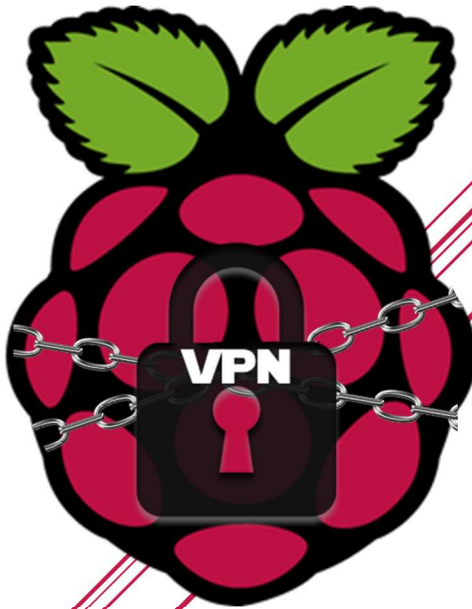


PROYECTO DE SEGURIDAD DE RED PRIVADA

CARLOS D. YAQUE JEFFS

15 DE JUNIO, 2020



CEU

CEU ANDALUCIA

CARLOS D. YAQUE JEFFS

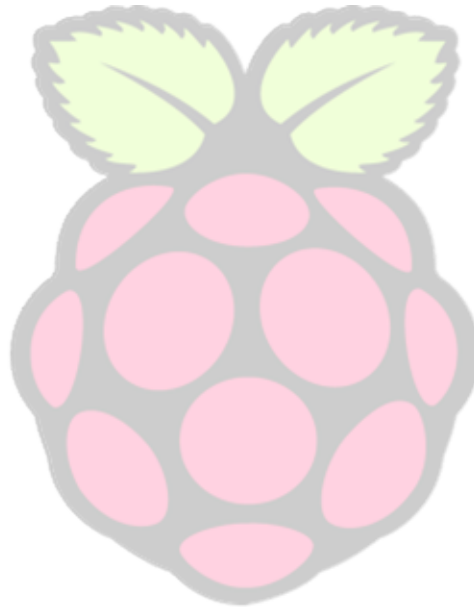
Índice de Contenidos

Índice de Contenidos	1
Justificación	2
Conceptos Básicos	3
➤ Software Usado:	3
➤ Diagrama de una Raspberry Pi 4	3
Marco Teórico	4
Objetivos del Proyecto	5
Metodología de investigación	6
Resultados y análisis	7
➤ Materiales necesarios:	7
➤ Instalación del Sistema Operativo:	8
➤ Preparación del Sistema Operativo:	10
➤ Instalación del primer servicio (Pi-Hole):	14
➤ Comprobación del servicio (Pi-Hole):	15
➤ Configuración del servicio (Pi-Hole):	17
➤ Instalación del segundo servicio (OpenVPN):	19
➤ Comprobación y Configuración del servicio (OpenVPN):	22
Conclusiones	25
➤ Pi-Hole:	25
➤ PiVPN (OpenVPN):	26
Referencias:	27

Justificación

El concepto principal del proyecto es conseguir montar un pequeño servidor de seguridad para cualquier persona particular y su red privada en casa o negocio. Esto se hará con un ordenador muy usado en la comunidad informática por su precio, utilidad, y tamaño. Este miniordenador se llama Raspberry Pi y el modelo 4 es el que vamos a usar.

Yo he elegido este tema por la importancia de la seguridad informática hoy en día. Desde nuestra vida personal, a la laboral, estamos rodeados de ordenadores, teléfonos móviles y dispositivos electrónicos que usan internet y en si están conectados a la red mundial. Por esto hay gente que usa esto para, robar, destruir, y arruinar las vidas y negocios de otros. Por estas razones pienso que este proyecto puede traer una fácil y barata forma de añadir otra barrera contra el cibercrimen.



Conceptos Básicos

➤ Software Usado:



PuTTY – para conexión Remota



UltraVNC – Para control remoto del escritorio

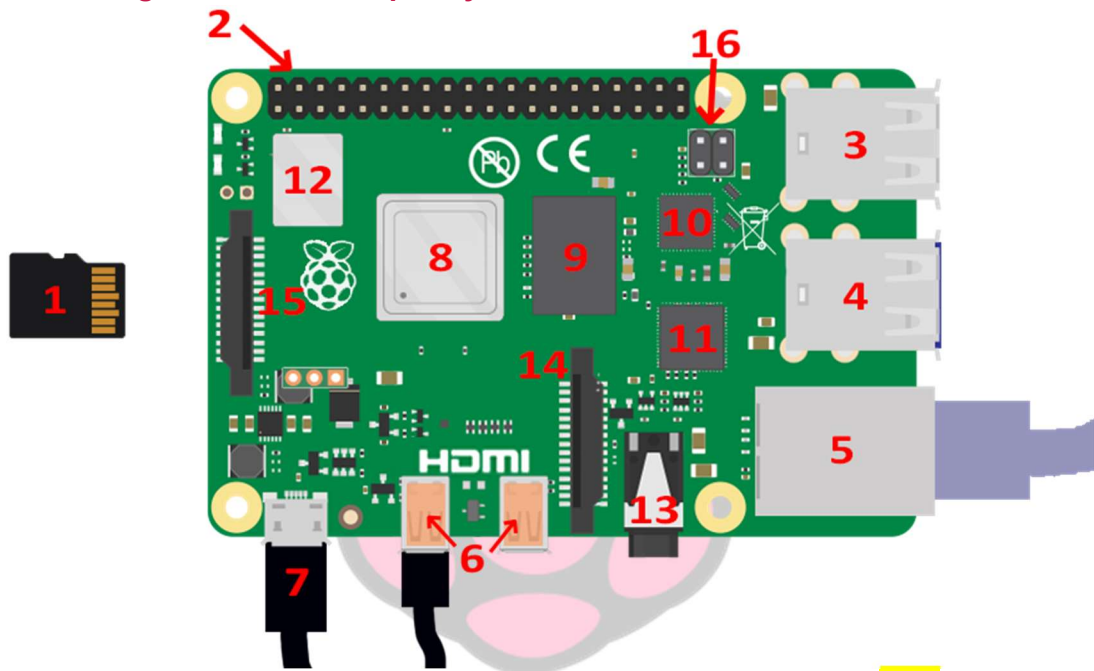


PI-Hole – Para crear el servidor Proxy/DNS y filtro de anuncios y páginas maliciosas.



OpenVPN – Para crear nuestro servidor VPN

➤ Diagrama de una Raspberry Pi 4



- | | |
|--|---|
| <ol style="list-style-type: none"> 1. Tarjeta SD (donde se instalará el sistema operativo) 2. Cabecera de 40 Pins 3. 2x Puertos USB 2.0 4. 2x Puertos USB 3.0 5. Puerto Gigabit Ethernet 6. 2x Puertos Micro-HDMI <ol style="list-style-type: none"> a. único = 4k60fps b. dual = 4k30fps 7. 5V@3A USB-C Entrada de alimentación 8. CPU 1.5GHz quad-core Cortex A72 | <ol style="list-style-type: none"> 9. LPDDR4 SDRAM [1GB,2GB,4GB] 10. Controlador de Ethernet 11. Controlador de USB 12. WIFI Dual Band (2.5GHz y 5GHz) y Bluetooth 5.0 13. Salida estéreo y puerto de video compuesto 14. Puerto de cámara CSI 15. DSI Display Port 16. Soporte PoE con PoE HAT |
|--|---|

Marco Teórico

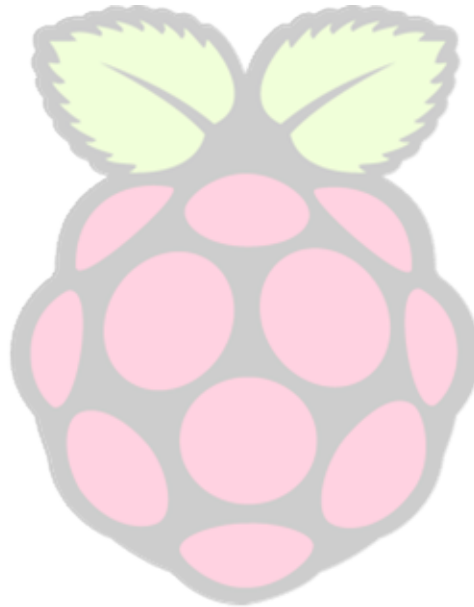
El marco teórico básico de mi proyecto es la seguridad y redes informáticas. Dos cosas que en el mundo de hoy todo el mundo debería de tener un básico conocimiento por la gran cantidad de tecnología que nos rodea. Los objetivos principales de mi proyecto en conjunto de este marco son:

- Proteger
 - Proteger a los usuarios con menos conocimiento de los peligros y riesgos del internet de páginas, y espionaje. Bloqueando paginas conocidas como maliciosas y peligrosas, y además prevenir rastreo de datos de empresas y corporaciones.
- Facilitar
 - Facilitar el acceso a recursos como poder conectarse a servicios y máquinas de su red privada desde cualquier sitio gracias a la VPN montada. También facilitando el acceso a una red segura para poder navegar páginas con información sensible sin riesgo de robo de datos.
- Asegurar/autenticar
 - Asegurando la seguridad de todos los usuarios y que toda información incluyendo contraseñas están protegidas y locales para no tener el riesgo de fuga de datos

Objetivos del Proyecto

Proteger a los usuarios locales será el objetivo principal del proyecto. Esto se hará con las siguientes medidas:

- Un tipo de proxy/firewall para prevenir y bloquear la posibilidad que usuarios con menos conocimiento informático no sean susceptibles a algún tipo de ataque o truco malicioso
- Tener una VPN que proteja la transmisión de datos entre los equipos y su destino por si hay alguna intrusión en la red privada.
- Y asegurar que esta todo bien montado para tenerlo siempre disponible con poco mantenimiento.



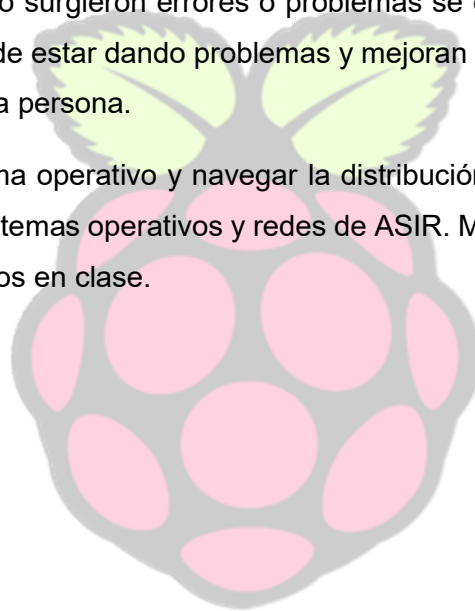
Metodología de investigación

La mayoría de la información usada para poder poner en efecto este proyecto viene de varios foros y paginas instruccionales oficiales del software y de comunidades informáticas con tutoriales y consejos para ayudar con este tipo de programas y sistemas operativos. Los sitios principales que yo use fueron:

- Reddit
- Foros de pi-hole
- Linus tech tips
- GitHub

También varios videos cuando surgieron errores o problemas se encontraron en YouTube que ayudan a visualizar lo que pude estar dando problemas y mejoran la habilidad de comprender lo que está intentando explicar la persona.

Al instalar configurar el sistema operativo y navegar la distribución de Linux, ese conocimiento vino de nuestras clases de sistemas operativos y redes de ASIR. Muchos comandos son iguales a los que usamos y aprendimos en clase.



Resultados y análisis

➤ Materiales necesarios:

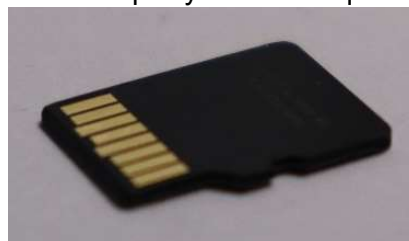
- Rasperry Pi (Modelo 4 con 1gb RAM usado en este manual)



- Cable Ethernet (Cat 5 o mejor preferible)



- Tarjeta SD de 8gb o más (para la Rasperry Pi 4 tiene que ser SD mini)



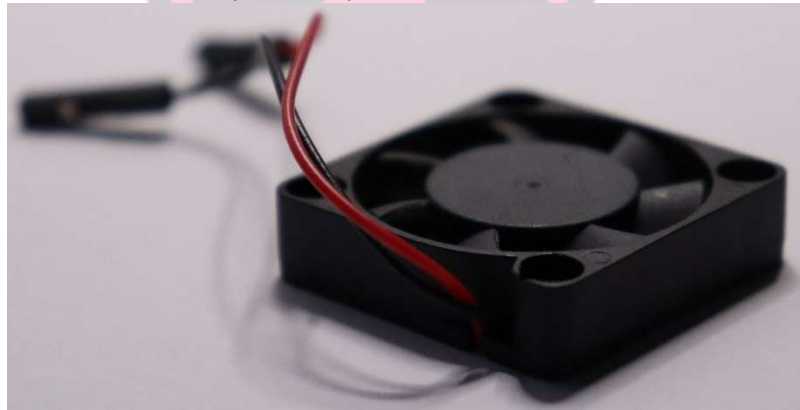
- Cable HDMI o micro HDMI para el modelo 4



- Monitor
- Caja para la Raspberry Pi (opcional)



- Ventilado o disipador de calor (opcional)

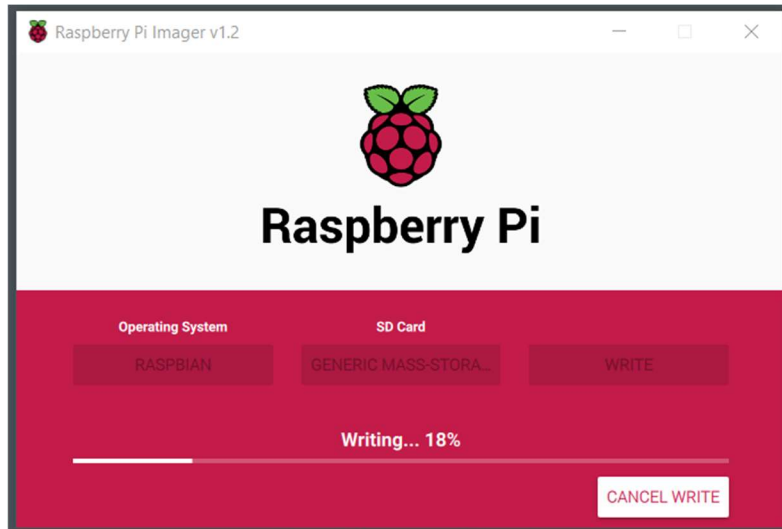


➤ **Instalación del Sistema Operativo:**

1. Lo primero que hay que hacer es como con cualquier sistema nuevo, elegir un sistema operativo para usar en nuestro proyecto. Para hacer lo más simple elegí usar Raspbian. Esta distribución de Linux es ligera y optimizada para ser usada en una Raspberry Pi's, aunque se pueden usar otras distribuciones ligeras. Esto se puede instalar directamente

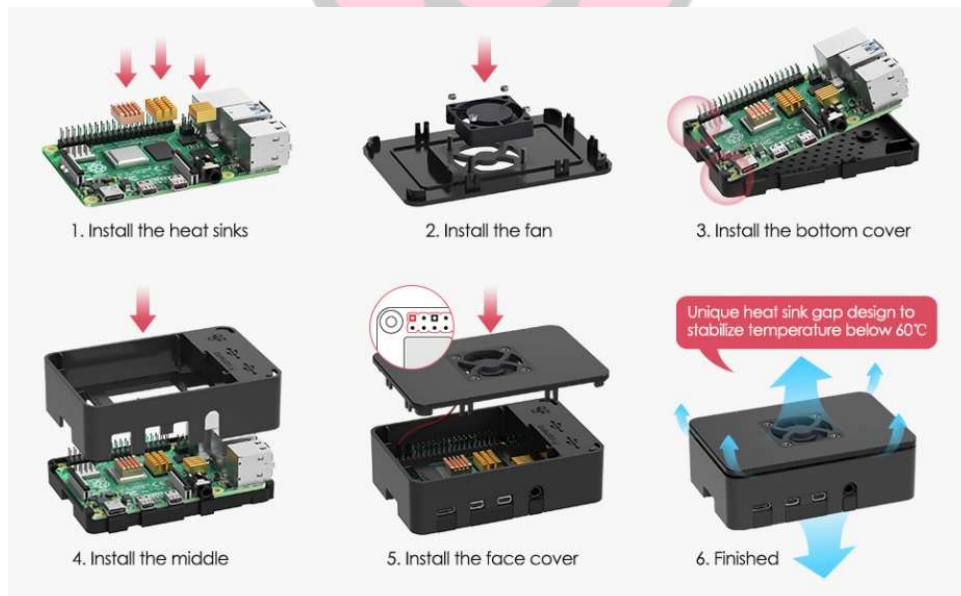
a la tarjeta SD o usar un Pen drive booteable para instalar el sistema directamente como cualquier sistema operativo.

- a. La Instalación se puede hacer de varios modos, con el software directamente descargable de: <https://www.raspberrypi.org/downloads/>



(Instalación de Raspberry Pi – Imagen 1)

- b. O con cualquier software para crear o instalar imágenes de sistemas.
2. Luego es importante que todos los componentes como el disipador, el ventilador y la Raspberry este todo montado correctamente y que todos los cables necesarios estén conectados.



(Ejemplo de montaje de funda Raspberry Pi – Imagen 2)

(Para la instalación inicial es necesario tener un ratón y teclado enchufado directamente a la Raspberry Pi.)

3. Cuando terminemos la Instalación de Raspbian tendremos un entorno grafico (o si elegimos no instalarlo tendremos una terminal). De aquí podemos empezar a instalar los servicios que harán posible usar la Pi remotamente sin deber de tener un teclado, ratón, y monitor conectado.

➤ Preparación del Sistema Operativo:

1. Lo primero que hay que instalar es el servicio SSH o “Secure Shell Protocolo”. Esto es un servicio que permite conectar servicios de red de forma segura a través de una red no segura. En nuestro caso vamos a usarlo para usar cualquier ordenador de nuestra red y conectarnos a la Raspberry Pi. Como vemos en la imagen 3 es muy posible que openssh (el software que vamos a usar para SSH) ya venga preinstalado. También con el comando `systemctl status ssh` podemos ver si el servicio está activo y corriendo o no.

```
pi@raspberrypi:~ $ sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:7.9p1-10+deb10u2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pi@raspberrypi:~ $
```

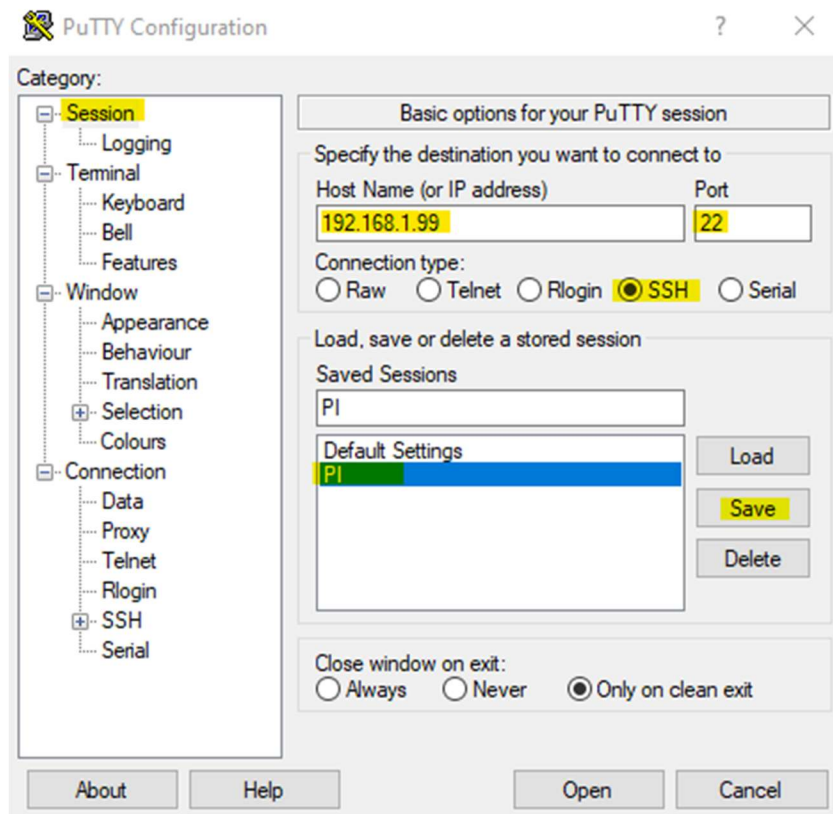
(Instalación de Servidor SSH – Imagen 3)

```
• ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Fri 2020-05-15 11:25:56 CEST; 1 weeks 2 days ago
    Docs: man:sshd(8)
          man:sshd_config(5)
 Main PID: 559 (sshd)
   Tasks: 1 (limit: 1599)
  Memory: 4.9M
 CGroup: /system.slice/ssh.service
         └─559 /usr/sbin/sshd -D

May 19 19:57:47 carlos-raspberrypi sshd[26075]: pam_unix(sshd:session): session opened for user pi by (uid=0)
May 19 23:23:19 carlos-raspberrypi sshd[6741]: Connection closed by 192.168.1.50 port 53901 [preauth]
May 21 13:20:24 carlos-raspberrypi sshd[4827]: Accepted password for pi from 192.168.1.50 port 58155 ssh2
May 21 13:20:24 carlos-raspberrypi sshd[4827]: pam_unix(sshd:session): session opened for user pi by (uid=0)
May 21 15:40:45 carlos-raspberrypi sshd[15233]: Accepted password for pi from 192.168.1.50 port 59406 ssh2
May 21 15:40:45 carlos-raspberrypi sshd[15233]: pam_unix(sshd:session): session opened for user pi by (uid=0)
May 22 10:50:56 carlos-raspberrypi sshd[16436]: Accepted password for pi from 192.168.1.50 port 64044 ssh2
May 22 10:50:56 carlos-raspberrypi sshd[16436]: pam_unix(sshd:session): session opened for user pi by (uid=0)
May 24 13:07:27 carlos-raspberrypi sshd[22422]: Accepted password for pi from 192.168.1.50 port 51163 ssh2
May 24 13:07:27 carlos-raspberrypi sshd[22422]: pam_unix(sshd:session): session opened for user pi by (uid=0)
```

(Comprobación de funcionamiento del servidor SSH– Imagen 4)

2. Ya que tenemos el servicio funcionando podemos desconectar el teclado, ratón y monitor y conectarnos por remoto. Para hacer esto desde un ordenador con Windows 10, vamos a necesitar un software llamado PuTTY. Este software usa el protocolo SSH (y varios más) para hacer conexiones de terminal remota.
3. Cuando abrimos PuTTY tenemos varias opciones presentadas para diferentes tipos de conexión. A nosotros solo no interesa la opción de "SSH". Como vemos en la imagen 5, usamos la IP de la Raspberry Pi. La IP en este momento será dinámica y hasta que lo pongamos estática.



(Panel de conexión SSH del programa PuTTY – Imagen 5)

4. Para entrar en la maquina usamos el usuario pi y la contraseña raspberry
5. En cuanto tengamos acceso al Raspberry Pi podemos entrar en /etc usando:

```
cd /etc
```

6. Y después podemos entrar dentro del archivo dhcpd.conf usando:

```
nano dhcpd.conf
```

7. Ahora aquí dentro podemos usar este archivo de configuración para ponerle una IP estática al “servidor”

```

GNU nano 3.2                                dhcpcd.conf

# define static profile
#profile static_eth0
#static ip_address=192.168.1.23/24
#static routers=192.168.1.1
#static domain_name_servers=192.168.1.1

# fallback to static profile on eth0
#interface eth0
#fallback static_eth0

interface eth0
inform 192.168.1.99
static routers=192.168.1.1
static domain_name_servers=8.8.8.8
static domain_search=8.8.4.4
interface eth0
    static ip_address=192.168.1.99/24
    static routers=192.168.1.1
    static domain_name_servers=127.0.0.1

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line

```

(Archivo de Configuración del DHCP – Imagen 6)

8. Aquí podemos poner lo que se ve arriba en blanco:

```

interface eth0

inform 192.168.99

static routers=192.168.1.1

static domain_name_servers=8.8.8.8

static domain_search=8.8.4.4

interface eth0

    static ip_address=192.168.1.99/24

    static routers=192.168.1.1

    static domain_name_servers=127.0.0.1

```

9. Después de hacer eso reiniciamos la máquina para asegurar que la configuración se ha guardado correctamente.
10. Para comprobar que lo que hemos hecho ha funcionado, usamos el comando:

```
ifconfig
```

Ahora siempre podemos conectarnos con PuTTY a la misma dirección IP.

10. Ahora podemos cambiar la contraseña por defecto a nuestra propia contraseña. Esto se hace con el comando

```
sudo raspi-config
```

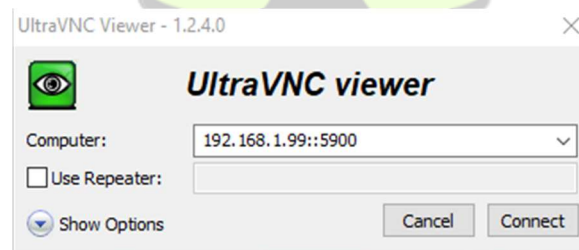
11. Ahora para acceder de forma gráfica tenemos que instalar un servicio VNC. En nuestro caso vamos a usar x11vnc. Usando el comando siguiente empezara la instalación:

```
sudo apt-get install x11vnc
```

12. Después de instalar este servicio dentro de PuTTY ponemos el comando para arrancar el servicio:

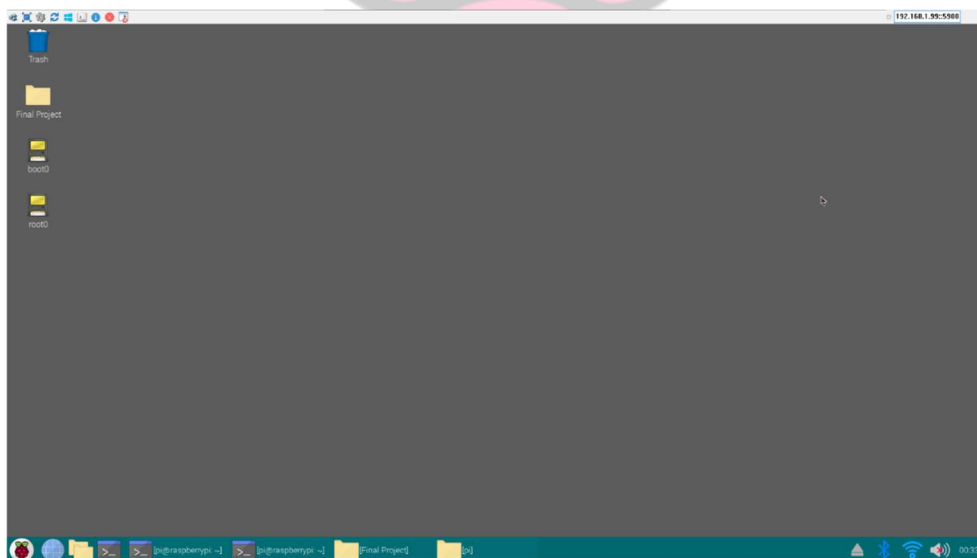
```
x11vnc
```

13. Y con el software UltraVNC que descargamos del internet (<https://www.uvnc.com/>) podemos meter la IP fija y el puerto 5900 así para conectarnos:



(Panel de conexión VNC de UltraVNC Viewer – Imagen 7)

14. Y como vemos en la imagen 8 estamos conectados gráficamente:



(Conexión sobre VNC al servidor Raspberry Pi – Imagen 8)

➤ **Instalación del primer servicio (Pi-Hole):**

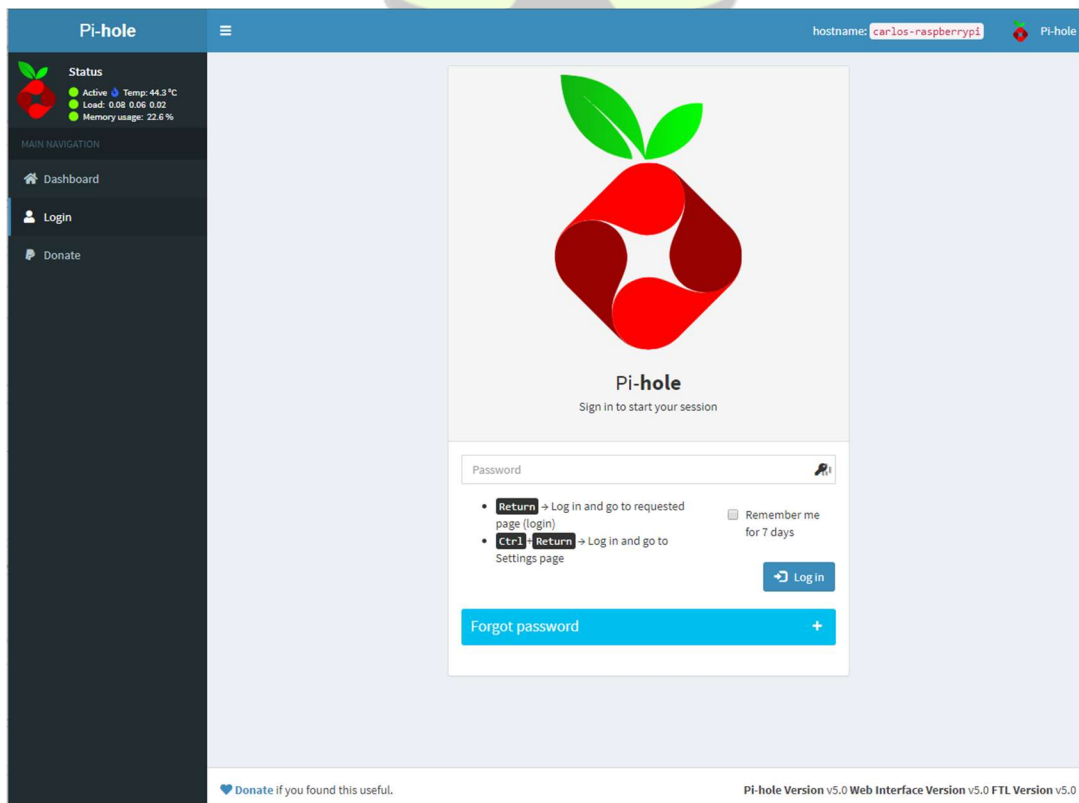
1. En principio Pi-Hole es un proceso muy simple de instalación. El primer paso es usar:

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

2. En cuanto usemos este comando la instalación de PI-Hole empezara.
 - a. Primero elegimos un DNS para que el servidor funcione
 - b. Siguiendo, asignamos la IP y la puerta de enlace.
3. Cuando termine la instalación reiniciamos la máquina.
4. Con el comando cambamos la contraseña para el panel de administrador:

```
pihole -a -p
```

5. Para acceder al panel admin vamos a cualquier buscador y usar la URL `http://192.168.x.y/admin` pero en vez de la x, y ponemos la IP que le asignamos.
6. Y con eso deberíamos estar aquí:

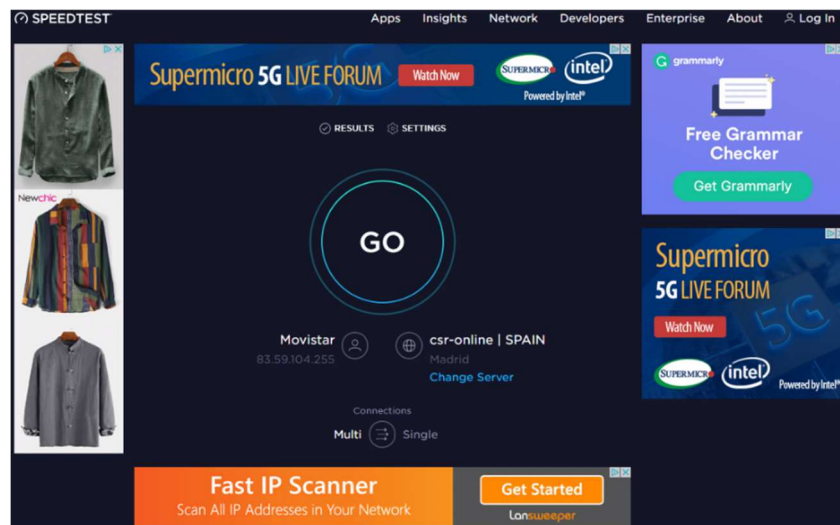


(Panel Administrativo de Pi-Hole – Imagen 9)

7. Cuando entremos podemos entrar en todas las páginas. El historial, las listas de bloqueo y de permiso, las herramientas, y mucho más.

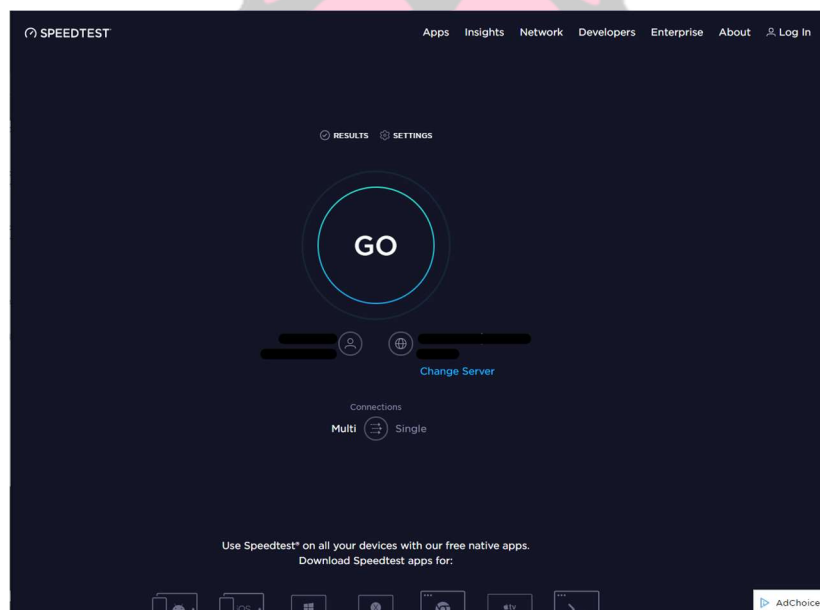
➤ **Comprobación del servicio (Pi-Hole):**

Aquí tenemos un ejemplo de lo que hace la Pi-Hole como vemos, antes de usar el servicio, vemos que hay 5 anuncios diferentes.



(Página de ejemplo con publicidad – Imagen 10)

En cuando cambiamos nuestro servidor a la nuestra podemos ver abajo como ya no hay ningún anuncio:



(Página sin publicidad gracias a Pi-Hole – Imagen 11)

Y para comprobar que de verdad es nuestro servidor el que ha conseguido esto podemos ir a nuestro panel administrativo en 192.168.1.99/admin y ver que se han bloqueado un montón de páginas de anuncios:

Time	Type	Domain	Client	Status	Reply	Action
2020-05-18 17:57:57	A	adserver-us.adtech.advertising.com	192.168.1.50	Blocked (gravity)	-(1.2ms)	✓ Whitelist
2020-05-18 17:57:57	A	ib.adnxs.com	192.168.1.50	Blocked (gravity)	-(0.3ms)	✓ Whitelist
2020-05-18 17:57:57	A	ookla-d.openx.net	192.168.1.50	Blocked (gravity)	-(0.6ms)	✓ Whitelist
2020-05-18 17:57:57	A	hbopenbid.pubmatic.com	192.168.1.50	Blocked (gravity)	-(0.6ms)	✓ Whitelist
2020-05-18 17:57:57	A	as-sec.casalemedia.com	192.168.1.50	Blocked (gravity)	-(3.6ms)	✓ Whitelist
2020-05-18 17:57:56	A	secure-us.imrworldwide.com	192.168.1.50	Blocked (gravity)	-(0.6ms)	✓ Whitelist
2020-05-18 17:57:56	A	gurgle.zdbb.net	192.168.1.50	Blocked (gravity)	-(0.3ms)	✓ Whitelist
2020-05-18 17:57:56	A	www.google-analytics.com	192.168.1.50	Blocked (gravity)	-(0.2ms)	✓ Whitelist
2020-05-18 17:57:56	A	www.google.es	192.168.1.50	OK (forwarded)	IP (31.0ms)	⊗ Blacklist
2020-05-18 17:57:56	A	analytics.google.com	192.168.1.50	Blocked (gravity)	-(0.2ms)	✓ Whitelist
Time	Type	Domain	Client	Status	Reply	Action

(Página de dominios bloqueados– Imagen 12)

Las cosas más importantes que vemos bloqueada aquí es analytics.google.com o google-analytics.com. Esto es un servicio de Google que colecciona información sobre los usuarios a los que se conecta. Colecciona información como edad, genero, intereses, nacionalidad, y muchas cosas más.

También vemos que se ha bloqueado varias páginas de anuncios como hboopenbid.pubmatic.com que puede ser un enlace para anuncios de HBO, y adserver-us.adtech.advertising.com que por lo que dice parece un servidor que gestiona anuncios a varias páginas web.

➤ **Configuración del servicio (Pi-Hole):**

Ahora, si por alguna razón descubrimos que Pi-hole está dejando pasar una página maliciosa, o nos ha salido un anuncio hay dos formas de añadir ese enlace a la lista de dominios bloqueados, o nos metemos en la lista y pulsamos “Blacklist” como vemos al lado de www.google.com:

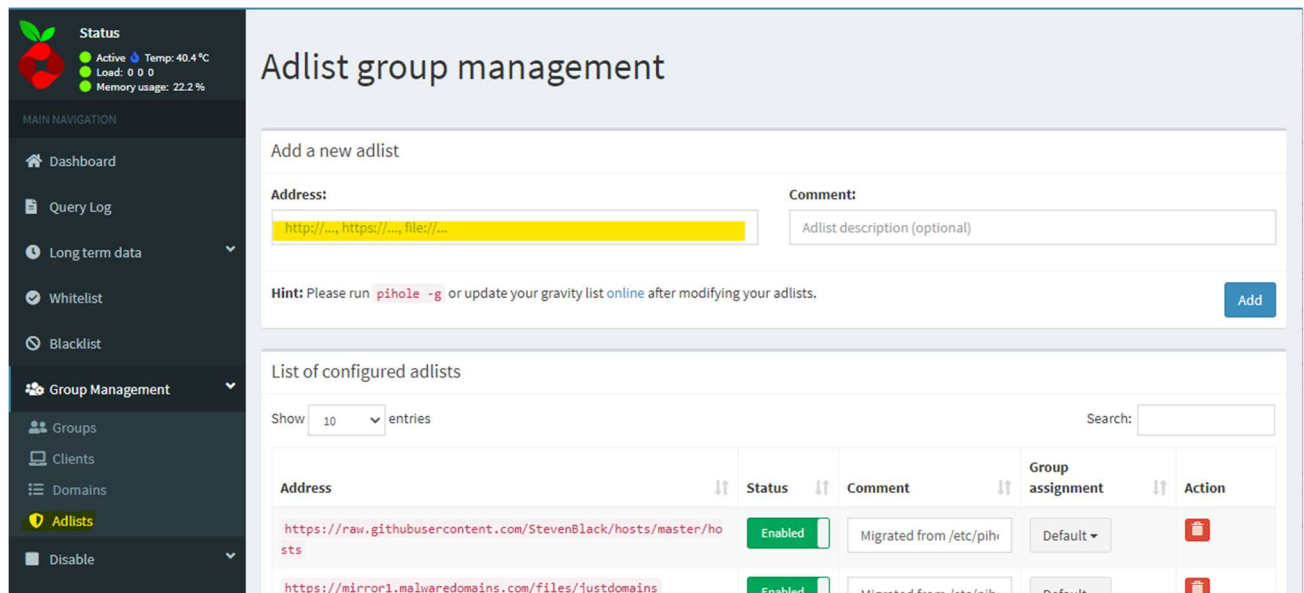
2020-05-18 17:57:56	A	www.google-analytics.com	192.168.1.50	Blocked (gravity)	-(0.2ms)	Whitelist
2020-05-18 17:57:56	A	www.google.es	192.168.1.50	OK (forwarded)	IP (31.0ms)	Blacklist

(Dominios bloqueados comunes de google – Imagen 13)

O, podemos ir a el panel de “Blacklist” y añadirlo manualmente en el campo que dice “domain to be added”:

(Página de Blacklist – Imagen 14)

Otra herramienta útil es la sección de “Adlist” dentro de “Group Management”. Aquí podemos introducir listas generadas por gente online. Por ejemplo, yo puedo entrar en el foro de Pi-Hole y encontrar una lista de todas las paginas relacionadas con el Sevilla FC que una persona ha creado, introducirla en esta página, y así automáticamente todas esas páginas estarán bloqueadas:



Status

- Active
- Temp: 40.4 °C
- Load: 0.0 0.0
- Memory usage: 22.2%

MAIN NAVIGATION

- Dashboard
- Query Log
- Long term data
- Whitelist
- Blacklist
- Group Management**
 - Groups
 - Clients
 - Domains
 - Adlists**
 - Disable

Adlist group management

Add a new adlist

Address:

Comment:

Hint: Please run `pihole -g` or update your gravity list online after modifying your adlists.

List of configured adlists

Show entries

Address	Status	Comment	Group assignment	Action
https://raw.githubusercontent.com/StevenBlack/hosts/master/hosts	<input checked="" type="checkbox"/> Enabled	Migrated from /etc/pihole	Default	<input type="button" value="Delete"/>
https://mirror1.malwaredomains.com/files/justdomains	<input checked="" type="checkbox"/> Enabled	Migrated from /etc/pihole	Default	<input type="button" value="Delete"/>

(Página de añadir listas – Imagen 15)

➤ **Instalación del segundo servicio (OpenVPN):**

El siguiente servicio que vamos a instalar es un servidor VPN para que el cliente pueda conectarse de forma segura de cualquier sitio y tener una conexión segura.

1. Lo primero que tenemos que hacer es montar un servidor DNS. Como la mayoría de las redes privadas en hogares no tienen una IP pública estática por esto tenemos que montar el DNS para que los clientes no tengan problemas conectándose. Para esto vamos a usar un DNS gratis llamado Duck DNS. Si vamos a <https://www.duckdns.org/> y creamos una cuenta, podemos entrar en la sección de dominios y crear uno en este caso he creado uno con mi nombre y apellido:

- a. carlosyaque.duckdns.org

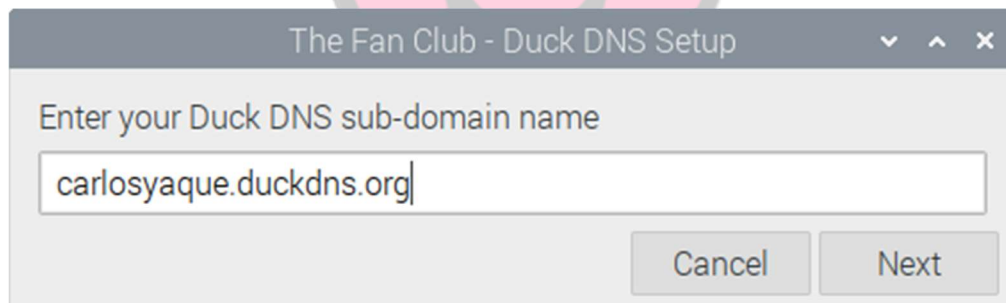
2. Después de crear nuestro dominio, hay que asociar nuestra IP pública a ese dominio y que siempre tenga la IP actualizada. Esto se hace instalando un software o en nuestro router, o en la Pi. Esto lo hacemos con el comando en modo gráfico:

```
sudo apt-get install zenity cron curl
```

```
chmod +x duck-setup-gui.sh
```

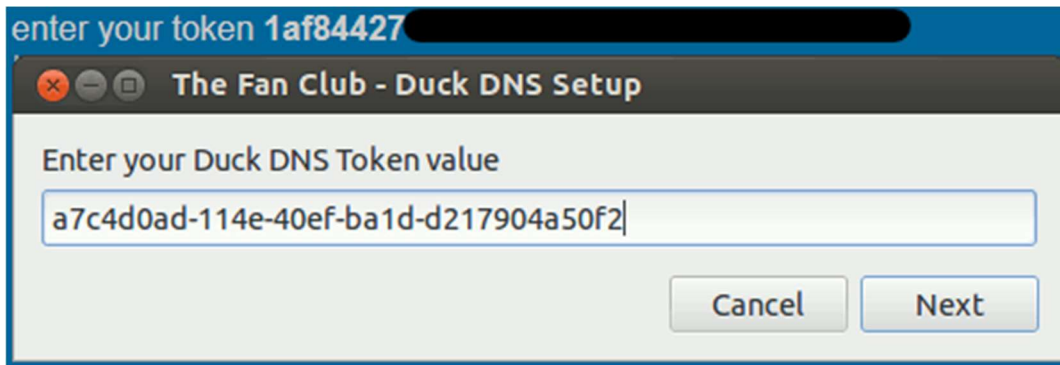
```
./duck-setup-gui.sh
```

3. Cuando terminemos estos comandos debería saltar un panel con un mensaje, aquí tenemos que meter nuestro dominio. El mío es carlosyaque.duckdns.org.



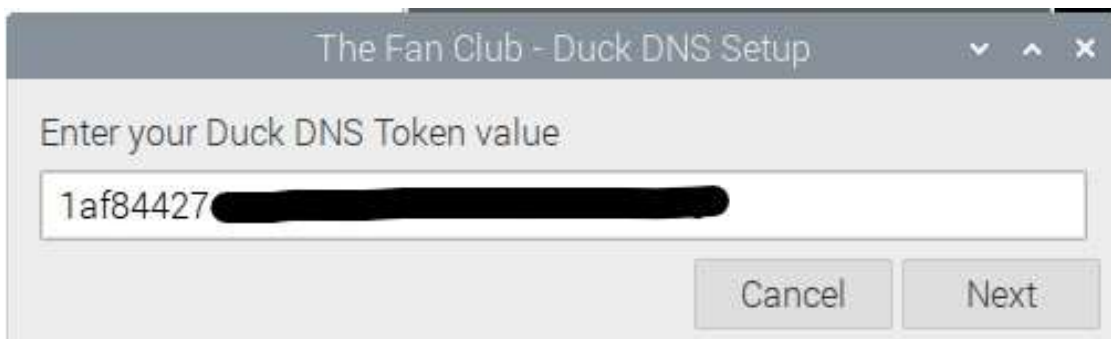
(Panel de configuración de DuckDNS – Imagen 16)

4. Después nos va a pedir el “token” esto se consigue entrando la página web y entrando en la sección “Install” seleccionando tu dominio y entrando en la sección de [linux GUI](#) :



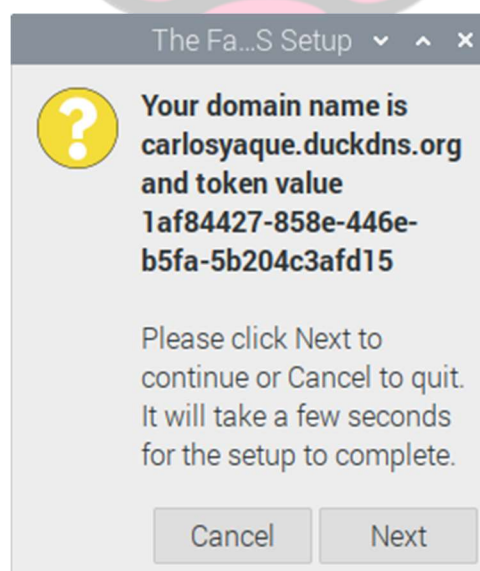
(Panel de configuración de DuckDNS – Imagen 16.2)

5. Aquí como vemos en la imagen 16.3 mi "token" empieza por "1af84427". Copiamos el número entero y lo ponemos en la caja que sale en el instalador:



(Panel de configuración de DuckDNS – Imagen 16.3)

6. Finalmente, para terminar la instalación del DNS solo hay que pulsar "next" y comprobar que la información esta correcta:



(Panel de configuración de DuckDNS – Imagen 16.4)

7. Ahora podemos instalar la VPN. Lo primero que hay que hacer es usar el comando:

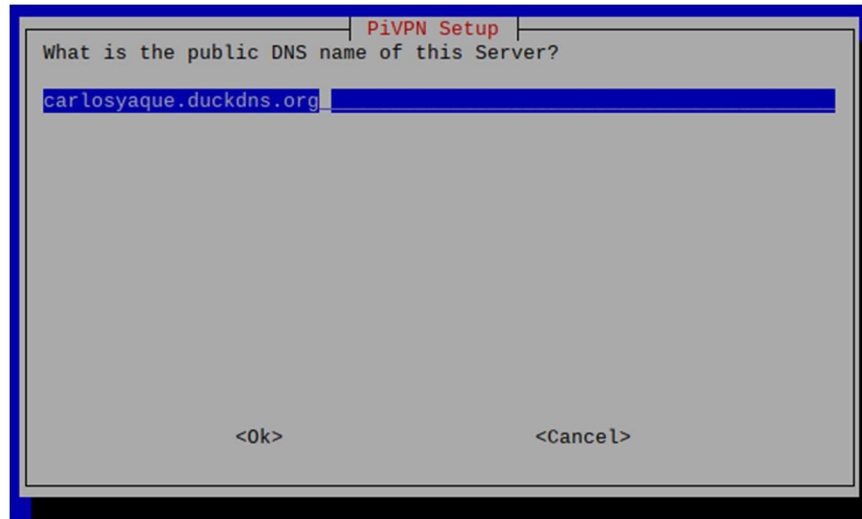
```
curl -L https://install.pivpn.io | bash
```

8. En cuando pongamos este comando, saltara una página que iniciara la instalación del servicio como vemos en la imagen 17.1:



(Instalación de Pi-VPN – Imagen 17.1)

9. En este instalador tendremos que configurar los siguientes parámetros:
- La IP del servidor
 - La distribución de VPN que queremos usar (en este caso vamos a usar OpenVPN)
 - Wireguard
 - OpenVPN
 - El Puerto
 - Que por defecto para OpenVPN es 51820
 - La siguiente página detecta que tenemos pi-hole instalado y nos pregunta si queremos usar el proxy de Pi-hole para la VPN.
 - A continuación, la instalación pide si queremos que los clientes se conecten con nuestra IP publica, o un DNS. Vamos a seleccionar DNS ya que lo hemos creado en el principio.



(Instalación de Pi-VPN – Imagen 17.2)

- f. Después de esto empezar a generar las claves públicas y privadas para la encriptación de los datos. Esto puede tardar 5-10 minutos.
 - g. Finalmente pedirá reiniciar el servidor.
- **Comprobación y Configuración del servicio (OpenVPN):**
1. Para comprobar que el servicio se ha instalado correctamente lo que podemos hacer es escribir el comando `pivpn` y ver si sale una lista de todos los diferentes comandos que podemos usar para este servicio.

```

pi@carlos-raspberrypi:~$ pivpn
::: Control all PiVPN specific functions!
:::
::: Usage: pivpn <command> [option]
:::
::: Commands:
::: -a, add          Create a client conf profile
::: -c, clients     List any connected clients to the server
::: -d, debug       Start a debugging session if having trouble
::: -l, list        List all clients
::: -qr, qr         Show the qrcode of a client for use with the mobile app
::: -r, remove      Remove a client
::: -h, help        Show this help dialog
::: -u, uninstall  Uninstall pivpn from your system!
::: -up, update     Updates PiVPN Scripts
::: -bk, backup    Backup VPN configs and user profiles

```

(Comandos de Pi-VPN – Imagen 18)

2. Después de comprobar que el servicio esta arrancado, podemos empezar a configurarlo. Lo primero que deberíamos hacer es usar el comando `pivpn -a` para añadir un cliente nuevo.

```

pi@carlos-raspberrypi:~$ pivpn -a
Enter a Name for the Client: Carlos
How many days should the certificate last? 3650
Enter the password for the client:
Enter the password again to verify:
spawn ./easysrsa build-client-full Carlos

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1d 10 Sep 2019
Generating a RSA private key
.....+++++
.....+++++
writing new private key to '/etc/openvpn/easy-rsa/pki/private/Carlos.key.QjFWE1K0dS'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
Using configuration from /etc/openvpn/easy-rsa/pki/safessl-easysrsa.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'Carlos'
Certificate is to be certified until May 25 10:06:12 2030 GMT (3650 days)

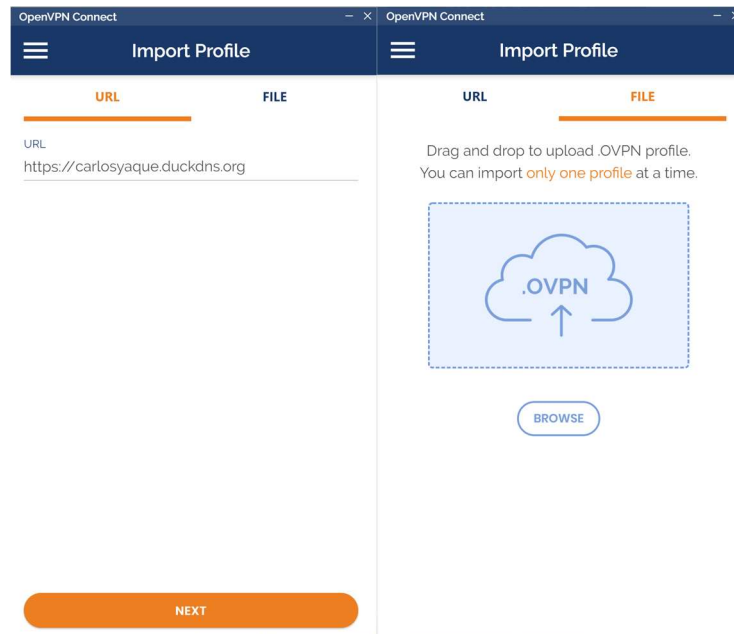
Write out database with 1 new entries
Data Base Updated
Client's cert found: Carlos.crt
Client's Private Key found: Carlos.key
CA public Key found: ca.crt
tls Private Key found: ta.key
::: Updated hosts file for Pi-hole

=====
Done! Carlos.ovpn successfully created!
Carlos.ovpn was copied to:
  /home/pi/ovpns
for easy transfer. Please use this profile only on one
device and create additional profiles for other devices.
=====
pi@carlos-raspberrypi:~$ █

```

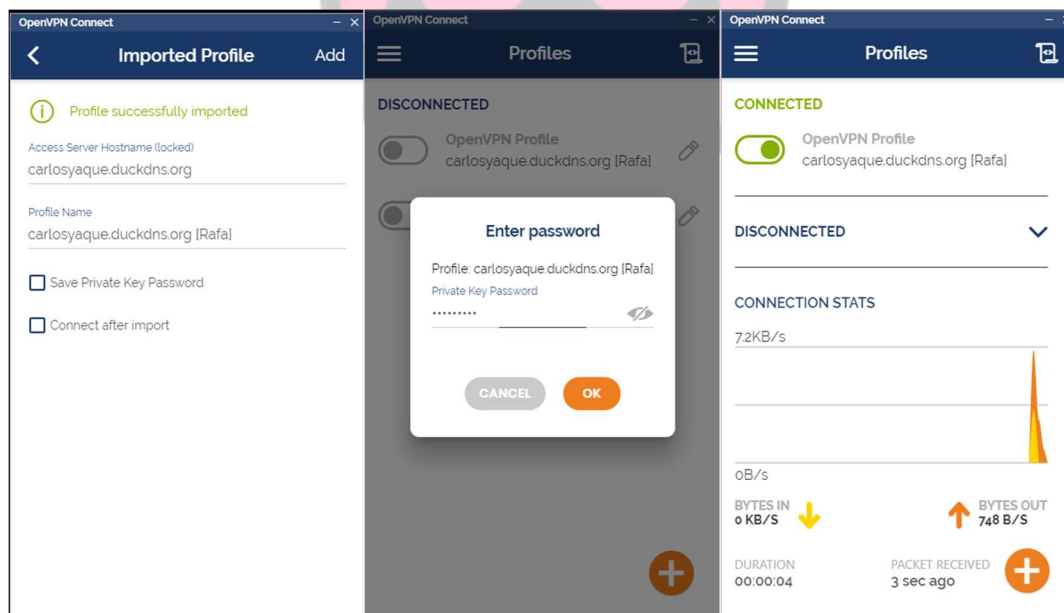
(Creación de Usuario para Pi-VPN – Imagen 19)

3. Cuando creamos un usuario nuevo, nos va a pedir un nombre, y una contraseña para el cliente. En este caso yo he creado uno llamado Carlos con la contraseña Carlos. Cuando termine creando la clave y demás nos creara un archivo .ovpn que es con que nos vamos a conectar a la VPN. (Si hubiéramos elegido instalar WireGuard, la otra VPN, tendríamos también la opción de añadir la VPN por código QR).
4. Ahora con un pendrive o alguna forma de mover el archivo de la PI al ordenador al que queremos conectar, vamos al archivo home/pi/ovpns y copiamos el archivo Carlos.ovpn a nuestro disco/pendrive.
5. Ahora en el otro dispositivo vamos a: <https://openvpn.net/client-connect-vpn-for-windows/> para descargarnos el cliente de Windows. Cuando lo tengamos descargado, podemos conectarnos con en DNS que hemos montado, o con el archivo que se creó con el usuario:



(Aplicación de OpenVPN – Imagen 20)

- Después de usar cualquier de los dos métodos, deberíamos de poder pulsar, importar, añadir, y meter la contraseña para poder conectarse. Como vemos en las imágenes 21 compañero de clase Rafa, pudo conectarse a mi VPN con un usuario creado para él desde su casa:



(Conexión de OpenVPN a nuestro servidor VPN – Imagen 21)

Conclusiones

➤ Pi-Hole:

Inicialmente instalé y puse a prueba el nuevo dispositivo el 17 de marzo del 2020. Y el 13 de mayo 2020 mire las estadísticas para ver como de bien a funcionado:



(Datos de Pi-Hole – Imagen 22.1)

Como se puede ver en la imagen de arriba, en solo 1 mes y 26 días de las 20,184 peticiones 4,496 fueron bloqueadas por nuestro Pi-hole. Y como vemos eso es casi un 25% de todo el tráfico de solo 3 dispositivos de nuestra red. Eso es una gran cantidad de tráfico que en redes normales sin protección está entrando y saliendo constantemente. Cosas como anuncios, malware, spyware, phishing y muchas más cosas. Esto es también gracias a las listas añadidas que puso nuestra "Blocklist" a 135,036 dominios bloqueados.

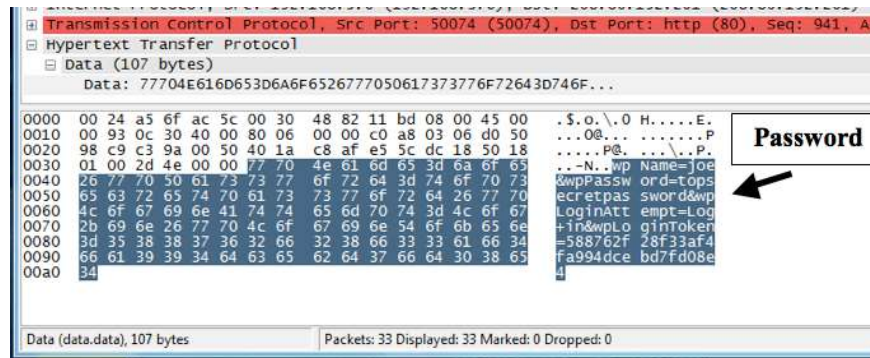
Top Blocked Domains		
Domain	Hits	Frequency
lcprd1.samsungcloudsolution.net	33974	<div style="width: 100%;"></div>
mobile.pipe.aria.microsoft.com	13704	<div style="width: 40%;"></div>
sb.scorecardresearch.com	11500	<div style="width: 35%;"></div>
browser.pipe.aria.microsoft.com	7030	<div style="width: 20%;"></div>
settings-win.data.microsoft.com	6877	<div style="width: 18%;"></div>
api.stathat.com	6740	<div style="width: 18%;"></div>
www.google-analytics.com	5558	<div style="width: 15%;"></div>
cdn.ap.bittorrent.com	4713	<div style="width: 12%;"></div>
vortex.data.microsoft.com	4250	<div style="width: 11%;"></div>
ads.samsungads.com	3614	<div style="width: 9%;"></div>

(Datos de Pi-Hole – Imagen 22.2)

El dominio más bloqueado es el de Samsung por tener la Smart TV de Samsung conectada al Pi-Hole. Y también uno de los más bloqueados son los de Microsoft y Google que ayudan contra la venta de datos personales y anuncios que salen personalizado por las búsquedas.

➤ **PiVPN (OpenVPN):**

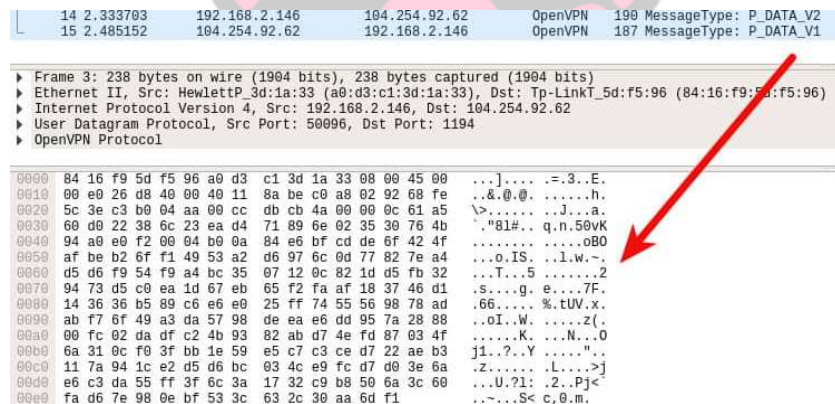
La VPN si montada correctamente puede ayudar a proteger los datos y privacidad de cualquier cliente, como vemos en la imagen 23, usando un software llamando Wireshark, un software gratis y que cualquier persona puede descargar, podemos “sniffear” la red y ver contraseñas, usuarios, claves, datos bancarios, y cualquier otra información sensible que no queremos que cualquier persona vea:



(Paquete sin encriptación – Imagen 23)

(Project 3: Stealing Passwords with a Packet Sniffer, n.d.)

Y como vemos en la imagen 24, después de activar la VPN la información se convierte incomprensible:



(Paquete con encriptación – Imagen 24)

(Watson, 2018)

OpenVPN usa “256-bit OpenSSL encryption” que significa que cada clave generada tendrá 115,792,089,237,316,195,423,570,985,008,687,907,853,269,984,665,640,564,039,457,584,007,913,129,639,936 (78 dígitos) posibles combinaciones. Aun usando el super ordenador más potente del mundo tardaría millones de años en “crackear” esa clave. (Nohe, 2019)

Referencias:

(n.d.). Obtenido de Github: <https://github.com/>

(n.d.). Obtenido de Reddit: [reddit.com](https://www.reddit.com/)

(n.d.). Obtenido de LinusTechTips: [linustechtips.com](https://www.linuxtechtips.com/)

(n.d.). Obtenido de Pi-Hole Forums: <https://discourse.pi-hole.net/>

Nohe, P. (2019, Mayo 2). *How strong is 256-bit Encryption?* Obtenido de Hashedout: <https://www.theslstore.com/blog/what-is-256-bit-encryption/>

Project 3: Stealing Passwords with a Packet Sniffer. (n.d.). Obtenido de samsclass.info: <https://samsclass.info/123/proj10/p3-sniff.htm>

Watson, J. (2018, June 20). *What is packet sniffing and how can you avoid it?* Obtenido de comparitech: <https://www.comparitech.com/blog/information-security/what-is-packet-sniffing/>

