# REMOTE CONTROL

**Table Of Contents**

## Introduction

This report outlines the creation and use of a comprehensive system designed to automate network and server analysis tasks. The system starts by installing necessary tools, or skipping the process if the tools are already installed. It performs an anonymity check on the network connection, providing alerts if the connection is not anonymous and revealing the spoofed country after the connection is masked. Additionally, the system accepts user inputs for the scan targets, executes remote SSH commands to gather server information, and performs network scans such as Whois queries and open port detection. All collected data is stored in files in the local machine, and an audit log is maintained to track the activities.

To better understand network and security, the remote network traffic is captured during the automated system connection. Finally, an unsecure protocol will be analysed.

## Methodologies

As the script needs to be automated, certain packages are needed in the machine for the script to run.

We first check if the machine had the needed packages and download them if it did not.


Figure 1: The dpkg Command in Linux (Digital Ocean, 2024)

The command dpkg was used to check if the needed packages were installed.

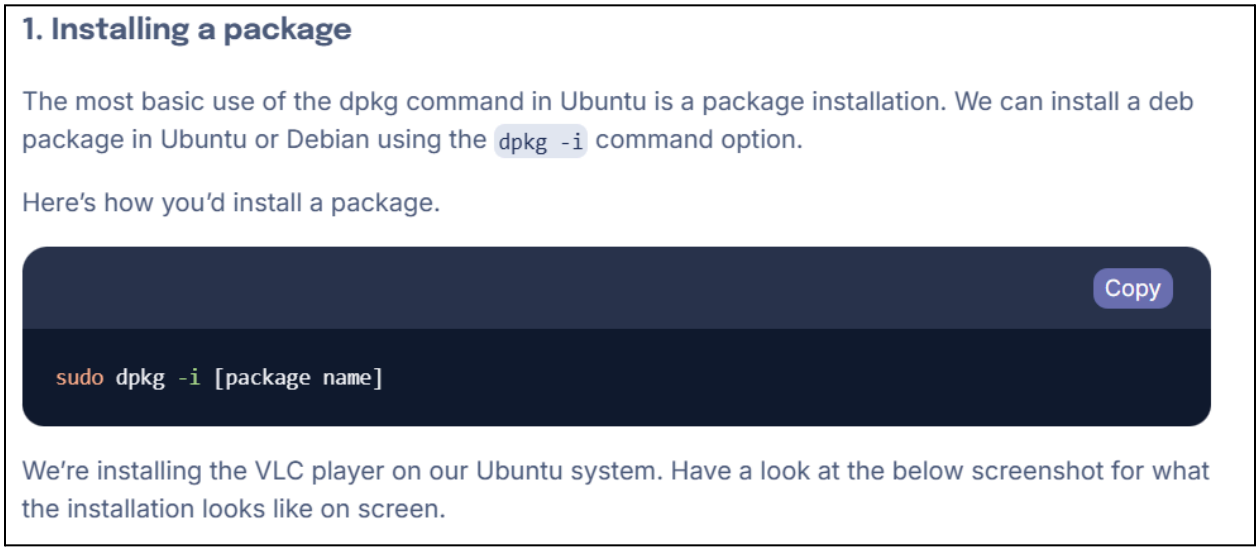



Figure 2: The dpkg Command in Linux (Digital Ocean, 2024)

REMOTE CONTROL

## Example Script:

bash Copy code

```bash
#!/bin/bash

# List of desired packages
PACKAGES=("curl" "git" "vim" "htop")

# Function to check if a package is installed
is_installed() {
    dpkg-query -l "$1" &>/dev/null
}

# Loop through the list of packages
for package in "${PACKAGES[@]}"; do
    if is_installed "$package"; then
        echo "$package is already installed."
    else
        echo "$package is not installed. Installing..."
        sudo apt-get update
        sudo apt-get install -y "$package"
    fi
done

echo "Package check and installation complete."
```

## Explanation:

1. **Desired Packages**:
   - The `PACKAGES` array holds a list of package names you want to ensure are installed.
2. **Checking if a Package is Installed**:
   - The `is_installed` function uses `dpkg-query -l` to check whether a package is installed. The command `dpkg-query -l "$1" &>/dev/null` will return success if the package is installed, and failure if it isn't.
3. **Package Installation**:
   - If the package is not installed, it uses `sudo apt-get install -y "$package"` to install the missing package.
   - `sudo apt-get update` ensures that your local package database is up to date before installation.
4. **Looping through Packages**:
   - The script loops through the `PACKAGES` array and checks if each package is installed, installing it if necessary.

## Important Notes:

- The script uses `apt-get` for installing packages instead of `dpkg` because `apt-get` handles dependencies automatically (whereas `dpkg` does not). `dpkg` can be used directly if you have `.deb` files, but `apt-get` is typically the preferred method for package installation from repositories.
- The script needs to be run with root privileges ( `sudo` ) to install packages.

Figure 3-5: "How to bash script to check and download missing packages using dpkg" prompt. (ChatGPT, 2024)

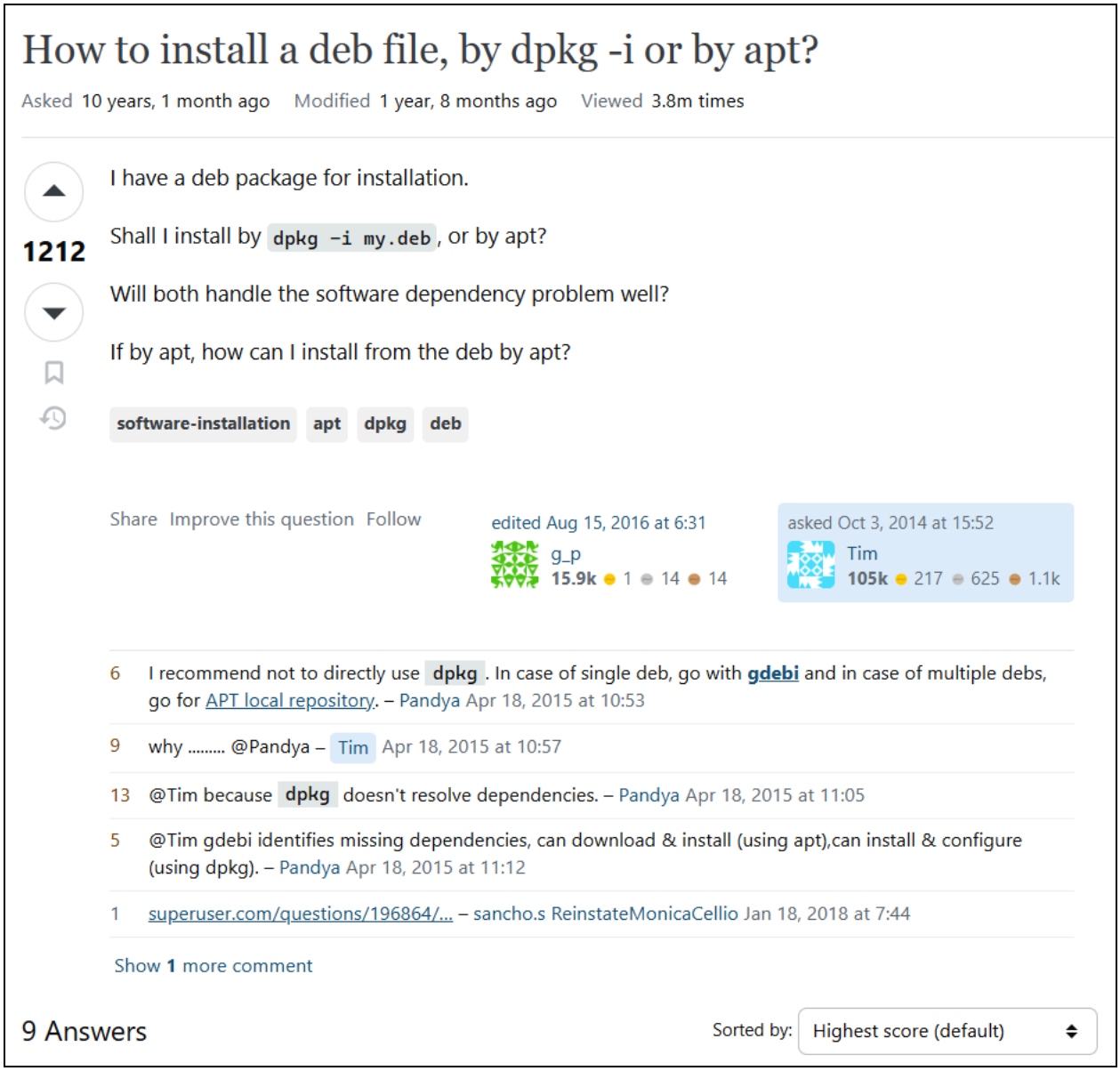



Figure 6: “How to Install a Deb File, by Dpkg -i or by Apt?” (Stack Exchange, 2024)

As multiple sources said don’t use dpkg -i command to install packages we went with sudo apt-get install instead in the script. The -y flag will auto reply yes to any questions during the install so the user does not need to enter any input, which automates the whole process.

REMOTE CONTROL

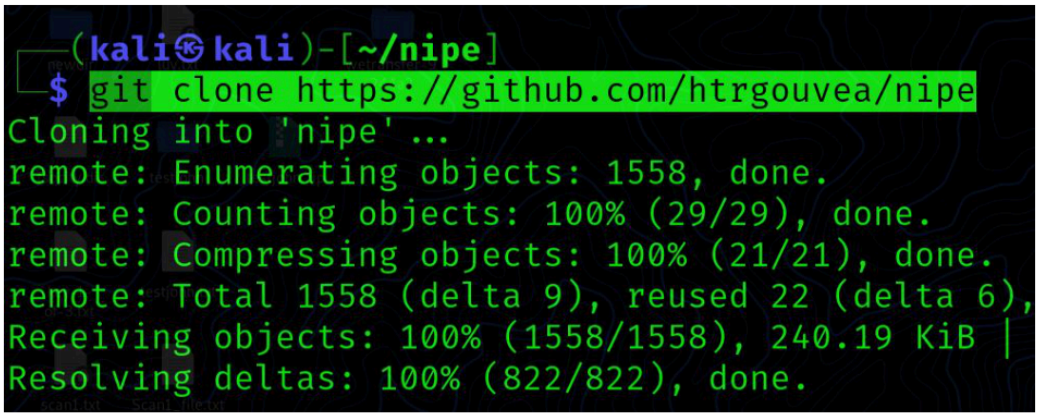
As we are also assuming the machine does not have Nipe, we need it to be installed as we want to connect to a remote server anonymously.

**Nipe** is a program that uses the Tor network as the user's default gateway, routing all traffic on the Tor network, which is often used to provide privacy and anonymity. It should be emphasized that hiding an IP address alone will not provide anonymity when using a tool for privacy and anonymity, as DNS information may still be exposed. Both IP and DNS information must be hidden.

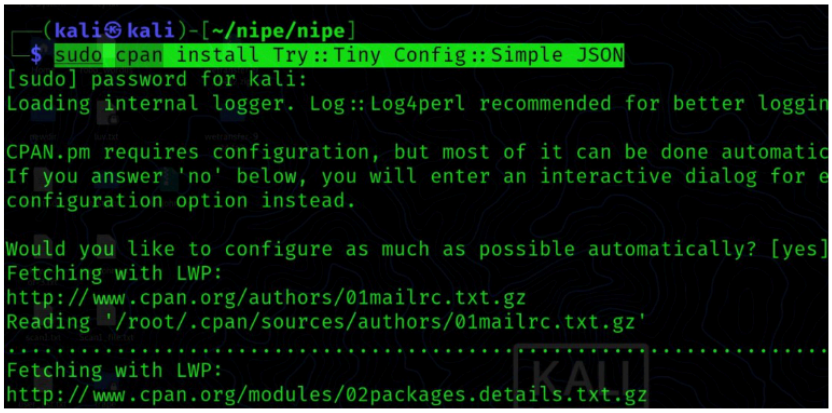
Staying anonymous is a great way to protect yourself from all kinds of surveillance. However, we only have a few options because VPNs, especially free ones, are quite ineffective. We can be tracked because a free VPN maintains logs. We can just use the TOR network instead of the browser. Tor is quite difficult to track (close to practically impossible).

**Step 2:** Then, using the following command, we must clone this repository from GitHub:

```
git clone https://github.com/htrgouvea/nipe
```



```
sudo cpan install Try::Tiny Config::Simple JSON
```




**Step 4:** After that, we can use the following command to install Nipe dependencies or a Perl script:

Nipe must be run as root.

```
sudo perl nipe.pl install
```

Type the following command to start the Nipe service:

```
sudo perl nipe.pl start
```


Figure 7-11: “How to Install Nipe Tool in Kali Linux?” (GeeksforGeeks, 2024)

As nipe is not a package,we cannot use apt-get install, instead the command git clone was used to copy nipe from github.

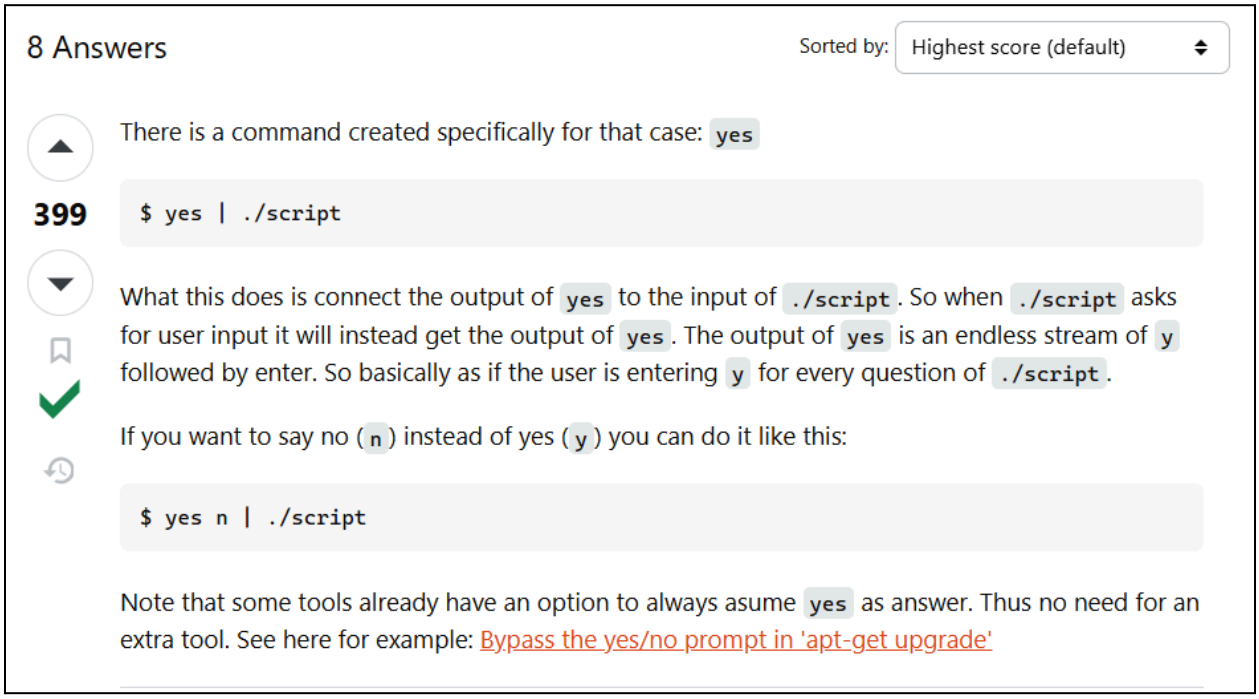



Figure 12: "Automatically Enter Input in Command Line." (Ask Ubuntu, 2024)

The $ yes flag will automatically reply yes to the prompt from the download so the user does not need to enter any input.

Next we need to install perl.

Once both nipe and perl are on the machine, we can run nipe by using the syntax sudo perl nipe start.

When the command perl nipe.pl status is entered it will show if it is an anonymous connection or if nipe failed to establish a connection.

We need the connection to be anonymous before moving on to the next step. If the connection is not anonymous, the script will restart the nipe service until an anonymous connection is established.

Once the connection is anonymous, the script asks the user to provide the information needed to connect to a remote server to scan their desired domain.

```
106  # This section ask user for the needed variables to access the remote server
107
108  echo Please provide a domain/url
109  read domain
110  echo
111  echo Please provide the remote server
112  read server
113  echo Please provide the username
114  read username
115  echo 'Please provide the password (input is hidden, press enter when done)'
116  read -s password
```

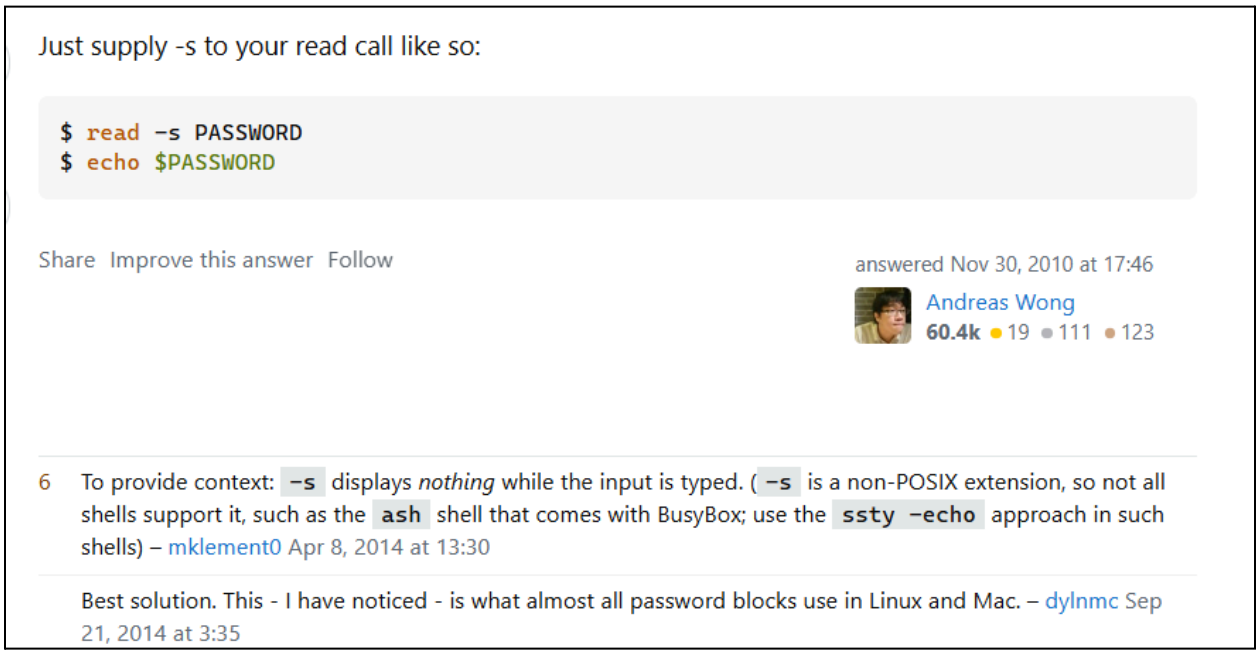



Figure 13: “Hiding User Input on Terminal in Linux Script.”( Stack Overflow, 2024)

The -s flag will hide the user input from displaying on the terminal.

Next we used sshpass to connect to the remote server.

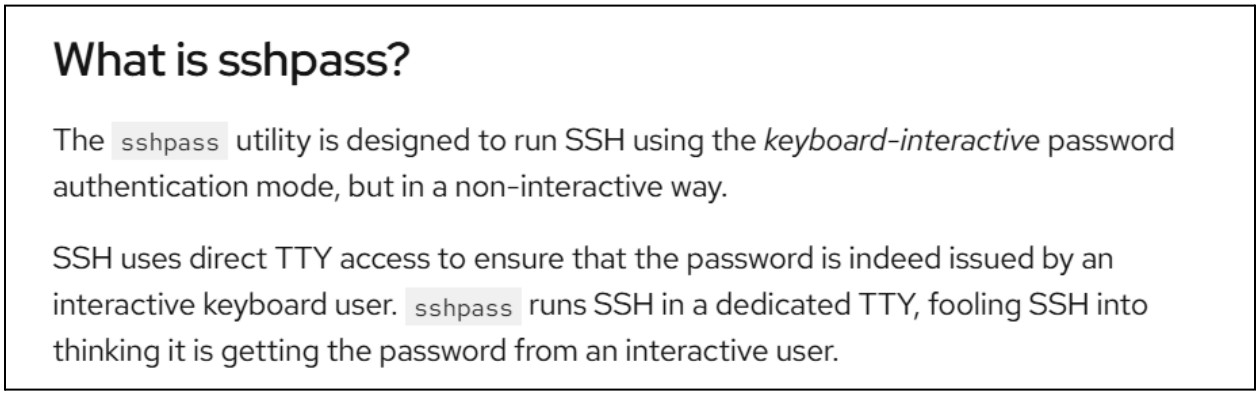



Figure 14: “SSH Password Automation in Linux with Sshpass.” (Redhat.com, 2024)


Figure 15: “How to Put Sshpass Command inside a Bash Script?” (Stack Overflow,2024)

Sshpass helps to run the ssh connection in a non-interactive way. The -p flag is for the password. The -o flag with the string StrictHostKeyChecking=no will add the host key to the known host if this is the first time establishing a connection to the remote server.

Rather than adding it to your `~/.ssh/config` file for all Host *, it would be a safer to specify a particular host.

373

You can also pass a parameter on the command-line like this:

```
ssh -o StrictHostKeyChecking=no yourHardenedHost.com
```

This will automatically add the host key to your known_hosts file if it's not already there.

If there's a mismatch, it will display a big warning and not update known_hosts. It will also disable password-based authentication to prevent MITM attacks. Private key authentication will still automatically get through though, which you may not want.

Share Improve this answer Follow

edited Sep 14, 2021 at 9:31
mwfearnley
3,437 ● 2 ● 23 ● 29

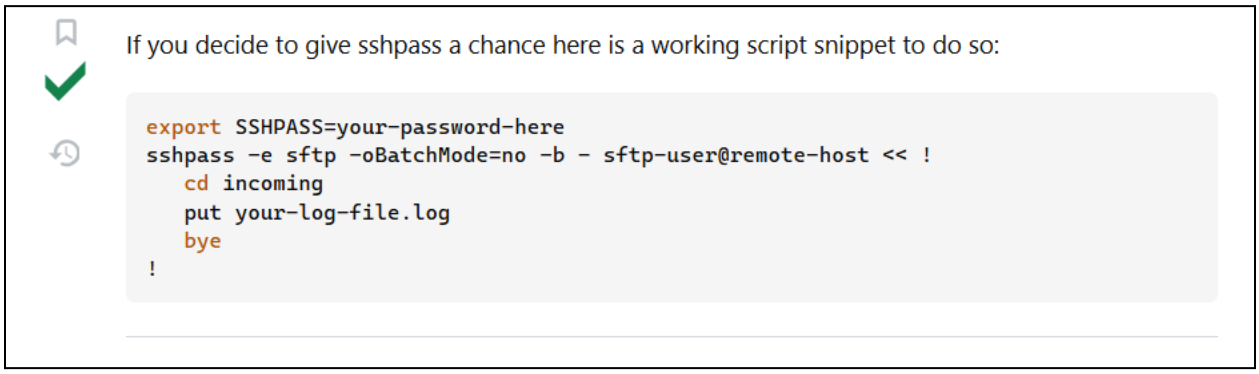
answered Jul 25, 2012 at 1:27
MarkHu
6,100 ● 2 ● 17 ● 15

Figure 16: “How to Disable Strict Host Key Checking in Ssh?” (Ask Ubuntu,2024)

Once we have connected to the remote server, we run the scans and save the outputs into files on the remote server.

Next we need to download the files from the remote server to the local machine.

We used sshpass and sftp to download the files to the local machine.



If you decide to give sshpass a chance here is a working script snippet to do so:

```
export SSHPASS=your-password-here
sshpass -e sftp -oBatchMode=no -b - sftp-user@remote-host << !
   cd incoming
   put your-log-file.log
   bye
!
```

Figure 17: “How to Run the Sftp Command with a Password from Bash Script?” (Stack Overflow,2024)

Lastly, we need to log down the events.

## Create a log entry

To manually create a log entry in Linux, you can use the *logger* command. This command serves as an interface to the **syslog system log module** and it is commonly used in scripts. For example, a backup script might use the *logger* command to record details such as its start and stop times and the number of files it has backed up.

Here is a simple example:

```
susel:~ # logger This is a log message
susel:~ # tail /var/log/messages
Nov  5 19:32:40 susel dhcpcd[3022]: eth0: adding default route via 192.168.198.2
 metric 0
Nov  5 19:32:40 susel ifup:     eth0      device: Intel Corporation 82545EM Giga
bit Ethernet Controller (Copper) (rev 01)
Nov  5 19:32:40 susel SuSEfirewall2: SuSEfirewall2 not active
Nov  5 19:47:40 susel dhcpcd[3022]: eth0: renewing lease of 192.168.198.128
Nov  5 19:47:40 susel dhcpcd[3022]: eth0: leased 192.168.198.128 for 1800 second
s
Nov  5 19:47:40 susel dhcpcd[3022]: eth0: adding IP address 192.168.198.128/24
Nov  5 19:47:40 susel dhcpcd[3022]: eth0: adding default route via 192.168.198.2
 metric 0
Nov  5 19:47:40 susel ifup:     eth0      device: Intel Corporation 82545EM Giga
bit Ethernet Controller (Copper) (rev 01)
Nov  5 19:47:40 susel SuSEfirewall2: SuSEfirewall2 not active
Nov  5 19:55:28 susel root: This is a log message
```

Figure 18: “Create a Log Entry ” (*Geek University*, 2024)

Logger command was used to log down the events.

By default, *logger* includes its name in the log file as the tag. To change the tag, use the *-t TAG* option:

```
susel:~ # logger -t SOME_TEXT Another message.
susel:~ # tail /var/log/messages
Nov  5 19:58:12 susel syslog-ng[1320]: Log statistics; dropped='pipe(/dev/xconso
le)=0', dropped='pipe(/dev/tty10)=0', processed='center(queued)=991', processed=
'center(received)=536', processed='destination(messages)=506', processed='destin
ation(mailinfo)=2', processed='destination(mailwarn)=0', processed='destination(
localmessages)=117', processed='destination(newserr)=0', processed='destination(
mailerr)=0', processed='destination(netmgm)=0', processed='destination(warn)=170
', processed='destination(console)=83', processed='destination(null)=10', proces
sed='destination(mail)=2', processed='destination(xconsole)=83', processed='dest
ination(firewall)=0', processed='destination(acpid)=18', processed='destination(
newscrit)=0', processed='destination(newsnotice)=0', processed='source(src)=536'
Nov  5 20:01:53 susel logger: This is some text.
Nov  5 20:01:53 susel logger: This is also some text.
Nov  5 20:02:41 susel dhcpcd[3022]: eth0: renewing lease of 192.168.198.128
Nov  5 20:02:41 susel dhcpcd[3022]: eth0: leased 192.168.198.128 for 1800 second
s
Nov  5 20:02:41 susel dhcpcd[3022]: eth0: adding IP address 192.168.198.128/24
Nov  5 20:02:41 susel dhcpcd[3022]: eth0: adding default route via 192.168.198.2
 metric 0
Nov  5 20:02:41 susel ifup:     eth0      device: Intel Corporation 82545EM Giga
bit Ethernet Controller (Copper) (rev 01)
Nov  5 20:02:41 susel SuSEfirewall2: SuSEfirewall2 not active
Nov  5 20:04:05 susel SOME_TEXT: Another message.
```

Figure 19: “Create a Log Entry ” (*Geek University*, 2024)

The -t flag was added to tag the log entries.

Next we will be capturing the network traffic on the remote server.

We used tcpdump for this.

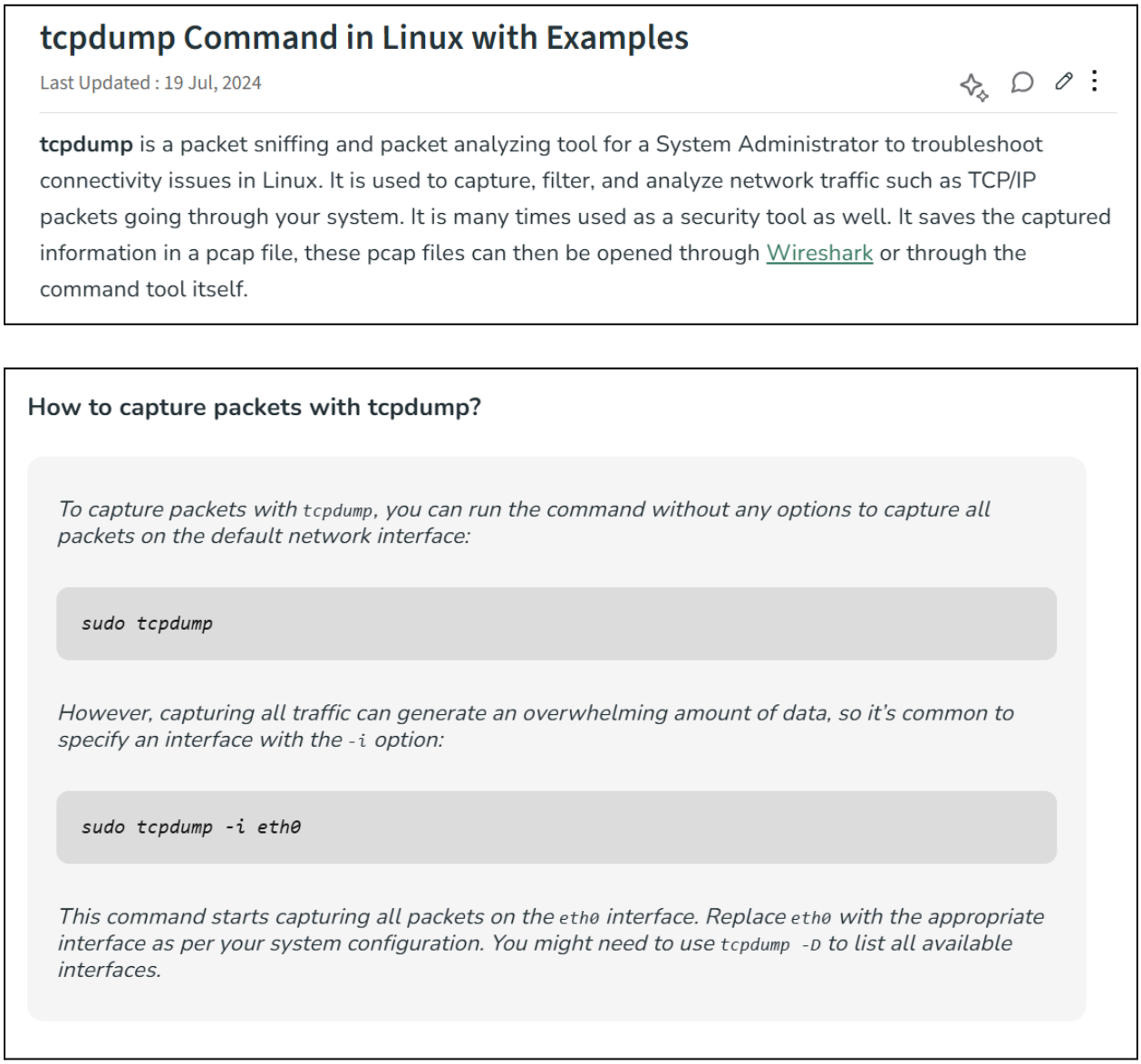



Figure 20-21 "Tcpdump Command in Linux with Examples." (GeeksforGeeks, 2024)

We used the -D flag to display all the available interfaces

```
tc@server:~$ sudo tcpdump -D
1.ens33 [Up, Running, Connected]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]
5.nflog (Linux netfilter log (NFLOG) interface) [none]
6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
7.dbus-system (D-Bus system bus) [none]
8.dbus-session (D-Bus session bus) [none]
```

Then tcpdump was activated on the remote server before we started the script on the other machine.

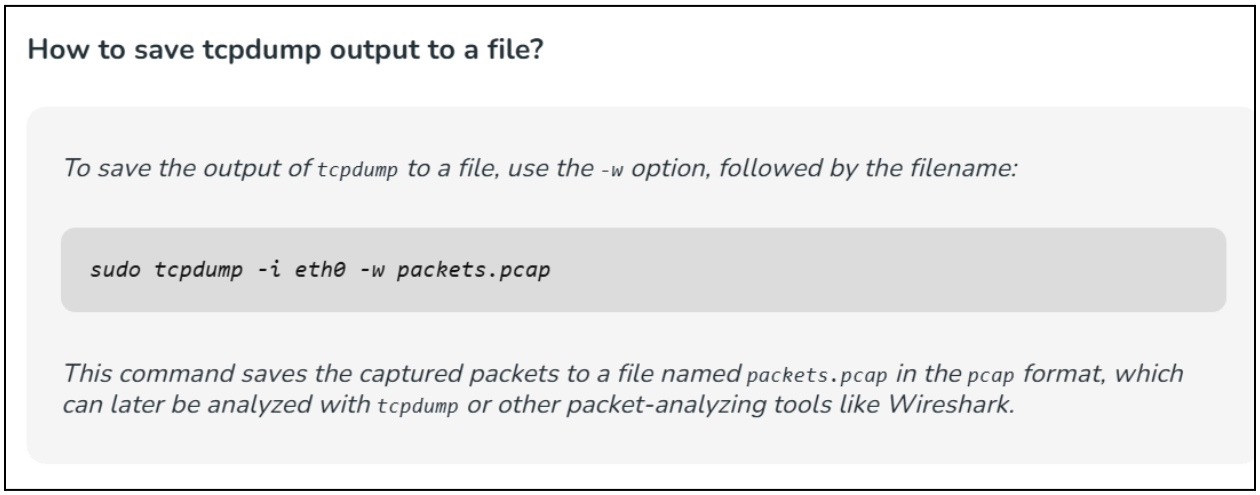
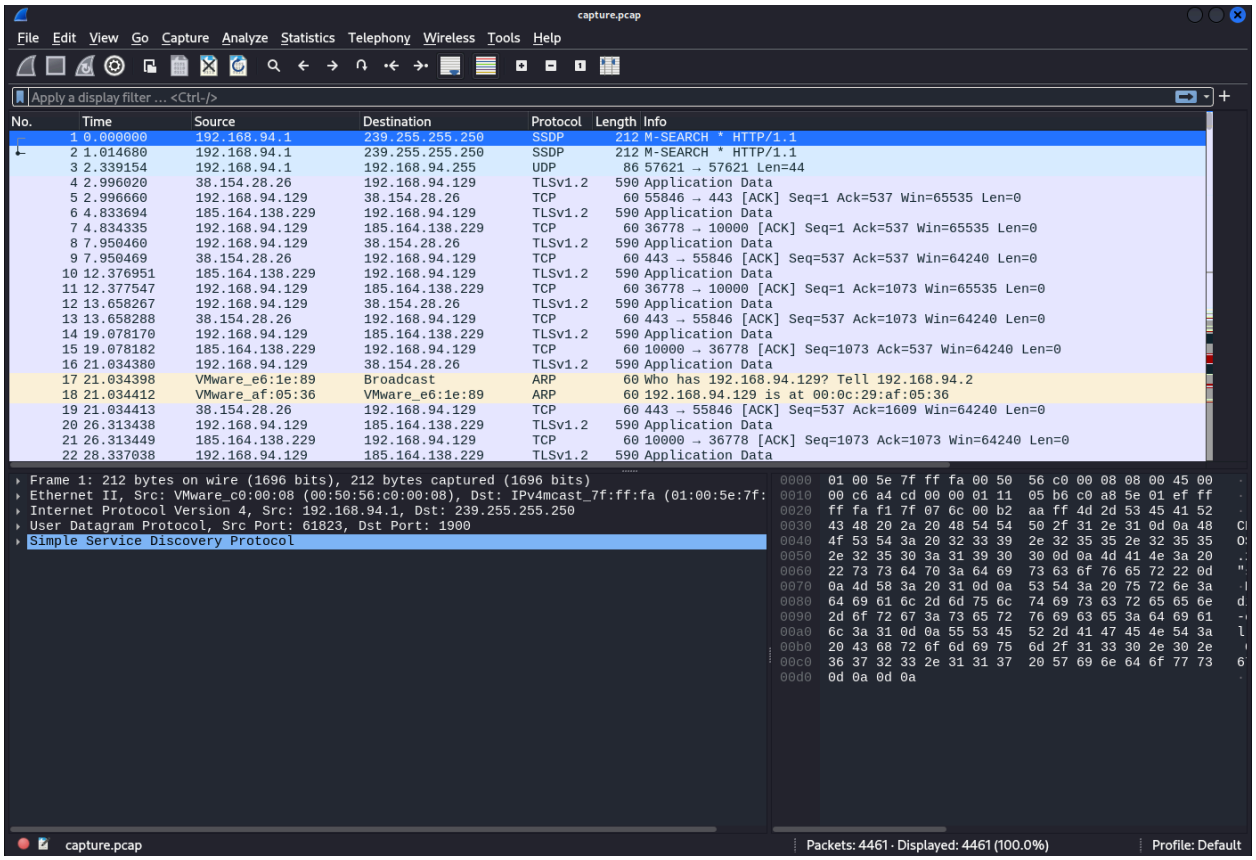



Figure 22: “Tcpdump Command in Linux with Examples.” (GeeksforGeeks, 2024)

REMOTE CONTROL

The file was then analysed using wireshark.




```
183 46.265073   192.168.94.129   192.168.94.130   TCP      66 55554 → 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1388321905 TSecr
184 46.265434   192.168.94.129   192.168.94.130   SSHv2    98 Client: Protocol (SSH-2.0-OpenSSH_9.9p1 Debian-1)
185 46.265465   192.168.94.130   192.168.94.129   TCP      66 22 → 55554 [ACK] Seq=1 Ack=33 Win=65152 Len=0 TSval=273490210 TSecr
186 46.283950   192.168.94.130   192.168.94.129   SSHv2   108 Server: Protocol (SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.10)
187 46.284784   192.168.94.129   192.168.94.130   TCP      66 55554 → 22 [ACK] Seq=33 Ack=43 Win=64256 Len=0 TSval=1388321925 TSe
188 46.285662   192.168.94.129   192.168.94.130   TCP    1514 55554 → 22 [ACK] Seq=33 Ack=43 Win=64256 Len=1448 TSval=1388321926
189 46.285674   192.168.94.129   192.168.94.130   SSHv2   186 Client: Key Exchange Init
190 46.286207   192.168.94.130   192.168.94.129   SSHv2  1178 Server: Key Exchange Init
191 46.329158   192.168.94.129   192.168.94.130   SSHv2  1274 Client: Diffie-Hellman Key Exchange Init
192 46.341555   192.168.94.130   192.168.94.129   SSHv2  1514 Server: Diffie-Hellman Key Exchange Reply, New Keys
193 46.341652   192.168.94.130   192.168.94.129   SSHv2   182 Server:
194 46.342668   192.168.94.129   192.168.94.130   TCP      66 55554 → 22 [ACK] Seq=2809 Ack=2719 Win=64000 Len=0 TSval=138832198
195 46.347906   192.168.94.129   192.168.94.130   SSHv2    82 Client: New Keys
196 46.390242   192.168.94.130   192.168.94.129   TCP      66 22 → 55554 [ACK] Seq=2719 Ack=2825 Win=64128 Len=0 TSval=273490335
```

We can see the SSH connection.

```
181 46.263933   192.168.94.129   192.168.94.130   TCP      74 55554 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=138832
182 46.263998   192.168.94.130   192.168.94.129   TCP      74 22 → 55554 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM T
183 46.265073   192.168.94.129   192.168.94.130   TCP      66 55554 → 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1388321905 TSecr=27
184 46.265434   192.168.94.129   192.168.94.130   SSHv2    98 Client: Protocol (SSH-2.0-OpenSSH_9.9p1 Debian-1)
185 46.265465   192.168.94.130   192.168.94.129   TCP      66 22 → 55554 [ACK] Seq=1 Ack=33 Win=65152 Len=0 TSval=273490210 TSecr=13
```

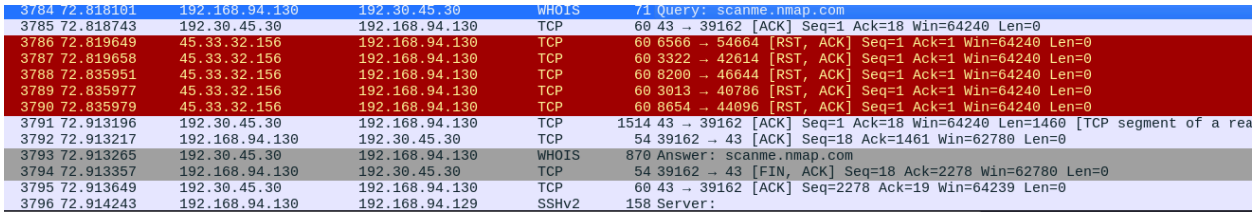
We can also see the TCP 3 way handshake.

```
221 46.908448   192.168.94.130   192.168.94.2     DNS      75 Standard query 0x6b6a A scanme.nmap.com
222 46.908581   192.168.94.130   192.168.94.2     DNS      75 Standard query 0xb2d7 AAAA scanme.nmap.com
223 46.914493   VMware_e6:1e:89  Broadcast        ARP      60 Who has 192.168.94.130? Tell 192.168.94.2
224 46.914511   VMware_91:88:15  VMware_e6:1e:89  ARP      42 192.168.94.130 is at 00:0c:29:91:88:15
```

REMOTE CONTROL

We can also see a connection to DNS to get the IP address for the domain the user wanted to scan.

There is also an ARP protocol to get the mac address so the machine knows who to send the packets to.



| | | | | | |
|---|---|---|---|---|---|
| 3784 | 72.818101 | 192.168.94.130 | 192.30.45.30 | WHOIS | 71 Query: scanme.nmap.com |
| 3785 | 72.818743 | 192.30.45.30 | 192.168.94.130 | TCP | 60 43 → 39162 [ACK] Seq=1 Ack=18 Win=64240 Len=0 |
| 3786 | 72.819649 | 45.33.32.156 | 192.168.94.130 | TCP | 60 6566 → 54664 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0 |
| 3787 | 72.819658 | 45.33.32.156 | 192.168.94.130 | TCP | 60 3322 → 42614 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0 |
| 3788 | 72.835951 | 45.33.32.156 | 192.168.94.130 | TCP | 60 8200 → 46644 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0 |
| 3789 | 72.835977 | 45.33.32.156 | 192.168.94.130 | TCP | 60 3013 → 40786 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0 |
| 3790 | 72.835979 | 45.33.32.156 | 192.168.94.130 | TCP | 60 8654 → 44096 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0 |
| 3791 | 72.913196 | 192.30.45.30 | 192.168.94.130 | TCP | 1514 43 → 39162 [ACK] Seq=1 Ack=18 Win=64240 Len=1460 [TCP segment of a rea |
| 3792 | 72.913217 | 192.168.94.130 | 192.30.45.30 | TCP | 54 39162 → 43 [ACK] Seq=18 Ack=1461 Win=62780 Len=0 |
| 3793 | 72.913265 | 192.30.45.30 | 192.168.94.130 | WHOIS | 870 Answer: scanme.nmap.com |
| 3794 | 72.913357 | 192.168.94.130 | 192.30.45.30 | TCP | 54 39162 → 43 [FIN, ACK] Seq=18 Ack=2278 Win=62780 Len=0 |
| 3795 | 72.913649 | 192.30.45.30 | 192.168.94.130 | TCP | 60 43 → 39162 [ACK] Seq=2278 Ack=19 Win=64239 Len=0 |
| 3796 | 72.914243 | 192.168.94.130 | 192.168.94.129 | SSHv2 | 158 Server: |

It also shows the whois and domain name that was scanned.

## Discussion

The notes that needed to be in the script were added with a # symbol so it will not be run by the script. These notes will help anyone understand what is happening throughout the rest of the script.

```
53  # I listed the things the user will need so they can get it ready now instead of at every prompt
54
55  # The packages I want to check is installed in the user machine is in this list. The loop will check if each package is instal
```

The beginning of the script explains what service the script will do and what information the user needs to prepare.

```
45  echo
46  echo "Thanks for using this automated service to connect to a remote server and scan your desired domain/url."
47  sleep 1
48  echo "This service will also install the missing tools needed to run."
49  sleep 1
50  echo "There are a few things you will need to submit when prompted."
51  sleep 1
52  echo "Please prepare an IP address for the remote server, username and password to access the remote server and lastly a domain/url you would
53  # I listed the things the user will need so they can get it ready now instead of at every prompt
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo bash NRProj.sh
[sudo] password for kali:
Welcome!

Thanks for using this automated service to connect to a remote server and scan your desired domain/url.

This service will also install the missing tools needed to run.

There are a few things you will need to submit when prompted.

Please prepare the IP address of the remote server, credentials to access the remote server and lastly a domain/u
rl you would like to be scanned.
```

The script will check if the needed packages are installed in the machine using the dpkg command.

The function is_installed was created so the script can check through the list of packages and check and install the missing packages. The script has a for loop and if/else statement to determine the correct course of action. If the package is installed, it will let the user know that it is already installed. If the package is not installed, it will let the user know it is not installed and proceed to install it.

REMOTE CONTROL

```
# The packages I want to check is installed in the user machine is in this list.
# The loop will check if each package is installed and install them if needed.
PACKAGES=("nmap" "curl" "whois" "sshpass" "tor" "geoip-bin" "rsyslog")

is_installed() {
    dpkg -s "$1" &>/dev/null
}

# Loop through the list of packages
for package in "${PACKAGES[@]}"; do
    if is_installed "$package"; then
        echo "$package is already installed."
    else
        echo "$package is not installed. Installing... This may take some time"
        sudo apt update > /dev/null 2>&1
        sudo apt install -y "$package" > /dev/null 2>&1
    fi
done

echo "Package check and installation complete."
```

```
nmap is already installed.
curl is already installed.
whois is already installed.
sshpass is not installed. Installing... This may take some time
tor is not installed. Installing... This may take some time
geoip-bin is not installed. Installing... This may take some time
rsyslog is not installed. Installing... This may take some time
Package check and installation complete.
```

We used sudo apt-get install to download the missing packages as this is the best way to install packages.

Before installing, the machine needs to determine the location to install from so the script will run sudo apt-get update before installation.

```
# The next component is nipe. nipe is not in built in kali and it is not a package so it needs to be downloaded
echo
echo Cloning nipe... This may take some time
git clone https://github.com/htrgouvea/nipe > /dev/null 2>&1
cd nipe
sudo apt-get install -y cpanminus > /dev/null 2>&1

cpanm --installdeps . > /dev/null 2>&1

$yes | sudo cpan install Switch JSON LWP::UserAgent Config::Simple > /dev/null 2>&1

sudo perl nipe.pl install > /dev/null 2>&1
sudo perl nipe.pl start > /dev/null 2>&1
echo
echo
echo
```

```
Cloning nipe... This may take some time
```

REMOTE CONTROL

We are assuming nipe is not installed in the machine and so proceeds to install it to the machine.

As nipe is not a package, it cannot be downloaded in the previous step with sudo apt-get install.

It has to be cloned from github using the command git clone and followed by the location.

Then perl needs to be installed in the nipe folder so the script also cd into nipe.

After installing we use sudo perl nipe.pl start to establish a connection.

```
 96  echo
 97  while true; do
 98      ip_check=$(sudo perl nipe.pl status | wc -l)
 99      # this checks that the connection is anonymous before moving on to the next step
100      if [ $ip_check -eq 4 ]
101          then
102          spoof_ip=$(sudo perl nipe.pl status | grep Ip | awk '{print $3}')
103              echo You are anonymous...this is your spoofed IP address : "$spoof_ip"
104              location_ip=$(geoiplookup "$spoof_ip"| awk '{print $5,$6}')
105              echo The location is : "$location_ip"
106              break
107
108          else
109          echo The connection is not anonymous. Retrying...
110          sleep 3
111          echo
112          sudo perl nipe.pl restart
113  fi
114  done
115
```

```
The connection is not anonymous. Retrying...

The connection is not anonymous. Retrying...

You are anonymous...this is your spoofed IP address : 192.42.116.210
The location is : Netherlands
```

If/else statement was used in the script to determine if the script should keep running or needs to be terminated.

As the connection needs to be anonymous, the script will wait 3 seconds then restart the connection if the connection is not anonymous and informs the user as the connection needs to be anonymous to continue. This was added to a while loop as we need the script to continue after making an anonymous connection.

REMOTE CONTROL

The spoofed ip address needed to be shown so the syntax sudo perl nipe.pl status | grep Ip | awk '{print $3}' will only print out the IP address. This syntax was put into the variable spoof_ip and the variable was called in the echo.

The location of this spoofed IP address also needs to be shown so the syntax geoiplookip "$spoof_ip" | awk '{print $5,$6}' was used to show the country. As some countries print out with 2 words, the script will print both column 5 and 6. This syntax was put into the variable location_ip and the variable was called in the echo.

```
You are anonymous...this is your spoofed IP address : 45.148.10.111
The location is : Netherlands


Please provide a domain/url
scanme.nmap.com
```

Next we needed the domain/url to scan the remote server IP address and login details from the user so it was added to the script.

```
112
113   # This section ask user for the needed variables to access the remote server
114   echo
115   echo
116   echo Please provide a domain/url
117   read domain
118   echo
119   echo Please provide the remote server
120   read server
121   echo Please provide the username
122   read username
123   echo 'Please provide the password (input is hidden, press enter when done)'
124   read -s password
125   echo
126   echo
127   echo
128
```

The -s flag hides the user input from displaying on the terminal so there is at least some form of privacy.

```
111   echo 'Please provide the password (input is hidden, press enter when done)'
112   read -s password
113
```

REMOTE CONTROL

The information was also relayed to the user fearing they would get confused if they did not see any keystrokes displayed on the terminal.

```
Please provide a domain/url
scanme2.nmap.com

Please provide the remote server
[redacted]

Please provide the username
[redacted]

Please provide the password (input is hidden, press enter when done)
```

Next we needed to connect to the remote server.

We used sshpass to connect via ssh in a non-interactive way.

```
133  # Here the script will access a server remotely using the inputs from the user as the variables
134
135
136  function sshpass_command()
137  {
138      sshpass -p "$password" ssh -o StrictHostKeyChecking=no -T "$username@$server" "$1"
139  }
140
141  # Run Nmap scan and save results to a file
142  echo Scanning $domain using nmap...
143  sshpass_command "nmap $domain -oN nmap_file" > /dev/null 2>&1
144  echo Nmap scan results saved to nmap_file on remote server.
145
146
147  # Run whois scan and save results to a file
148  echo Scanning $domain using whois...
149  sshpass_command "whois $domain > whois_file" > /dev/null 2>&1
150  echo Whois scan results saved to whois_file on remote server.
151
152  echo
153
```

```
Scanning scanme2.nmap.com using nmap...
Nmap scan results saved to nmap_file on remote server.
Scanning scanme2.nmap.com using whois...
Whois scan results saved to whois_file on remote server.
```

Once connected the script will scan the domain given by the user and export the output to a file on the remote server. The 2 scans done on the remote server are nmap and whois. The flag -oN for nmap command will save the output to a file.

REMOTE CONTROL

```
└─$ cat nmap_file
# Nmap 7.80 scan initiated Sun Nov 24 15:25:03 2024 as: nmap -oN nmap_file scanme.nmap.com
Nmap scan report for scanme.nmap.com (45.33.32.156)
Host is up (0.20s latency).
Other addresses for scanme.nmap.com (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org
Not shown: 996 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
9929/tcp  open  nping-echo
31337/tcp open  Elite

# Nmap done at Sun Nov 24 15:25:27 2024 -- 1 IP address (1 host up) scanned in 24.06 seconds
```

Output from the remote server shows the scan was done and saved successfully

Next the file needs to be downloaded from the remote server to the user machine.

```
156
157   # Following step is to download the files from the remote server to the user machine
158   sshpass -p "$password" sftp -oBatchMode=no -b - "$username@$server" <<EOF
159   get /home/"$username"/nmap_file .
160   get /home/"$username"/whois_file .
161   bye
162   EOF
163   echo Nmap scan results in the file called nmap_file has been downloaded to your machine.
164   echo Whois scan results in the file called whois_file has been downloaded to your machine.
165
```

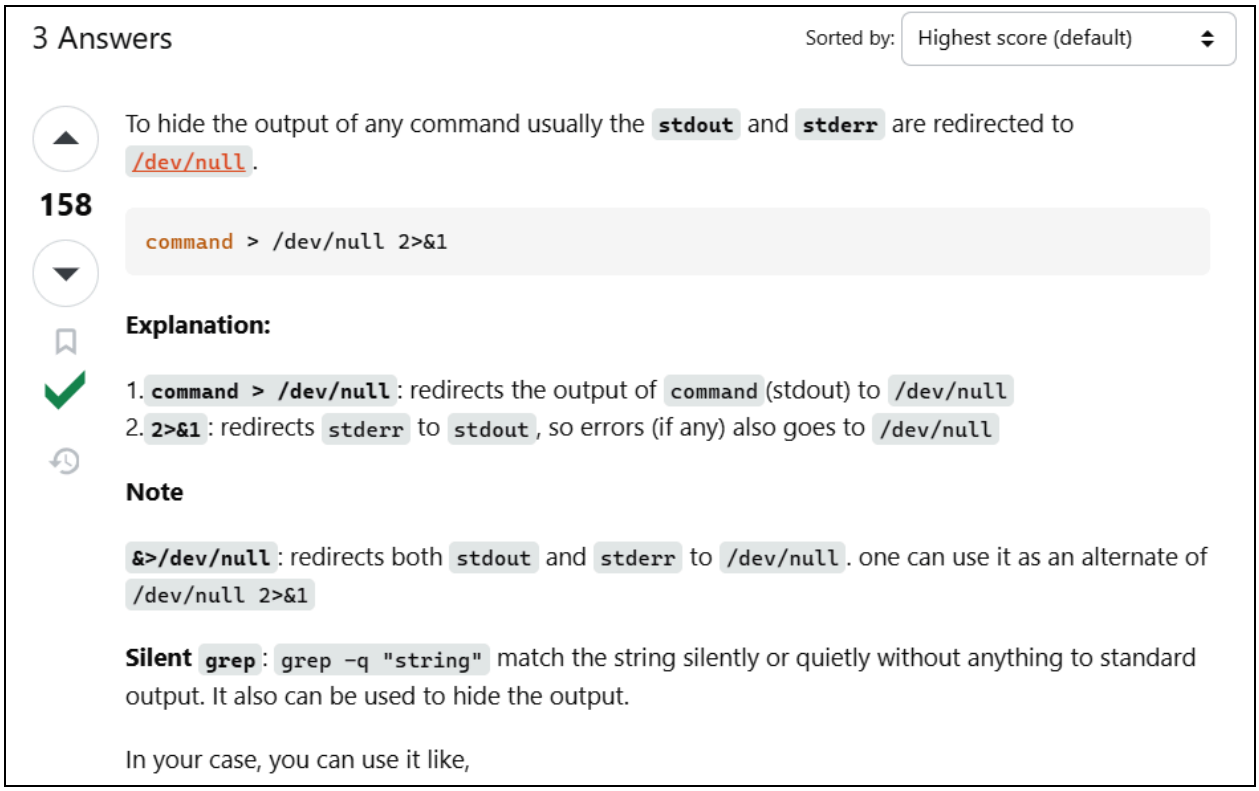
We used sshpass and sftp to securely copy the files.

```
┌──(kali㉿kali)-[~/Desktop/nipe]
└─$ ls
cpanfile  Dockerfile  lib  LICENSE.md  nipe.pl  nmap_file  README.md  SECURITY.md  tests  whois_file
```

I saw that the output will show the sftp lines and search for ways to avoid displaying them in the output.

```
Nmap scan results saved to nmap_file on remote server.
Scanning scanme.nmap.com using whois...
Whois scan results saved to whois_file on remote server.


sftp> get /home/"tc"/nmap_file .
sftp> get /home/"tc"/whois_file .
sftp> bye
Nmap scan results in the file called nmap_file has been downloaded to your machine.
Whois scan results in the file called whois_file has been downloaded to your machine.
```



3 Answers

Sorted by: Highest score (default)

158

To hide the output of any command usually the **stdout** and **stderr** are redirected to /dev/null.

```
command > /dev/null 2>&1
```

**Explanation:**

1. **command > /dev/null**: redirects the output of command (stdout) to /dev/null
2. **2>&1**: redirects stderr to stdout, so errors (if any) also goes to /dev/null

**Note**

**&>/dev/null**: redirects both stdout and stderr to /dev/null. one can use it as an alternate of /dev/null 2>&1

**Silent grep**: grep -q "string" match the string silently or quietly without anything to standard output. It also can be used to hide the output.

In your case, you can use it like,

Figure 23: "Hiding Output of a Command." (Ask Ubuntu,2024)

```
Scanning scanme.nmap.com using whois...
Whois scan results saved to whois_file on remote server.


Nmap scan results in the file called nmap_file has been downloaded to your machine.
Whois scan results in the file called whois_file has been downloaded to your machine.
```

I tested the syntax out and saw that it worked and added to every part of the script that I did not need to show the user.

Lastly, whatever the user did has to be logged. The command logger was used.

REMOTE CONTROL

```
161  function log_scan() {
162      logger -t logforscan "$1"
163  }
164
165  log_scan "Scans were executed on : [$(date "+%Y-%m-%d %H:%M:%S")]. "
166  log_scan "[$(date "+%Y-%m-%d %H:%M:%S")] '$domain' was scanned"
167  log_scan "[$(date "+%Y-%m-%d %H:%M:%S")] Nmap data collected "
168  log_scan "[$(date "+%Y-%m-%d %H:%M:%S")] Whois data collected "
169  echo "[$(date "+%Y-%m-%d %H:%M:%S")] - Nmap data collected for $domain "
170  echo "[$(date "+%Y-%m-%d %H:%M:%S")] - Whois data collected for $domain "
171  echo "The location of log is: /var/log/user.log. (For some machines it can be saved at /var/log/messages or /var/log/syslog)"
172  echo The logs have been tagged with logforscan to make it easier to pull up when needed.
173
174  echo
175  echo
176
```

```
Scans were logged on : [2024-11-26 10:02:06].
[2024-11-26 10:02:06] - Nmap data collected for scanme2.nmap.com
[2024-11-26 10:02:06] - Whois data collected for scanme2.nmap.com
The location of log is: /var/log/syslog. (For some machines it can be saved at /var/log/messages or /var/log/user
.log)
The logs have been tagged with logforscan to make it easier to pull up when needed.


Thank you for using this service.
Goodbye!
```

The logger command was made into a function as it will be used multiple times. By default, logger writes to /var/log/messages. However for some machines it can write to /var/log/syslog or /var/log/user.log. The -t flag will tag the entries with the given tag. This flag was added so the entries will be easy to find for future use. In the future, the user can grep for the tagged word to find the entries when needed.

```
┌──(kali㉿kali)-[/var/log]
└─$ cat user.log | grep logforscan
2024-11-23T22:04:18.712838-05:00 kali logforscan: Scans were executed on : [2024-11-23 22:04:18].
2024-11-23T22:04:18.720494-05:00 kali logforscan: [2024-11-23 22:04:18] '' was scanned
2024-11-23T22:04:18.726127-05:00 kali logforscan: [2024-11-23 22:04:18] Nmap data collected
2024-11-23T22:04:18.730702-05:00 kali logforscan: [2024-11-23 22:04:18] Whois data collected
```

The script was cleaned up and notes were added so anyone will be able to read the script and use it immediately.

The full output of the script run on a ‘fresh’ kali is shown in the Appendix.

REMOTE CONTROL

Files can be transferred from one machine to another using different protocols.

FTP or file transfer protocol is one of the most common ways of transferring files over the internet.

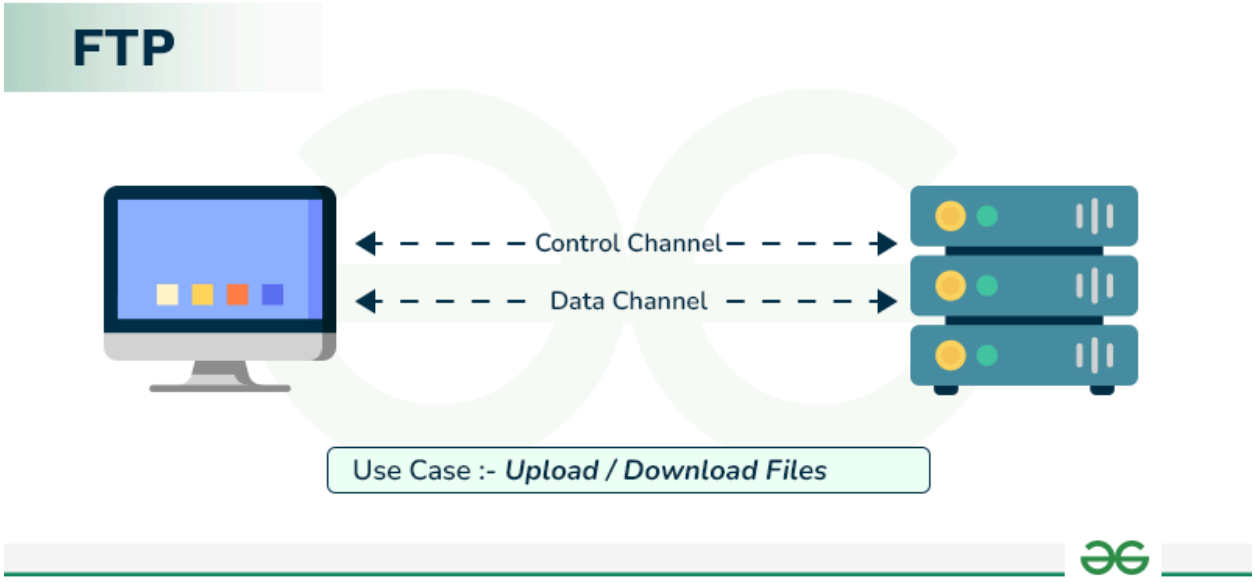



Figure 24: “File Transfer Protocol (FTP) in Application Layer” (GeeksforGeeks,2024)

There are many uses for FTP. FTP is used for many different reasons such as transferring large files or large quantities of files. FTP can be automated to help handle tasks more efficiently. FTP transfers files reliably even if the machines are different operating systems. FTP uses TCP as a transport layer protocol.

We can use an FTP client to connect to a server. An example of this is FileXilla. It has graphical user interface which makes it easy to use.

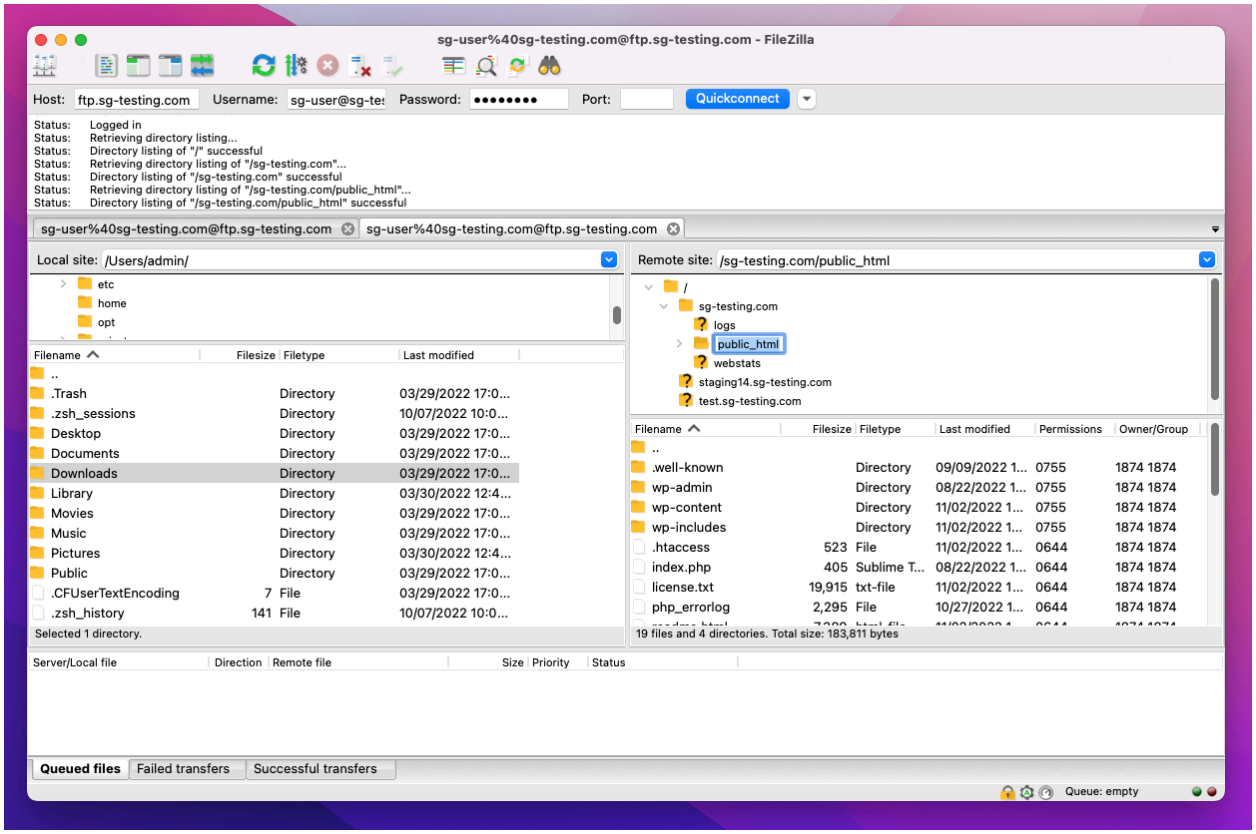



Figure 25: “What is FTP.” (SiteGround, 2024)

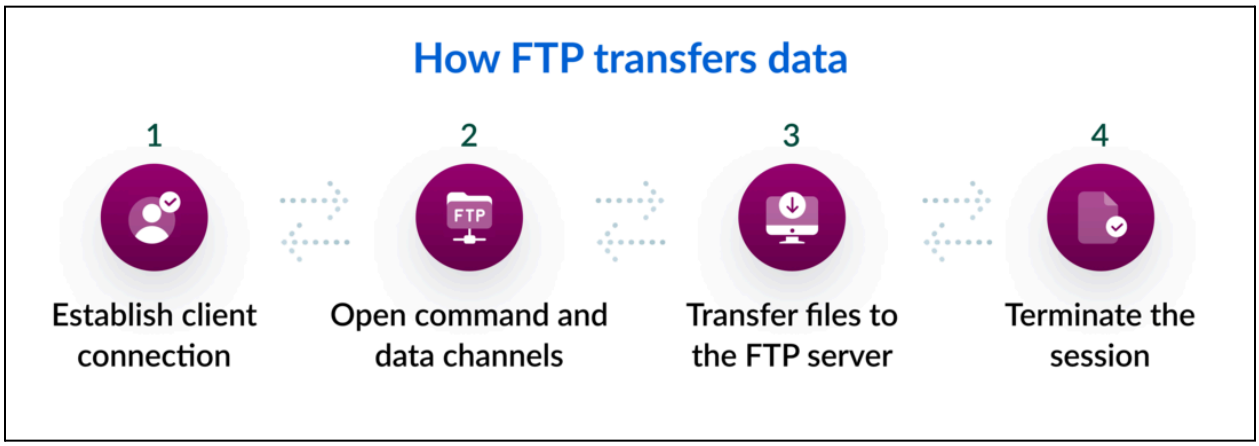



Figure 26: “What Is File Transfer Protocol (FTP)?” (Box,2024)

FTP has 4 main steps. Using a FTP client, it will first establish a connection with the FTP server.

Then it opens 2 channels, one for sending commands between the client and the server and the data channel transfers the file data. Next it transfers the file to the FTP server. Lastly it terminates the connection so it does not continue to send data.

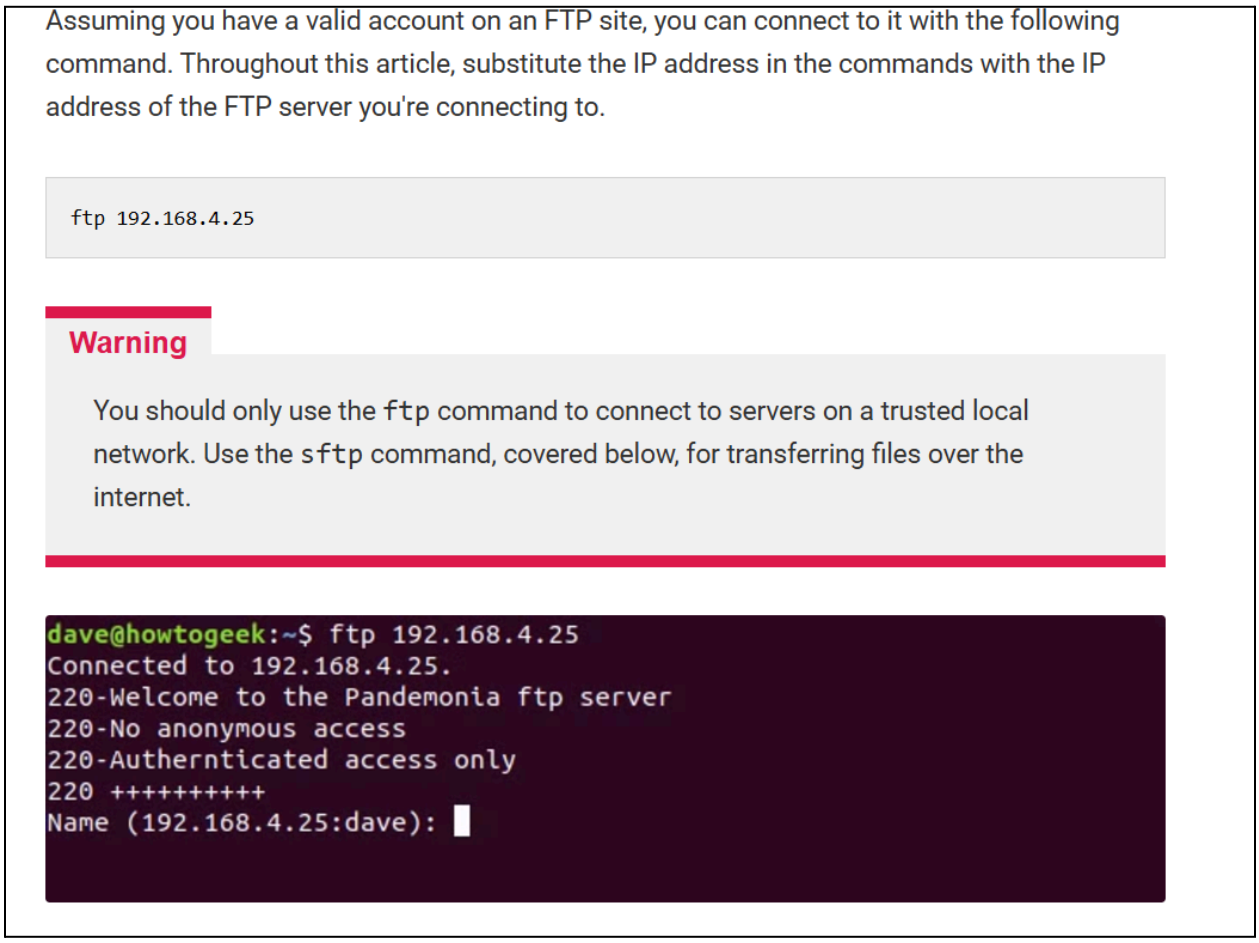


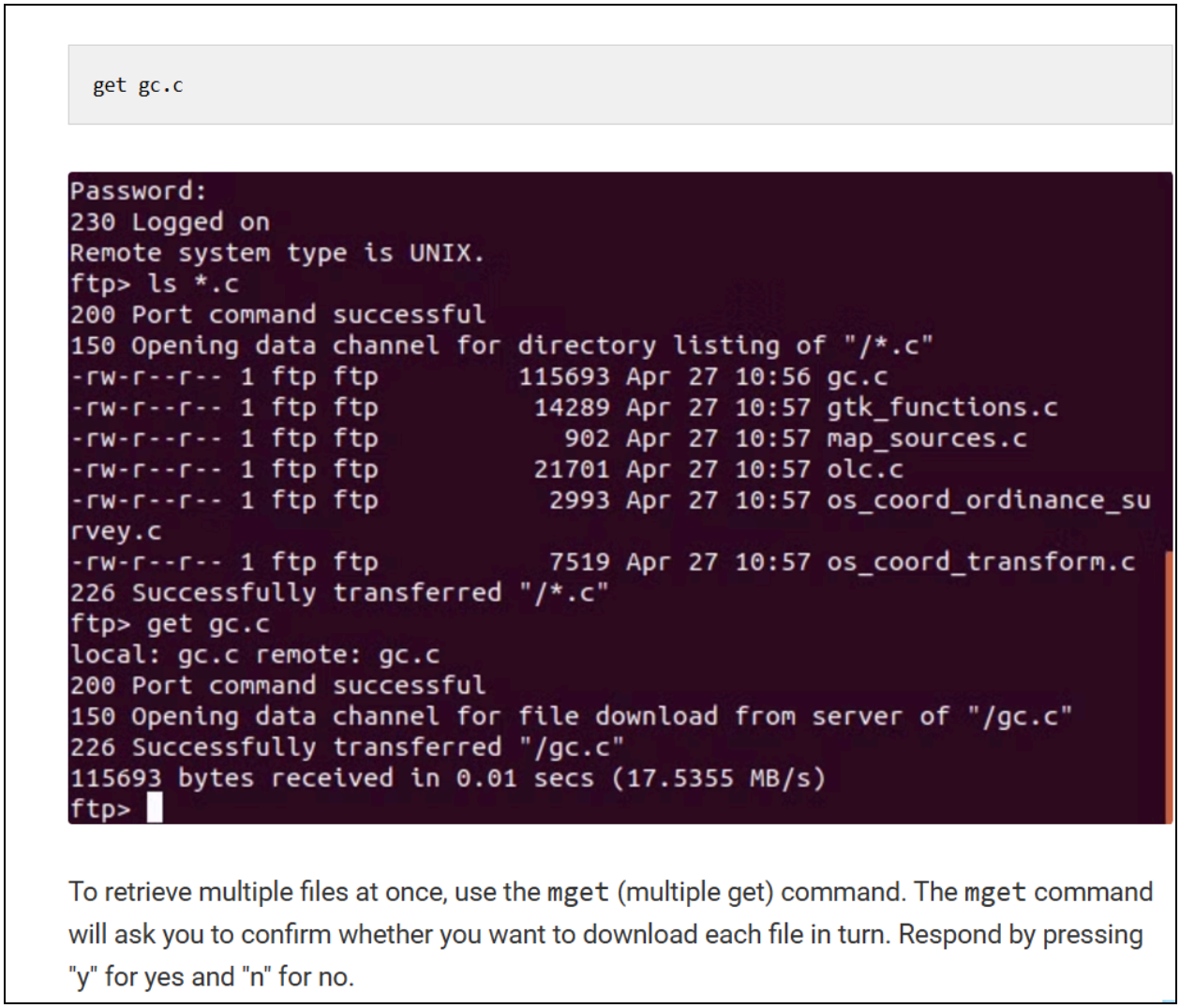

Figure 27: “How to Use the FTP Command on Linux.” (How-to Geek,2024)

FTP can also be used from the command prompt. The syntax is ftp <ip address>.

Lastly, you can also use a browser to access and download files.

FTP can work in passive mode or active mode.

FTP uses port 20 and 21.



```
get gc.c
```

```
Password:
230 Logged on
Remote system type is UNIX.
ftp> ls *.c
200 Port command successful
150 Opening data channel for directory listing of "/*.c"
-rw-r--r-- 1 ftp ftp         115693 Apr 27 10:56 gc.c
-rw-r--r-- 1 ftp ftp          14289 Apr 27 10:57 gtk_functions.c
-rw-r--r-- 1 ftp ftp            902 Apr 27 10:57 map_sources.c
-rw-r--r-- 1 ftp ftp          21701 Apr 27 10:57 olc.c
-rw-r--r-- 1 ftp ftp           2993 Apr 27 10:57 os_coord_ordinance_su
rvey.c
-rw-r--r-- 1 ftp ftp           7519 Apr 27 10:57 os_coord_transform.c
226 Successfully transferred "/*.c"
ftp> get gc.c
local: gc.c remote: gc.c
200 Port command successful
150 Opening data channel for file download from server of "/gc.c"
226 Successfully transferred "/gc.c"
115693 bytes received in 0.01 secs (17.5355 MB/s)
ftp>
```

To retrieve multiple files at once, use the `mget` (multiple get) command. The `mget` command will ask you to confirm whether you want to download each file in turn. Respond by pressing "y" for yes and "n" for no.

Figure 28: “How to Use the FTP Command on Linux.” (How-to Geek,2024)

To get files the syntax is get <filename> or use mget for multiple files.

FTP connection is not encrypted as it was not designed for this purpose. This is one of their main security flaws and hackers will have no issues getting the information used via a FTP connection. This does not follow the CIA triad as the confidentiality is not attained.

For a secure file transfer the protocol to be used would be SFTP.

SFTP would encrypt the data being sent over the internet and decrypt the data received. So even if anyone intercepts the data, they would not get access to the information in the files.

SFTP uses Secure Shell or SSH protocol to send and receive data over a secure channel. SFTP is supported by many Operating Systems like Mac, Linux and Windows.

SFTP is the best protocol to transfer files securely.

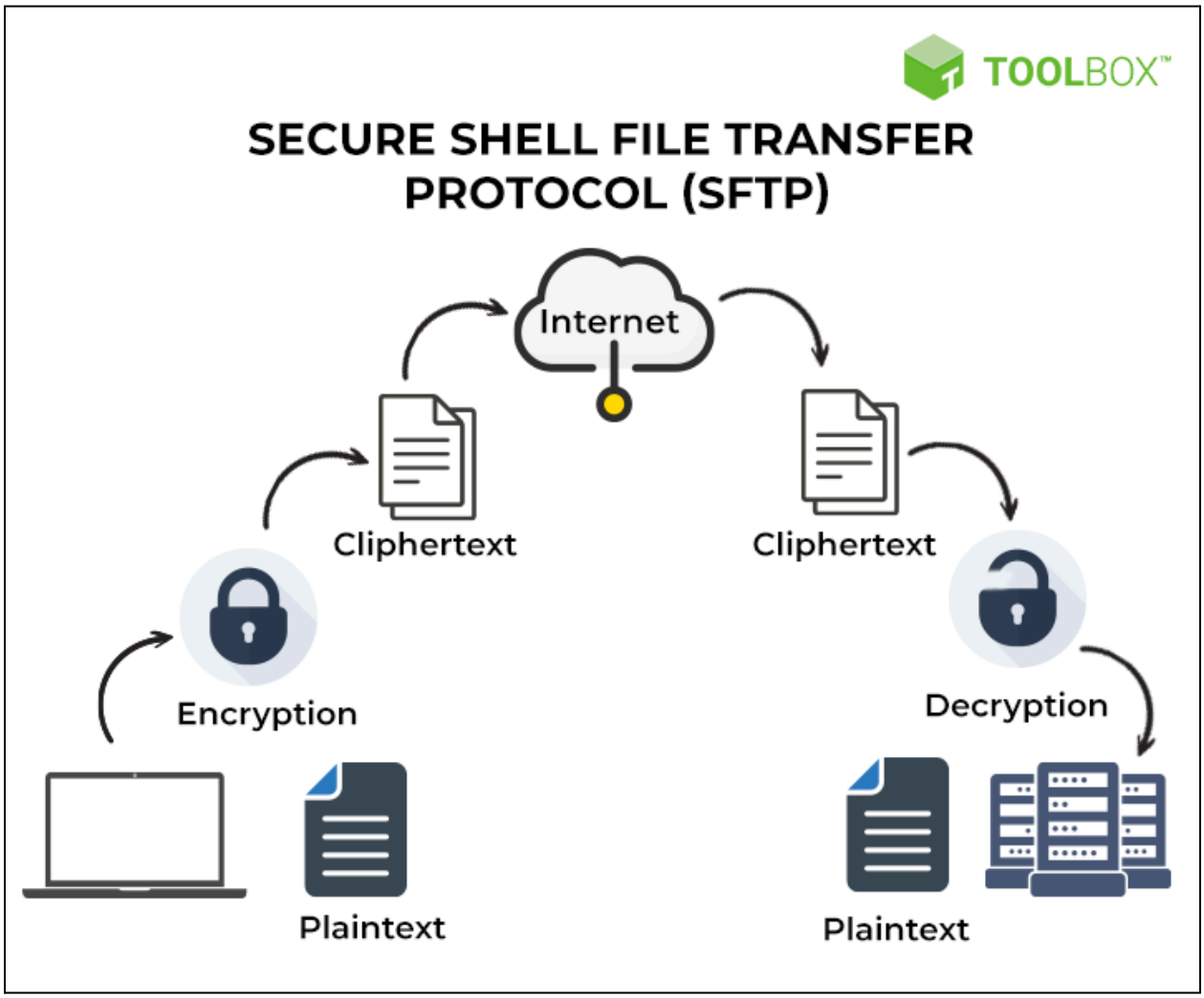
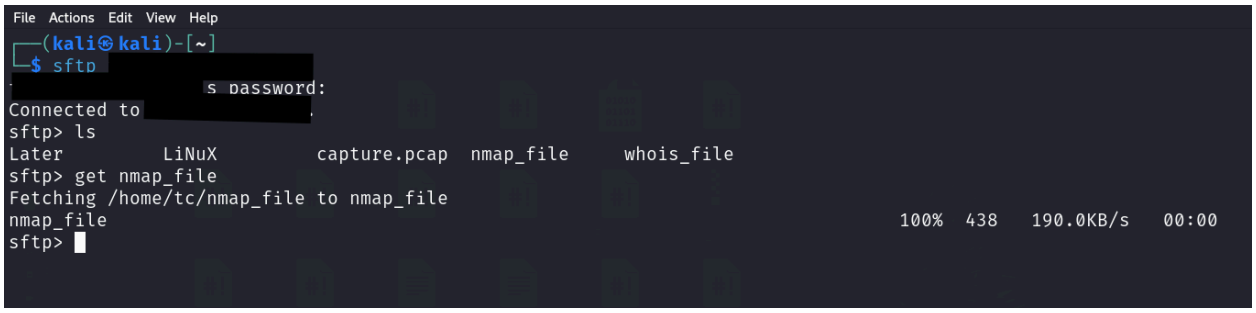



Figure 29: "SFTP vs. FTPS: Understanding the 8 Key Differences." (Spiceworks,2024)

The default port number for SFTP is 22.

REMOTE CONTROL

An example will be when we use SFTP to get a file from another machine:




This will securely allow us to download the file which satisfies the CIA triad as the data is encrypted, even if anyone intercepts the data, it is not compromised and it cannot be altered.

## Conclusion

This project was a massive task which helped our understanding of how important cybersecurity is. There are many tasks we do that can be automated which will greatly increase our efficiency. This will allow us to focus on more important task while the minor task can be automated. However in every task that we do, we need to ensure we use secure channels or we are not risking our information. There are also many protocols used when machines are establishing connections. We have learnt how to create an automated program to help us scan for domains without revealing our IP address and instead connect to a remote server to execute the commands, effectively masking our identity to get the same job done. We learn how some protocols are vulnerable to attacks and what secure alternatives we can use instead. We learn the importance of the CIA triad and how we need to ensure every task we do satisfies the CIA triad. Networking protocols are ingrained deep into the internet and we have only touched on one protocol, there is so much more for us to learn and understand and we will take it one step at a time.

## Recommendations

We have learnt how some protocols are vulnerable to attacks and we should be vigilant when we use the internet as we do not know if anyone is targeting us or if we are unknowingly giving our information away. As online communication will be a massive part of everyone's daily lives, our suggestion is that we should send or download files using an encrypted protocol or use services that have added security only.

REMOTE CONTROL

## References

*The Dpkg Command in Linux - a Beginners Reference | DigitalOcean*. www.digitalocean.com/community/tutorials/dpkg-command-in-linux.

"How to bash script to check and download missing packages using dpkg" prompt. *ChatGPT*, GPT-4, OpenAI, 21 Nov 2024, chatgpt.com

deb. "How to Install a Deb File, by Dpkg -i or by Apt?" *Unix & Linux Stack Exchange*, 3 Oct. 2014, unix.stackexchange.com/questions/159094/how-to-install-a-deb-file-by-dpkg-i-or-by-apt. Accessed 24 Nov. 2024.

GeeksforGeeks. "How to Install Nipe Tool in Kali Linux?" *GeeksforGeeks*, 18 July 2021, www.geeksforgeeks.org/how-to-install-nipe-tool-in-kali-linux/. Accessed 24 Nov. 2024.

"Hiding User Input on Terminal in Linux Script." *Stack Overflow*, 30 Nov. 2010, stackoverflow.com/questions/4316730/hiding-user-input-on-terminal-in-linux-script.

in. "Automatically Enter Input in Command Line." *Ask Ubuntu*, 29 Aug. 2013, askubuntu.com/questions/338857/automatically-enter-input-in-command-line.

rcsimoes. "How to Put Sshpass Command inside a Bash Script?" Stack Overflow, 10 Oct. 2013, stackoverflow.com/questions/19302572/how-to-put-sshpass-command-inside-a-bash-script.

"SSH Password Automation in Linux with Sshpass." *Redhat.com*, 2020, www.redhat.com/en/blog/ssh-automation-sshpass. Accessed 24 Nov. 2024.

in. "How to Disable Strict Host Key Checking in Ssh?" *Ask Ubuntu*, 13 Dec. 2011, askubuntu.com/questions/87449/how-to-disable-strict-host-key-checking-in-ssh. Accessed 25 Nov. 2024.

anubhava. “How to Run the Sftp Command with a Password from Bash Script?” *Stack Overflow*, 22 Mar. 2011, stackoverflow.com/questions/5386482/how-to-run-the-sftp-command-with-a-password-from-bash-script.

“Hiding Output of a Command.” *Ask Ubuntu*, askubuntu.com/questions/474556/hiding-output-of-a-command.

University, Geek. “Create a Log Entry | Linux#.” *Geek University*, geek-university.com/create-a-log-entry/.

“Tcpdump Command in Linux with Examples.” *GeeksforGeeks*, 22 May 2020, www.geeksforgeeks.org/tcpdump-command-in-linux-with-examples/.

GeeksforGeeks. “File Transfer Protocol (FTP) in Application Layer - GeeksforGeeks.” *GeeksforGeeks*, 25 Sept. 2017, www.geeksforgeeks.org/file-transfer-protocol-ftp-in-application-layer/.

SiteGround Web Hosting. “SiteGround.” *SiteGround*, 2024, world.siteground.com/kb/what-is-ftp/. Accessed 26 Nov. 2024.

“What Is File Transfer Protocol (FTP)? | Box.” *Box*, 19 Mar. 2024, www.box.com/resources/what-is-ftp. Accessed 25 Nov. 2024.

Postel, J., and J. Reynolds. *File Transfer Protocol*. Oct. 1985, https://doi.org/10.17487/rfc0959.

McKay, Dave. “How to Use the FTP Command on Linux.” *How-to Geek*, www.howtogeek.com/412626/how-to-use-the-ftp-command-on-linux/.

BasuMallick, Chiradeep. “SFTP vs. FTPS: Understanding the 8 Key Differences.” *Spiceworks*, 22 Apr. 2022, www.spiceworks.com/tech/networking/articles/sftp-vs-ftps/.

REMOTE CONTROL

## Appendix - Full output of the script run on a ‘fresh’ kali

```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo bash NRProj.sh
[sudo] password for kali:
Welcome!

Thanks for using this automated service to connect to a remote server and scan your desired domain/url.

This service will also install the missing tools needed to run.

There are a few things you will need to submit when prompted.

Please prepare the IP address of the remote server, credentials to access the remote server and lastly a domain/u
rl you would like to be scanned.


nmap is already installed.
curl is already installed.
whois is already installed.
sshpass is not installed. Installing... This may take some time
tor is not installed. Installing... This may take some time
geoip-bin is not installed. Installing... This may take some time
rsyslog is not installed. Installing... This may take some time
Package check and installation complete.

Cloning nipe... This may take some time



The connection is not anonymous. Retrying...
```

```
The connection is not anonymous. Retrying...

You are anonymous... this is your spoofed IP address : 192.42.116.210
The location is : Netherlands


Please provide a domain/url
scanme2.nmap.com

Please provide the remote server
[redacted]

Please provide the username
[redacted]

Please provide the password (input is hidden, press enter when done)

Scanning scanme2.nmap.com using nmap...
Nmap scan results saved to nmap_file on remote server.
Scanning scanme2.nmap.com using whois...
Whois scan results saved to whois_file on remote server.

Nmap scan results in the file called nmap_file has been downloaded to your machine.
Whois scan results in the file called whois_file has been downloaded to your machine.

Scans were logged on : [2024-11-26 10:02:06].
[2024-11-26 10:02:06] - Nmap data collected for scanme2.nmap.com
[2024-11-26 10:02:06] - Whois data collected for scanme2.nmap.com
The location of log is: /var/log/syslog. (For some machines it can be saved at /var/log/messages or /var/log/user
.log)
The logs have been tagged with logforscan to make it easier to pull up when needed.


Thank you for using this service.
Goodbye!
```

```
tc@server:~$ ls
capture.pcap  Later  LiNuX  nmap_file  whois_file
tc@server:~$ cat nmap_file
# Nmap 7.80 scan initiated Mon Nov 25 07:27:05 2024 as: nmap -oN nmap_file scanme.nmap.com
Nmap scan report for scanme.nmap.com (45.33.32.156)
Host is up (0.20s latency).
Other addresses for scanme.nmap.com (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org
Not shown: 996 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
9929/tcp  open  nping-echo
31337/tcp open  Elite

# Nmap done at Mon Nov 25 07:27:29 2024 -- 1 IP address (1 host up) scanned in 23.87 seconds
tc@server:~$ _
```

```
┌──(kali㉿kali)-[~/Desktop/nipe]
└─$ ls
cpanfile  Dockerfile  lib  LICENSE.md  nipe.pl  nmap_file  README.md  SECURITY.md  tests  whois_file
```

```
┌──(kali㉿kali)-[/var/log]
└─$ cat syslog | grep logforscan
2024-11-25T05:31:04.154658-05:00 kali logforscan: Scans were executed on : [2024-11-25 05:31:04].
2024-11-25T05:31:04.157118-05:00 kali logforscan: [2024-11-25 05:31:04] 'scanme.nmap.com' was scanned
2024-11-25T05:31:04.160436-05:00 kali logforscan: [2024-11-25 05:31:04] Nmap data collected
2024-11-25T05:31:04.163469-05:00 kali logforscan: [2024-11-25 05:31:04] Whois data collected
```