use/enabling-root/#enabling-root-for-gnome-and-kde-login .

- Same line, 1>/dev/null. No’ 1 represents output in linux channels* (See at the bottom of the folders) and it Injects to the trash (/dev/null).

- Note: In case of errors during nipe installation perform the recommendations specified apt-get update --fix-missing.

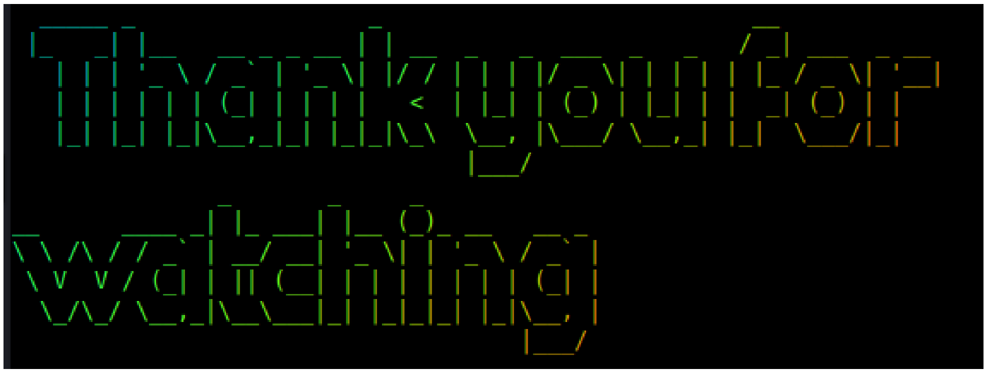


```
#Thanks & a little jock
function TNKS()
{
    figlet " Thank you for watching" | lolcat -af
echo "
dddOkkO0OKOK00kOO0d0OO000kOxxdddddddddddxxkkxxooooooollllldxkkxxkkkkkO
dxxkkkOkkkkOkOkkOkxkkkkxxxxxxxxxxxxxxollcclllll:;;;;;:llllllodkkkkkOOOOOO
xxxxkkkkkkkkkkkkkxoooxkkkxdodoodoolc;''.'....... .,:cloooox0OOOOOOOOO
xxkkkkkkoooookkkkxollxkkkkxol;;col,...............coloooookO000O0K0KKK
xkkkkkkkloloxkkkkdxdkkkOkkkkol;:;';ccccc;,',;,'...:dkxdddk0KXK0KKKKKK
xxxxxkkOOOOOOOO0000000000000OOkc.''cooddddddxxxxxd:';:okdxdOKXXK0KXKKKK
kxxxxkOOOOOO000000000000000000d,.'cooooddddxxxkkkOOd;:oOxxxOXXXXKKXKKKX
kxxxxkOOOOO000000000000000000:.'cooooodxxxxxxxxxkO0KxokOxxxkXXNXKKXKKKX
kxoxxkOOOOO00000000000000000Od''loooooodxxxxkkkO0KXXd00xxxkXXNXKKXXKKX
dooodkO0000000K0O0OKKKKKKK000k,.cl;;,,,:lddxkxdoookNdO0xddxXNNNKKXXXXX
xdollkO00000KKK00OO0KKKKKKK000;.cc;,',,',cdkkl;;cdOKdkOxocdXNNNKKXXXX
kdoodO00000KK00000OOOKK00000Ol,'olcc::::coxOdccokOXkOoodooKNNNKKKXKKK
kkxxxkkkkkkkkkkkkkkkkxxxxxxxxxoc:ollloooccokkKKkkkk0XkxlcooO0NXNX0KKKKK
kxkkkOOOOOOOOOOOOOOOOOOOOOkkkdllllloocclddxkKKkkkOKXO:c::;:OXXNX0KKKKK
oxxkkkOOOOxdxkOOOOOkkkkkoOOOkkkdlcccllc:;;lxxkkkkOKK:;;c;;,kXXXX00KKKK
xxxxkkkkxoOOKX0ocokolxkdlkkkkxdklc:cc:;::cloddxdkO0c,,.:ld;dXXXK00KKKK
dxdxxxxolox0XN0c''ccllol'ooo:::ll:::ll;;:loollox000.';'';:,oKXXK00K00K
oxxxxooooll0KNOlccododxkkKkxo;;'c;,::ccccccodkOO00xkkkkkkkkOKXKK00K000
looodll:cccx0X0c,;c;;o0N0Xl,,;..c:,';clllllldxkO00ddoddkkkkkk0KKK00K000
dolloxdooolldkkdxxddddOXXKk:;. .;;,,..;:coodxxkk0'.....::cokKKK000000
oo::codxkkkkkkkxdll;;dx:;...    .,,,,'..',;:cokkd....  ......,;clx0000
:;;;codxxxxdcllol,..''...  .     ',;;;',;:lodkxx'       .........;lk00
'loolll:'::::clo:,'.....         .',::;,;;clodo;          .......'';oO
oddxdxxddoooo:,..........   .  .:oo:,:cc:ccllo:             ........,:
ddxxxxxxkkkxccllooddxxkkkxxlcoOO00K0xdoccccl:.               ........'
xdxxxkkkkl',xxkOO00KKXXXKK0OOOkolxkk0KKOl,..                   .......
xxxxxkkkk;..;xxkOO000KXXXKOkkO0K0xolok0KKKl.                     .....
xdxxxxkk;...:o;;ox:O0KXXK0xxxxxxkkO00KKKKKk.                       ...
xdxxxxx:....clc;:lcx0KKX00kdl::clloxOO000KKK0l.                      .
ddxxxxl.....::ooolddoOKK0OxllooddxxxxkkOOOO00Kx'
dddxxd......:',,.,;'.k0KOkdlcclodk000OOkkkkkkOOOd;.
dddxxc..... looxxdkOO000Okdlc;;:ccdkkkkxkkxxxxkkkkkd:.
doddd'......oodxkkOOO000Okxlcllllcloddxddxxxxxxdoodxkkd:. .
doddl....  .oodxkkOOO000Oxol::;:codxdddddddddddddddddddxxo:.  ..     .
ooddc.......oodxxkkOO000kokkxxdoccloooooooooooooddddddddddxd;..... ..,
```

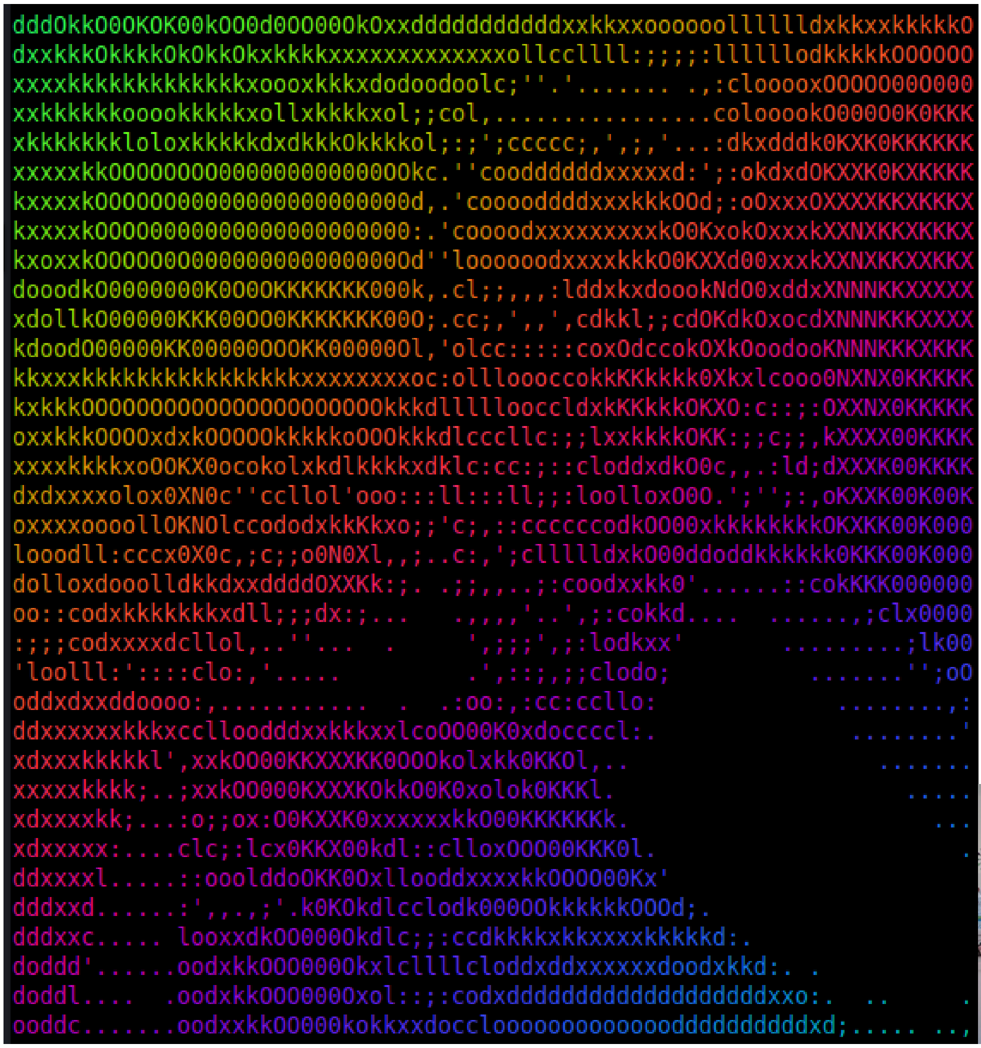


```

" | lolcat -a

echo " YOU HAVE BEEN HACKED !!!" | lolcat -af
echo "


"
echo " Just kidding ;) LOL " | lolcat -af
echo " Have a good day" | lolcat -af
}


INSTL
ANON
VPS
TNKS
```

Output:

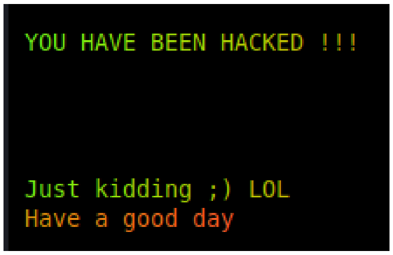


```
 _____ _                 _                            __            
|_   _| |__   __ _ _ __ | | __  _   _  ___  _   _    / _| ___  _ __ 
  | | | '_ \ / _` | '_ \| |/ / | | | |/ _ \| | | |  | |_ / _ \| '__|
  | | | | | | (_| | | | |   <  | |_| | (_) | |_| |  |  _| (_) | |   
  |_| |_| |_|\__,_|_| |_|_|\_\  \__, |\___/ \__,_|  |_|  \___/|_|   
                                |___/                               
                 _       _     _             
__      ____ _| |_ ___| |__ (_)_ __   __ _ 
\ \ /\ / / _` | __/ __| '_ \| | '_ \ / _` |
 \ V  V / (_| | || (__| | | | | | | | (_| |
  \_/\_/ \__,_|\__\___|_| |_|_|_| |_|\__, |
                                     |___/ 
```



```
dddOkkOOOKOK00kOO0d0OO000kOxxddddddddddddxxkkxxooooooollllllldxkkxxkkkkkO
dxxkkkOkkkkOkOkkOkxkkkkxxxxxxxxxxxxxxxollccllll:;;;;;:lllllllodkkkkkOOOOOO
xxxxkkkkkkkkkkkkkkxoooxkkkxdodoodoolc;''.'....... .,:clooooxOOOOOOOOOOO
xxkkkkkkooookkkkkxollxkkkkxol;;col,.................colooookOOOOOOKOKKK
xkkkkkkkloloxkkkkkdxdkkkOkkkkol;:;';ccccc;,',;,'...:dkxdddk0KXK0KKKKKK
xxxxxkkOOOOOOOOOOOOOOOOOOOOOOkc.''coodddddddxxxxxd:';:okdxdOKXXK0KXKKKK
kxxxxkOOOOOOOOOOOOOOOOOOOOOOOd,.'cooooddddxxxkkkOOd;:oOxxxOXXXXKKXKKKX
kxxxxkOOOOOOOOOOOOOOOOOOOOOOO:.'cooooodxxxxxxxxxkO0KxokOxxxkXXNXKKXKKKX
kxoxxkOOOOOOOOOOOOOOOOOOOOOOOd''loooooodxxxxkkkO0KXXd00xxxkXXNXKKXXKKX
dooodkO0000000K0O0OKKKKKKK000k,.cl;;,,,:lddxkxdoookNdO0xddxXNNNKKXXXXX
xdollkO00000KKK00OO0KKKKKKK000;.cc;,',,',cdkkl;;cdOKdkOxocdXNNNKKKXXXX
kdoodO00000KK0000000OKK000000l,'olcc:::::coxOdccokOXkOoodooKNNNKKKXKKK
kkxxxkkkkkkkkkkkkkkkkkkxxxxxxxxxoc:olllooocc okkKKkkkk0XkxlcoooONXNX0KKKKK
kxkkkOOOOOOOOOOOOOOOOOOOOOkkkdllllllooccldxkKKkkkOKXO:c::;:OXXNX0KKKKK
oxxkkkOOOOxdxkOOOOOkkkkkoOOOkkkdlcccllc:;;lxxkkkkOKK:;;c;;,kXXXX00KKKK
xxxxkkkkxoOOKX0ocokolxkdlkkkkxdklc:cc:;::cloddxdkO0c,,.:ld;dXXXK00KKKK
dxdxxxxolox0XN0c''ccllol'ooo:::ll:::ll;;:loollox000.';'';:,oKXXK00K00K
oxxxxooooll0KNOlccododxkkKkxo;;'c;,::ccccccodkOO00xkkkkkkkkOKXKK00K000
looodll:cccx0X0c,;c;;o0N0Xl,,;..c:,';clllllldxkO00ddoddkkkkkk0KKK00K000
dolloxdooolldkkdxxddddOXXKk:;. .;;,,..;:coodxxkk0'......::cokKKK000000
oo::codxkkkkkkkxdll;;;dx:;...   .,,,,'..',;:cokkd....  .......,;clx0000
:;;;codxxxxdcllol,..''...  .      ',;;;',;:lodkxx'        .........;lk00
'loolll:'::::clo:,'.....          .',::;,;;clodo;           .......'';oO
oddxdxxddoooo:,...........  .   .:oo:,:cc:ccllo:              ........,:
ddxxxxxxkkkxccllooddddxxkkkxxlcoOO00K0xdoccccl:.               ........'
xdxxxkkkkkl',xxkOO00KKXXXKK0OOOkolxkk0KKOl,..                     .......
xxxxxkkkk;..;xxkOO000KXXXKOkkO0K0xolok0KKKl.                        .....
xdxxxxkk;...:o;;ox:O0KXXK0xxxxxxkkO00KKKKKKk.                          ...
xdxxxxx:....clc;:lcx0KKX00kdl::clloxOOO00KKK0l.                          .
ddxxxxl.....::ooolddoOKK0Oxllooddxxxxkk000000Kx'
dddxxd......:',,.,;'.k0KOkdlcclodk0000OkkkkkkOOOd;.
dddxxc.....  looxxdkOO0000kdlc;;:ccdkkkkxkkxxxxxkkkkkd:.
doddd'......oodxkkOOOOOOOkxlclllllcloddxddxxxxxxdoodxkkd:. .
doddl....   .oodxkkOOOOOOOxol::;:codxdddddddddddddddddddddxxo:.  ..     .
ooddc.......oodxxkkO0000kokkxxdocclooooooooooooooddddddddddxd;.....  ..,
```

Converting jpg image to ascii with a simple command > jp2a <filename>



```
YOU HAVE BEEN HACKED !!!



Just kidding ;) LOL
Have a good day
```

(Page no’ 1) Linux channels* has 3 channels.

# Channel 0 - input   > keyboard/mouse. Building keylogger – listening to channel 0.

# Channel 1 - output > monitor. What appears on the screen.

# Channel 2 – error   > Automatic output for no exist files/directories.

Credit for a few genral details - https://www.kalilinux.in/ HYPERLINK "https://www.kalilinux.in/2022/03/bash-scripting-on-kali-linux.html" HYPERLINK "https://www.kalilinux.in/2022/03/bash-scripting-on-kali-linux.html"
Cloroed figlet using 'lolcat' command - https://www.tomshardware.com/how-to/customize-linux-terminal

Ascii art - https://asciiart.website/

https://phoenixnap.com/kb/bash-function