Remote Control

(Network Research Project)

John Saw

Objective

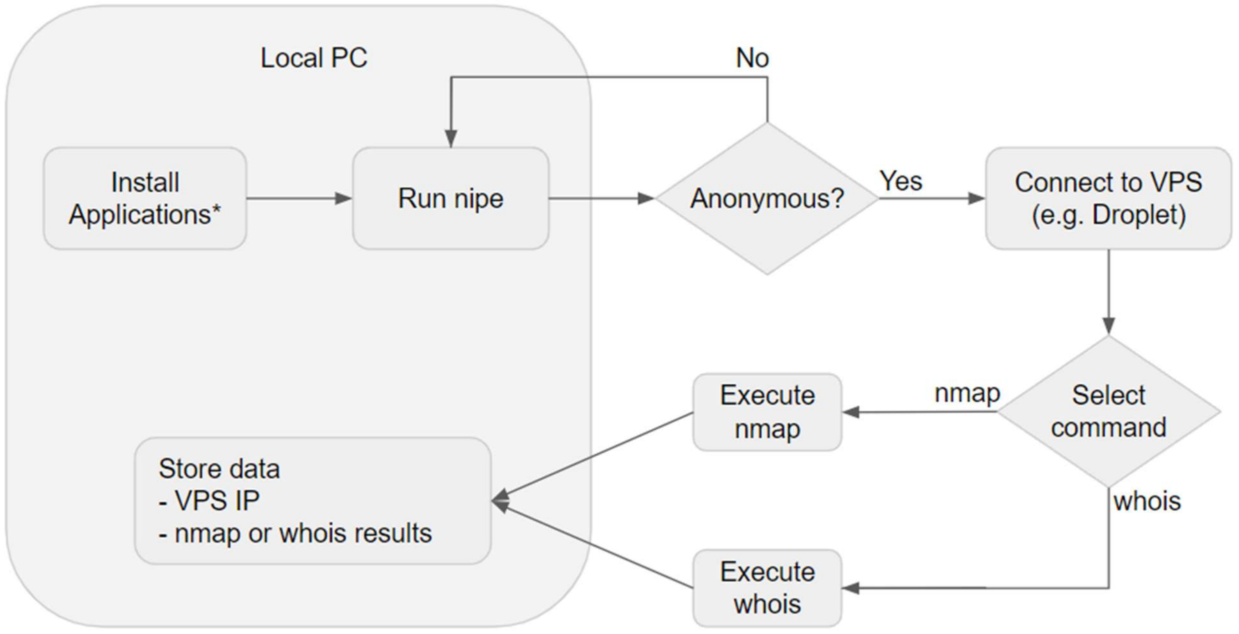
Create a script that communicates with a remote server and executes tasks anonymously.

Required Functions

1. Install relevant applications on the local computer.
2. Check if the connection is from your origin country. If not, continue.
3. Once connection is anonymous, connect to VPS via SSH and execute nmap scans and whois queries.

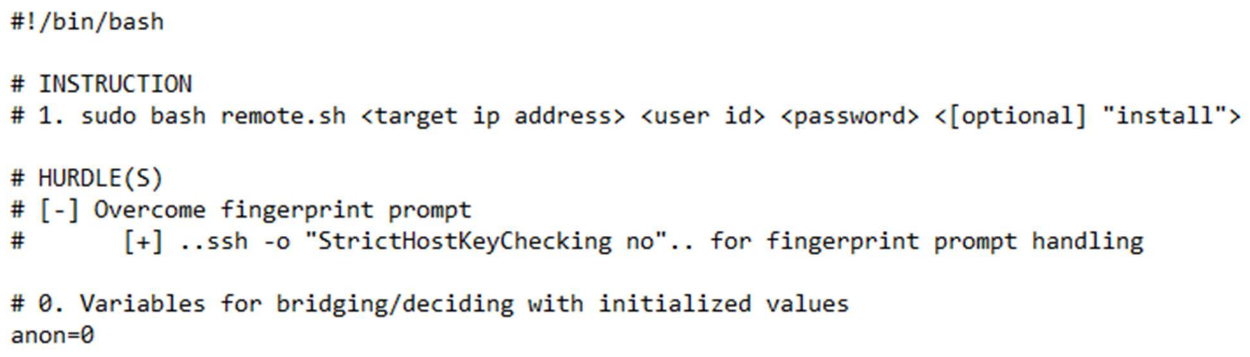
Proposed Solution

Flow map below attempts to explain the order of activities chained by checkpoints.



1. Starting with install applications, intended to be optional so that it occurs at most once for every new local pc that script runs in.
2. Run nipe function to validate whether the user is anonymous.
3. Once anonymity is successful, connection to VPS is allowed via if statement. If not, the nipe function will be auto-recalled for re-attempt(s).
4. Option will be prompted to the user via case statement to help determine whether nmap scan or whois query should be executed.
5. Execution will store data on the local pc with the ip address of VPS connected to.

Proposed Script



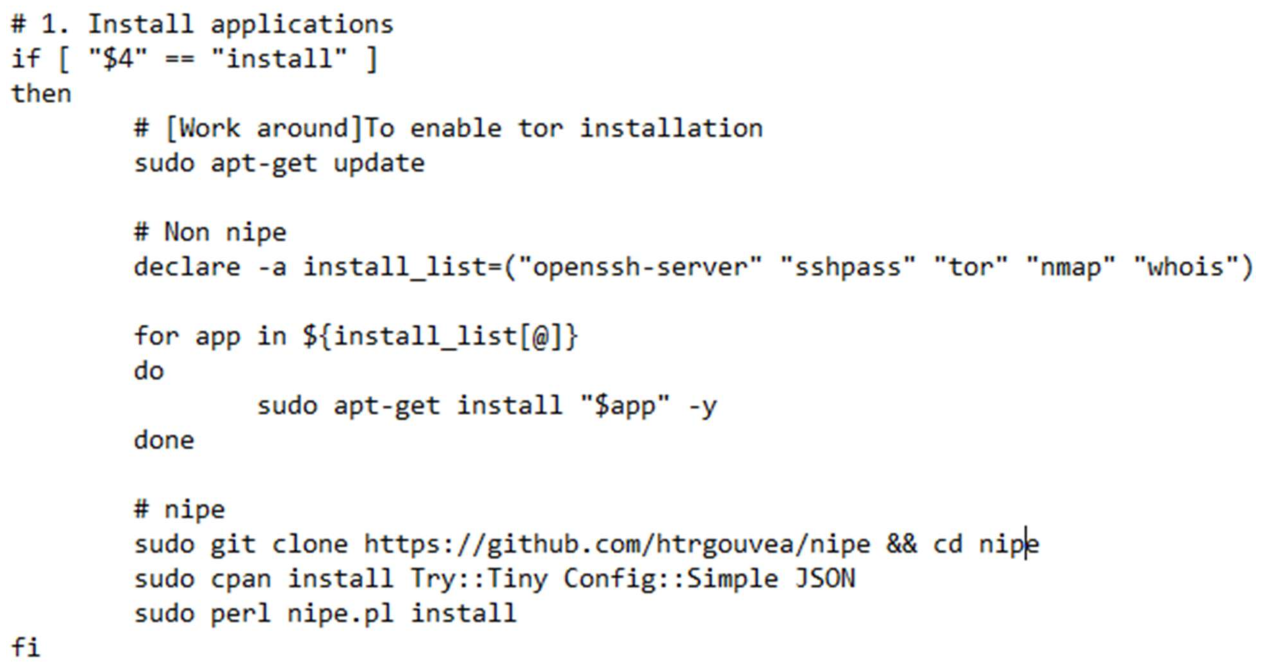
```
#!/bin/bash

# INSTRUCTION
# 1. sudo bash remote.sh <target ip address> <user id> <password> <[optional] "install">

# HURDLE(S)
# [-] Overcome fingerprint prompt
#       [+] ..ssh -o "StrictHostKeyChecking no".. for fingerprint prompt handling

# 0. Variables for bridging/deciding with initialized values
anon=0
```

Script Part I Explanation

Starts with instructions on how to use the script.
E.g. for maiden script run, recommended to indicate install.
sudo bash remote.ssh <target ip address> <user id> <password> *install*

Share on the current hurdle overcome for a new target ip address which prompts for fingerprinting. Thus, to minimize user input adding "-o "StrictHostKeyChecking no" during ssh connection helps.

Reveals that a variable "anon" is set up to help control decision making for connection with an initialized value of 0.

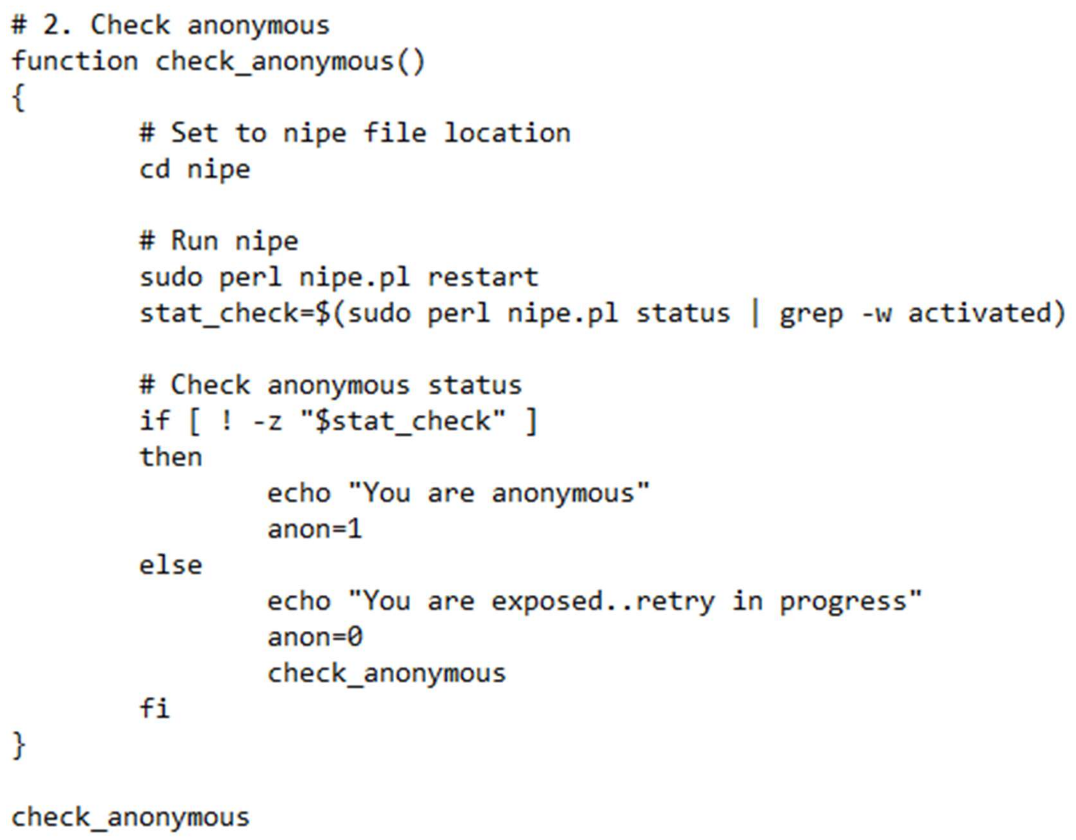


```
# 1. Install applications
if [ "$4" == "install" ]
then
        # [Work around]To enable tor installation
        sudo apt-get update

        # Non nipe
        declare -a install_list=("openssh-server" "sshpass" "tor" "nmap" "whois")

        for app in ${install_list[@]}
        do
                sudo apt-get install "$app" -y
        done

        # nipe
        sudo git clone https://github.com/htrgouvea/nipe && cd nipe
        sudo cpan install Try::Tiny Config::Simple JSON
        sudo perl nipe.pl install
fi
```

Script Part II Explanation

Leveraging on an if statement, 4th argument passed ($4) is used to help decide whether installation is required.

List of applications to install are; openssh-server, sshpass, tor, nmap, whois & nipe.
From trial tests, observed that tor was unable to proceed with installation prior to completing apt-get update activity. Apart from nipe, remaining applications can be installed via for loop.

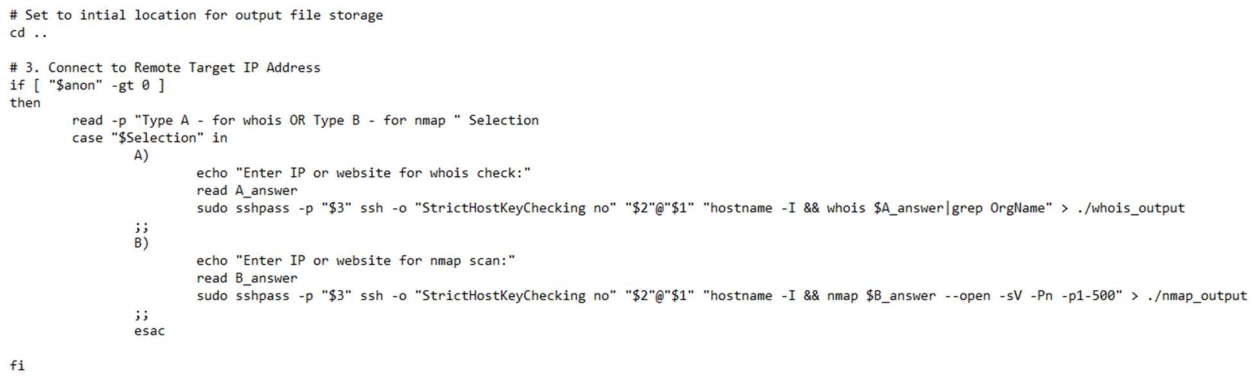


```
# 2. Check anonymous
function check_anonymous()
{
        # Set to nipe file location
        cd nipe

        # Run nipe
        sudo perl nipe.pl restart
        stat_check=$(sudo perl nipe.pl status | grep -w activated)

        # Check anonymous status
        if [ ! -z "$stat_check" ]
        then
                echo "You are anonymous"
                anon=1
        else
                echo "You are exposed..retry in progress"
                anon=0
                check_anonymous
        fi
}

check_anonymous
```

<u>Script Part III Explanation</u>

Leveraging on a function, anonymity can be enabled using nipe.

Upon successful anonymity, variable “anon” is set to 1 which will be used in the next part of the script. Through trial tests, it was observed that sometimes anonymity is unsuccessful therefore to address such incidents, the function is auto-recalled within the if statement in this function.



```
# Set to intial location for output file storage
cd ..

# 3. Connect to Remote Target IP Address
if [ "$anon" -gt 0 ]
then
        read -p "Type A - for whois OR Type B - for nmap " Selection
        case "$Selection" in
                A)
                        echo "Enter IP or website for whois check:"
                        read A_answer
                        sudo sshpass -p "$3" ssh -o "StrictHostKeyChecking no" "$2"@"$1" "hostname -I && whois $A_answer|grep OrgName" > ./whois_output
                ;;
                B)
                        echo "Enter IP or website for nmap scan:"
                        read B_answer
                        sudo sshpass -p "$3" ssh -o "StrictHostKeyChecking no" "$2"@"$1" "hostname -I && nmap $B_answer --open -sV -Pn -p1-500" > ./nmap_output
                ;;
                esac

fi
```

Script Part IV Explanation

From the previous update of variable "anon" to 1, the case statement will prompt the user to decide whether to proceed with whois query ("A") or nmap scan ("B").

Upon entering either "A" or "B", the user will also be asked for an ip address or a website for a targeted execution.

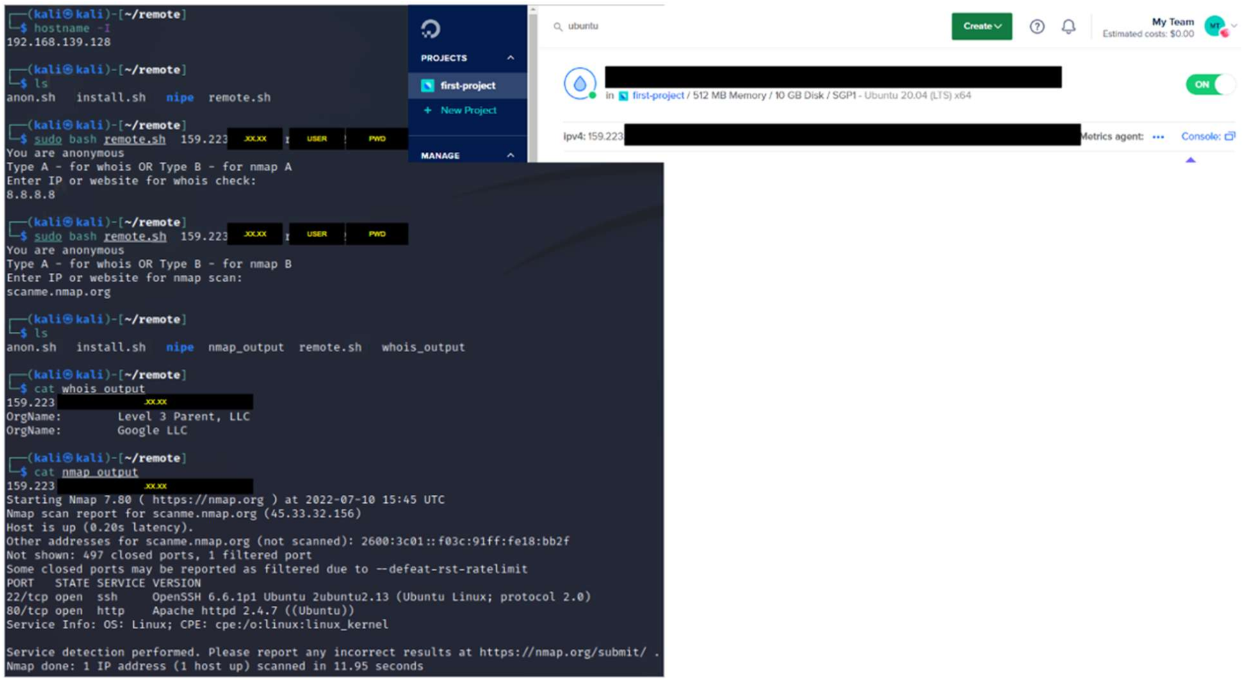
Leveraging on sshpass, script is able to automate ssh connection to ip address (e.g. VPS) using arguments passed during the initial script run via terminal.

E.g.
Terminal execution
> bash remote.sh <target ip address> <user id> <password>
Comparison to script
> sshpass -p <password> ssh -o "StrictHostKeyChecking no" <user id>@<target ip address> ..

Also on sshpass, commands can run on remote machine with output on local machine.
E.g. "Hostname -I && whois $A_answer|grep OrgName" > ./whois_ouput

Note: "cd .." is to set the current working directory back to /remote instead of /remote/nipe for data storage.

Proof of Work



```
┌──(kali㉿kali)-[~/remote]
└─$ hostname -I
192.168.139.128

┌──(kali㉿kali)-[~/remote]
└─$ ls
anon.sh   install.sh   nipe  remote.sh

┌──(kali㉿kali)-[~/remote]
└─$ sudo bash remote.sh  159.223[redacted] XX.XX [redacted] USER [redacted] PWD
You are anonymous
Type A - for whois OR Type B - for nmap A
Enter IP or website for whois check:
8.8.8.8

┌──(kali㉿kali)-[~/remote]
└─$ sudo bash remote.sh  159.223[redacted] XX.XX [redacted] USER [redacted] PWD
You are anonymous
Type A - for whois OR Type B - for nmap B
Enter IP or website for nmap scan:
scanme.nmap.org

┌──(kali㉿kali)-[~/remote]
└─$ ls
anon.sh   install.sh   nipe  nmap_output  remote.sh   whois_output

┌──(kali㉿kali)-[~/remote]
└─$ cat whois_output
159.223[redacted] XX.XX
OrgName:        Level 3 Parent, LLC
OrgName:        Google LLC

┌──(kali㉿kali)-[~/remote]
└─$ cat nmap_output
159.223[redacted] XX.XX
Starting Nmap 7.80 ( https://nmap.org ) at 2022-07-10 15:45 UTC
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.20s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 497 closed ports, 1 filtered port
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.95 seconds
```

[redacted]
in first-project / 512 MB Memory / 10 GB Disk / SGP1 - Ubuntu 20.04 (LTS) x64
ipv4: 159.223[redacted]

<u>Proof of Work Explanation</u>

1. Check the user's local pc ip address.
2. Check current working directory contents.
3. Run remote.sh script with the following arguments.
    - Ip address of remote pc/VPS
    - User ID
    - Password
    - 'install' (comment: for maiden run)
4. Enter 'A' for whois query or 'B' for nmap scan.
5. Check results via command cat 'whois_output' or 'nmap_output'.

Note: Ip address in 'whois_output' or 'nmap_output' will contain the ip address of remote pc/VPS which can be cross-checked with step 1..

Credit and Links

sshpass
https://levelup.gitconnected.com/execute-commands-on-remote-machines-using-sshpass-1f9bc4452e15

sshpass enables the user to run the command on a remote machine from the user's machine and can get the result also on the user's machine.

```
sshpass -p <password> ssh <user id>@<ip address> '<command>' > 'path/file name'
```

fingerprinting prompt
https://stackoverflow.com/questions/21383806/how-can-i-force-ssh-to-accept-a-new-host-fingerprint-from-the-command-line

fingerprinting message prompt when new ip address is used during ssh connection.
Adding -o "StrictHostKeyChecking no" between ssh and <user id> helps to address the prompt automatically.

```
ssh -o "StrictHostKeyChecking no" <user id>
```

```
sshpass -p <password> ssh -o "StrictHostKeyChecking no" <user id>@<ip address> '<command>' > 'path/file name'
```