Instalación de un servidor FreeDMR
(versión no Docker)
así como su MONITOR

Documento v1.5.1 Realizado por **Patrick F1FQN** (**Buda feliz**) miembro del Shield Group
al 17 de diciembre de 2021

FreeDMR: http://www.freedmr.uk/

Muchas gracias a todo el equipo de codificadores FreeDMR

Servidor TheShield: http://theshield.site:8080

Servidor TheGate (La French Connection)
http://thegate.hblink.net:8080

Gracias por su colaboración:

Nicolás **F4ISE** (Escudo Grupo)
Youness **CN8VX** (DMR Marruecos)
Daniel **F1RMB** (Desarrollador de la bifurcación Pistar y el equipo de firmware alternativo GD77)

Características de un servidor FREEDMR

- **Dial-a-TG** - Esta función le permite llamar a cualquier número TG mediante una llamada privada haciéndolo enrutar por el TG9 (similar al antiguo sistema reflector: en su transceptor, conexión al número TG9 + TG)
- **ID de voz** : Texto a voz que anuncia en qué servidor está conectado.
- **Puente dinámico** : En el modo OPENBRIDGE, no se debe escribir ninguna regla en el archivo rules.py.
- **Puente estático dinámico** : Transferencia automática de tráfico a todos los demás OPENBRIDGES).
- **Operación de puerto único para conectividad de punto de acceso** : (se requiere un puerto de inicio de sesión único: 62031 para todos los puntos de acceso de invitados
- **Gestión optimizada para OPENBRIDGE** paquetes duplicados, faltantes o fuera de servicio.
- **Modo singular** : Cuando está habilitado, solo un TG puede estar activo en un TS a la vez.
- **Control de temporizador** para grupos de noticias activados por el usuario.
- **Soporte para la configuración del cliente** enviado a través de la línea de Opciones de Protocolo Homebrew La línea de OPCIONES se puede programar en su PISTAR u otro Hotspot o en sus aplicaciones DMR (Droidstar, DudeStar, etc.)
- **Reconocimiento de paquetes HBP DMRA**
- **Soporte para idiomas internacionales** para indicaciones de voz para Dial-a-TG
- **Múltiples puentes abiertos y sin bucles.** - hace posible una red en malla sin perturbaciones.
- **Detección y manejo de paquetes duplicados y fallidos**.
- **Detección y gestión de bucles**.
- **Protocolo de control OpenBridge / Bridge mejorado**
- **OpenBridge funciona detrás del enrutamiento NAT y con IP dinámica**
- **Generador de sistemas modelados** (Le permite generar automáticamente una cierta cantidad de zonas de Hotspots.
- **Imagen de Docker para una instalación preconfigurada** pero no realmente documentado. Para estar reservado a los especialistas de Docker

1- Instalación básica del servidor

Lea y comprenda antes de comenzar a crear un servidor FreeDMR:

Tiene la posibilidad de crear un servidor independiente y privado pero también un servidor que integre una red existente. Este documento te brinda las bases técnicas para lograrlo. Sin embargo, si desea crear un servidor FreeDMR cuyo propósito sea integrar una red perteneciente a una o más organizaciones, necesariamente habrá que observar reglas de nomenclatura y conexión.

Por lo tanto, el propósito de este documento es solo para fines educativos y está orientado únicamente con el objetivo de crear un servidor privado.

En Debian 10 o superior

Después de instalar y asegurar una distribución Debian recién instalada:

actualización de sudo apt

actualización de sudo apt

sudo apt install git sudo apt install python3-pip sudo apt install python-pip

python3 -m pip install pip --upgrade cd / opt **(inútil en Debian 11)**

(inútil en Debian 11)

clon de git https://gitlab.hacknix.net/hacknix/FreeDMR.git

Este comando importa el directorio de instalación localmente en su servidor. Tras el lanzamiento de este, se creará un directorio FreeDMR en **/optar** Mientras permanece en el nivel de **/optar**

cd / opt / FreeDMR

chmod + x install.sh

sudo ./install.sh

cp FreeDMR-SAMPLE.cfg hblink.cfg **(aquí, copiamos el archivo de ejemplo en un archivo de configuración)**

cp rules_SAMPLE.py rules.py (ídem aquí para el archivo de configuración TG)

Atención, aquí la primera trampa establecida por los desarrolladores: FreeDMR-SAMPLE es un "-" y rules_SAMPLE, es un "_"

Hecho esto, nuestros 2 archivos de configuración **hblink.cfg** y **rules.py** se crean pero aún no pueden hacer que su servidor esté operativo.

Volveremos a cómo crear sus propios TG, pero por el momento y para ilustrar la configuración básica, conectaremos su servidor a un servidor asociado para recuperar los TG y hacerlos disponibles en el suyo.

2- Las 3 formas de conectar su servidor a un servidor asociado

- Modo PEER
- Modo XLXPEER
- Modo OPENBRIDGE

- **Modo PEER:** Su servidor se conecta como un Hotspot similar a Pistar y "recupera" uno o más TG de un servidor **socio (HBLink, FreeDMR, IPSC2, TGIF)**

- **Modo XLXPEER:** Su servidor se conecta como un Hotspot similar a Pistar y "recupera" uno o más TG de un servidor de puerta de enlace **Tipo XLX**

- **Modo PUENTE ABIERTO** es un canal de comunicación dedicado entre 2 servidores y no tiene límite en cuanto al número de TG. Entonces es necesario contactar al SYSOP del servidor remoto para acordar e intercambiar las direcciones IP, puertos de conexión y contraseñas.**(HBLink, FreeDMR, IPSC2, Brandmeister)**

Primero, vamos a hacer algunas adaptaciones a nuestros 2 archivos creados previamente (hblink.cfg y rules.py) con el fin de **conéctanos en modo PEER** en nuestro servidor THEGATE y así recuperar el **TG9379**. Vamos !!

Abramos nuestro archivo hblink.cfg (archivo de gestión de conexión) que debería aparecer por defecto así:

nano /opt/FreeDMR/hblink.cfg

[GLOBAL]
SENDERO: ./
PING_TIME: 10
MAX_FALTADOS: 3
USE_ACL: Verdadero
REG_ACL: PERMIT: TODOS
SUB_ACL: DENY: 1
TGID_TS1_ACL: PERMIT: TODOS
TGID_TS2_ACL: PERMIT: TODOS
GEN_STAT_BRIDGES: Verdadero
ALLOW_NULL_PASSPHRASE: Verdadero
ANNOUNCEMENT_LANGUAGES: en_GB, en_US, es_ES, fr_FR, de_DE, dk_DK, it_IT, no_NO, pl_PL, se_SE, pt_PT, cy_GB, el_GR, CW SERVER_ID: 0000

[INFORMES]
INFORME: Verdadero
INFORME_INTERVAL: 60
INFORME_PORTE: 4321
REPORT_CLIENTS: 127.0.0.1

[LOGGER]
LOG_FILE: freedmr.log
LOG_HANDLERS: archivo
cronometrado LOG_LEVEL: INFO
LOG_NAME: FreeDMR

[ALIAS]
TRY_DOWNLOAD: Verdadero
SENDERO: ./
PEER_FILE: peer_ids.json
SUBSCRIBER_FILE: suscriptor_ids.json
TGID_FILE: talkgroup_ids.json
PEER_URL: https://www.radioid.net/static/rptrs.json SUBSCRIBER_URL: https://www.radioid.net/static/users.json TGID_URL: http: // downloads.freedmr.uk/downloads/talkgroup_ids.json STALE_DAYS: 7

[MYSQL]
USE_MYSQL: Falso
USUARIO: hblink
PASS: mypassword
DB: hblink
SERVIDOR: 127.0.0.1
PUERTO: 3306
TABLA: repetidores

[OBP-TEST]
MODO: PUENTE ABIERTO
HABILITADO: Falso
IP:
PUERTO: 62044
NETWORK_ID: 1
CONTRASEÑA: mypass
TARGET_IP:
TARGET_PORT: 62044
USE_ACL: Verdadero
SUB_ACL: DENY: 1
TGID_ACL: PERMISO: TODOS
RELAX_CHECKS: Verdadero
ENHANCED_OBP: verdadero

[SISTEMA]
MODO: MAESTRO
HABILITADO: Verdadero
REPETIR: Verdadero
MAX_PEERS: 1
EXPORT_AMBE: Falso
IP: 127.0.0.1
PUERTO: 54000
CONTRASEÑA:
GROUP_HANGTIME: 5
USE_ACL: Verdadero
REG_ACL: DENY: 1
SUB_ACL: DENY: 1
TGID_TS1_ACL: PERMISO: TODOS
TGID_TS2_ACL: PERMISO: TODOS
DEFAULT_UA_TIMER: 60
SINGLE_MODE: verdadero
VOICE_IDENT: verdadero
TS1_STATIC:
TS2_STATIC:
DEFAULT_REFLECTOR: 0
ANNOUNCEMENT_LANGUAGE: **en_FR**
GENERADOR: 100

3-Creación de una sección PEER a un servidor (Ejemplo, el servidor THEGATE)

[**THEGATE**]
MODO: PAREJA
ACTIVADO: **Cierto**
SUELTO: Falso
EXPORT_AMBE: Falso
IP:
PUERTO: 54301
MASTER_IP: **thegate.hblink.net**
MASTER_PORT**: 62031**
CONTRASEÑA: **passw0rd**
SEÑAL DE LLAMADA: **Tu indicativo**
RADIO_ID: **Su ID de DMR CCS7 o CCS7 + sufijo de 01 a 99** RX_FREQ:
449000000
TX_FREQ: 444000000
TX_POWER: 25
CÓDIGO DE COLOR: 1
RANURAS: 2
LATITUD: 46.764043
LONGITUD: 5.835659
ALTURA: 320
ALQUILER: **Su ciudad, país o localizador de QRA**
DESCRIPCIÓN: URL del punto de acceso virtual:
www.google.fr
SOFTWARE_ID: 20170620
PACKAGE_ID: MMDVM_HBlink
GROUP_HANGTIME: 5
OPCIONES:
USE_ACL: Verdadero
SUB_ACL: DENY: 1
TGID_TS1_ACL: DENY: TODOS
TGID_TS2_ACL: PERMIT: **9379**
ANNOUNCEMENT_LANGUAGE: fr_FR

Ahora abramos nuestro segundo archivo de configuración: **rules.py**
(reglas de TG y Time Slots que desea que estén disponibles en su servidor)

nano /opt/FreeDMR/rules.py

```
PUENTES = {

}
if __name__ == '__main__':
    de pprint importar pprint
    pprint (PUENTES)
```

y agregaremos aquí nuestra regla de enrutamiento (entre 2 {})

```
PUENTES = {
                '9379': [ {'SYSTEM': 'THEGATE', 'TS': 2, 'TGID': 9379, 'ACTIVE': True, 'TIMEOUT': 10, 'TO_TYPE': 'NONE', 'ON': [9379], ' APAGADO ': [],' RESTABLECER ': []},
                    ],
            }
if __name__ == '__main__':
    de pprint importar pprint
    pprint (PUENTES)
```

PRECAUCIÓN: Tan pronto como active una zona PEER en **hblink.cfg**, debe haber configurado al menos una línea de enrutamiento correspondiente a esta conexión PEER en **rules.py**.

[**THEGATE**]
MODO: PAREJA
ACTIVADO: **Cierto**

Debes tener al menos una línea como:
'9379': [{'SYSTEM': 'THEGATE', 'TS': 2, 'TGID': 9379, 'ACTIVE': True, 'TIMEOUT': 10, 'TO_TYPE': 'NONE', 'ON': [9379], ' APAGADO ': [],' RESTABLECER ': []},

¡Una llave, un corchete, una coma faltante y su servidor no se iniciará!

Su servidor ahora está configurado para recopilar 1TG en modo PEER de THEGATE y ponerlo a disposición en su propio servidor.

El siguiente paso es automatizar el inicio del servidor FreeDMR como un "servicio" creando el archivo **freedmr.service** situado en **/lib / systemd / system / freedmr.service**

Para hacer esto, abrimos un archivo vacío con el comando nano.

nano /lib/systemd/system/freedmr.service

Luego copiamos el contenido a continuación dentro

[Unidad]
Descripción = FreeDMR Server After = network-online.target syslog.target Wants = network-online.target

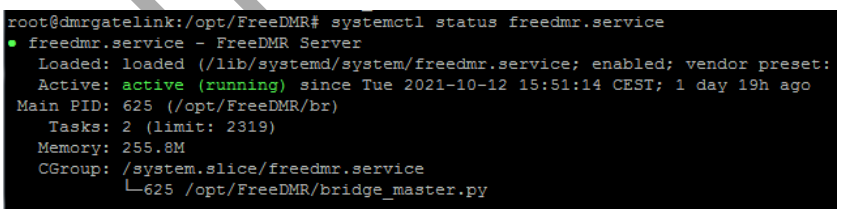
[Servicio]
StandardOutput = nulo
WorkingDirectory = / opt / FreeDMR
RestartSec = 3
ExecStart = / usr / bin / python3 /opt/FreeDMR/bridge_master.py
Reiniciar = al abortar

[Instalar en pc]
WantedBy = multi-user.target

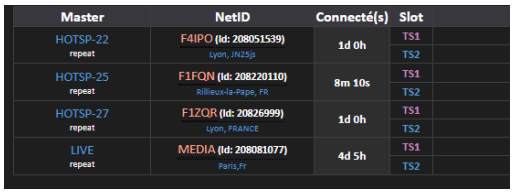
Una vez guardado este archivo, debes: activarlo, iniciarlo y finalmente probar el inicio del servidor con los siguientes comandos:

systemctl enable freedmr
systemctl start freedmr
systemctl status freedmr

Tanto los primeros 2 comandos, cuando los ejecuta, no son detallados como el tercero se encarga de verificar la sintaxis correcta de sus archivos hblink.cfg y rules.py. Un punto verde te dice que todo está bien.
¡Un punto rojo y tendrás que buscar el problema en tus archivos de configuración y esto puede limitarse a olvidar una sola coma o una marca de verificación!



En este punto, debe iniciar sesión en el servidor de TheGate y puede ver que su servidor está conectado como un punto de acceso simple.
Por ejemplo, mi servidor de prueba está aquí representado por la línea F1FQN en Hotspot-27



Master	NetID	Connecté(s)	Slot
HOTSP-22 repeat	F4IPO (Id: 208051539) Lyon, JN25js	1d 0h	TS1
			TS2
HOTSP-25 repeat	F1FQN (Id: 208220110) Rillieux-la-Pape, FR	8m 10s	TS1
			TS2
HOTSP-27 repeat	F1ZQR (Id: 20826999) Lyon, FRANCE	1d 0h	TS1
			TS2
LIVE repeat	MEDIA (Id: 208081077) Paris,Fr	4d 5h	TS1
			TS2

4-Creación del área de recepción para hotspots de invitados

Por cierto, hablando de puntos de acceso, ciertamente desea poder alojar conexiones de invitado en su servidor que usarán su TG local o los TG que ha importado de otros servidores (este también es el propósito de la manipulación)

Aquí es donde hay una modificación importante en comparación con el servidor HBlink de primera generación.
En las versiones anteriores, si deseaba distribuir un conjunto de TG disponibles en su servidor, tenía que crear manualmente tantas zonas de puntos de acceso (cada una con un puerto de comunicación diferente) como "suscriptores"; de lo contrario, una conexión en la zona predeterminada (puerto 62031) solo le dio acceso a uno o 2 TG arreglados de una vez por todas.

Mientras tanto, FreeDMR permite la gestión dinámica de puntos de acceso conectados. En pocas palabras, todos los hotspots se conectan al puerto 62031 y mediante un sistema de “proxy inverso”, se colocan en una zona personalizada en la que puedes elegir libremente usar 2 TGs entre los ofrecidos en tu servidor. Este mecanismo se configura gracias al archivo**/ opt / FreeDMR / hotspot_proxy_v2.py**

Dentro de este archivo encontramos esto:

```
#*** CONFIG HERE ***

    Master = "127.0.0.1"
    ListenPort = 62031
    # '' = all IPv4, '::' = all IPv4 and IPv6 (Dual Stack)
    ListenIP = ''
    DestportStart = 54000
    DestPortEnd = 54100
    Timeout = 30
    Stats = False
    Debug = False
    ClientInfo = False
    BlackList = [1234567]

```

En el archivo **/opt / FreeDMR / hblink.cfg** en la última línea de la sección **[SISTEMA],** puede ajustar la cantidad de puntos de acceso que desea alojar**. Por defecto, este número es 100 (GENERADOR: 100).**
Por supuesto, puede, según su elección, reducir el número de puertos en reserva. Para nuestras 100 conexiones, creará y reservará 100 puertos de conexión desde 54000 a 54100.

(Si desea alojar solo 10 puntos de acceso, solo abra de 54000 a 54010)

1er hotspot que se conectará a su servidor en el puerto predeterminado **62031** será por tanto redirigido de forma dinámica y aleatoria a un puerto libre entre 54000 y 54100, lo mismo para el segundo y los siguientes ...
Tenga en cuenta que estos puertos ahora estarán reservados y nunca deben usarse para ninguna conexión a su servidor.

Ahora vamos a crear un servicio para que el proxy inverso se lance cuando se inicie el servidor.
Para hacer esto, abrimos un archivo vacío con el comando nano:

nano /lib/systemd/system/hotspot_proxy.service

Luego copiamos el contenido a continuación dentro

[Unidad]
Descripción = FreeDMR Hotspot_proxy After = network-online.target syslog.target Wants = network-online.target

[Servicio]
StandardOutput = nulo
WorkingDirectory = / opt / FreeDMR
RestartSec = 3
ExecStart = / usr / bin / python3 /opt/FreeDMR/hotspot_proxy_v2.py
Reiniciar = al abortar

[Instalar en pc]
WantedBy = multi-user.target

Una vez guardado este archivo, debes: activarlo, iniciarlo y finalmente probar el inicio del hotspot_proxy con los siguientes comandos:

systemctl enable hotspot_proxy
systemctl start hotspot_proxy
systemctl status hotspot_proxy

```
root@dmrgatelink:/opt/FreeDMR# systemctl status hotspot_proxy.service
● hotspot_proxy.service - FreeDMR Hotspot_proxy
   Loaded: loaded (/lib/systemd/system/hotspot_proxy.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-10-12 15:51:14 CEST; 1 day 20h ago
 Main PID: 635 (/opt/FreeDMR/ho)
    Tasks: 1 (limit: 2319)
   Memory: 18.4M
   CGroup: /system.slice/hotspot_proxy.service
           └─635 /opt/FreeDMR/hotspot_proxy_v2.py
```

5-Creación de una zona dedicada a un hotspot en un puerto IP fijo

Acabamos de ver que el servidor puede alojar dinámicamente un cierto número de conexiones de invitados (100 por defecto) cuya conexión entrante está invariablemente en el puerto. **62031** y cuyo puerto de retorno es aleatorio entre **54000 y 54100 (para 100 puntos de acceso)**

También podría ser interesante (además de estas áreas dinámicas) poder hacer que una o más áreas fijas estén disponibles para conectar puntos de acceso específicos para trabajar. en un puerto fijo dedicado
.

En este caso, puede crear una nueva zona de SISTEMA estática que cambiaremos de nombre y a la que asignaremos un puerto fijo (aquí ejemplo del puerto **55555**)

[**NOMBRE DE SU ZONA DEDICADA**]
MODO: MAESTRO
HABILITADO: Verdadero
REPETIR: Verdadero
MAX_PEERS: 1
EXPORT_AMBE: Falso
IP:
PUERTO: **55555** **(no debe estar en el rango de 54000 a 54100)**
CONTRASEÑA: **una contraseña privada o la contraseña tradicional**
GROUP_HANGTIME: 5
USE_ACL: Verdadero
REG_ACL: DENY: 1
SUB_ACL: DENY: 1
TGID_TS1_ACL: PERMISO: TODOS
TGID_TS2_ACL: PERMISO: TODOS
DEFAULT_UA_TIMER: 60
SINGLE_MODE: verdadero
VOICE_IDENT: verdadero
TS1_STATIC:
TS2_STATIC:
DEFAULT_REFLECTOR: 0
ANNOUNCEMENT_LANGUAGE: fr_FR
GENERATOR: 0 **El GENERADOR debe permanecer fijo en 0**

Aquí está su servidor que ahora está listo para acomodar un centenar de conexiones y, en este punto, está en pleno funcionamiento.

6-Creando su monitor de servidor - Panel de control

Te falta algo importante para conocer de un vistazo la actividad de tu servidor, es decir, el monitor también llamado DASHBOARD, en definitiva el dashboard en formato Web de tus conexiones y eso es lo que instalaremos ahora.

Es importante saber que el monitor no es desarrollado por el equipo de FreeDMR sino por un OM independiente (SP2ONG), de hecho el servidor y el monitor son dos cosas separadas.

Algunos SYSOP personalizan su Monitor (Dashboard), por ejemplo el del servidor "TheGate" que puedes visitar en esta dirección
https://thegate.hblink.net:8080

De forma predeterminada, después de la instalación y sin la personalización del monitor, su panel se verá así:

Ahora discutiremos la instalación del monitor. Para ello crearemos un directorio de instalación local en nuestro servidor:

cd / opt
clon de git https://github.com/sp2ong/HBMonv2

Este comando importa el directorio de instalación localmente en su servidor. Tras el lanzamiento de este comando, un directorio**HBMonV2** será creado bajo **/optar**

Luego comenzamos la instalación:

cd / opt / HBMonv2
chmod + x install.sh
. /install.sh
cp config_SAMPLE.py config.py

Tendremos que instalar ahora **PHP** y uno **Servidor web**, que puede ser como se desee **lighttpd**, **ngnix** , Donde **apache2** !

Instalación de php:
apto instalar php

Instalación del servidor apache2: **apto instalar apache2**

La instalación del servidor WEB tuvo que crear el directorio **/var / www / html**

Vamos a limpiarlo

cd / var / www / html
rm *. *

El directorio html ahora está vacío

Volvemos a / opt / HBMonv2:

cd / opt / HBMonv2 cp -R
html / var / www /

Con este comando, copiamos de forma recursiva el contenido de **/opt / HBMonv2 / html** en **/ var / www / html**

Ahora puede probar su tablero conectándose con la dirección IP o DNS de su servidor: **http: // Ip_de_your_serveur**

- En el directorio **/var / www / html / include /** el archivo debe estar ahí **config.php**

Al editar este archivo, puede cambiar los colores y darle un nombre a su tablero

- En el directorio **/var / www / html** el archivo debe estar ahí **botones.html** para personalizar sus menús.

- En el directorio **/var / www / html / img /** el archivo debe estar ahí **logo.png** que puede reemplazar.

Finalmente, otro archivo cuyo tamaño debe ser verificado y limpiado es lastheard.log (este es el archivo de los últimos OM's conectados más de 2 segundos en el servidor). Esta tarea se realiza de forma automática y diaria según los parámetros incluidos en el archivo. **último escuchado.**

cp / opt / HBMonv2 / utils / lastheard /etc/cron.daily/
chmod + x /etc/cron.daily/lastheard

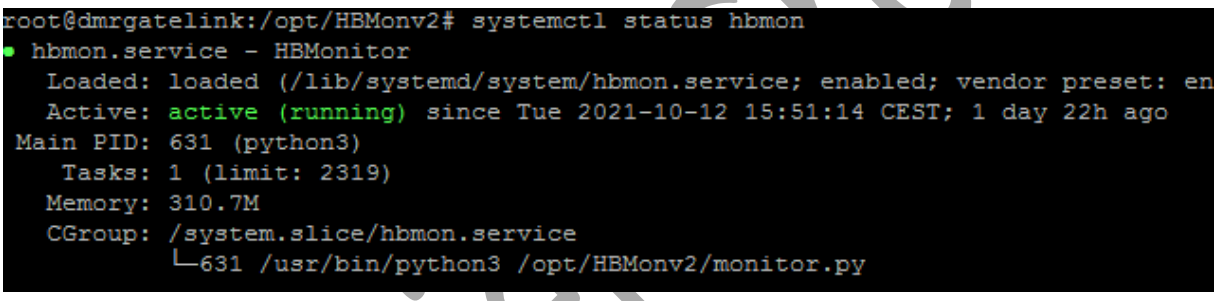
Como para todos los demás servicios, el monitor no es una excepción a la regla y será necesario crear un "servicio" para que se inicie automáticamente.

El archivo **hbmon.service** ya está configurado, solo necesita copiarlo al directorio de servicios.

cp /opt/HBMonv2/utils/hbmon.service / lib / systemd / system /

Ahora tienes que: activarlo, iniciarlo y finalmente probar el inicio del monitor hbmon con los siguientes comandos:

systemctl habilitar hbmon
systemctl start hbmon
systemctl status hbmon



7-Protección con contraseña de su tablero

Si su servidor es privado, o simplemente no quiere que las personas curiosas vean lo que sucede en su servidor, es posible que desee protegerlo con un nombre de usuario y una contraseña. En este caso

Debes crear un archivo vacío **.htpasswd** en / etc / apache2 con el comando **tocar** : **toque / etc/apache2/.htpasswd**

Una vez que se ha creado este archivo vacío, se debe alimentar con una cuenta y una contraseña. Eseno hecho con un editor de texto pero con el comando **htpasswd** (sin el punto esta vez).

htpasswd /etc/apache2/.htpasswd me (yo u otro siendo el nombre de su usuario)

Luego, el comando le pide que ingrese una contraseña y la confirme.. Puede verificar que su archivo **.htpasswd** se ha escrito con el usuario "yo" seguido de su contraseña cifrada.

cat /etc/apache2/.htpasswd

Ahora es necesario modificar el archivo de configuración del servidor Apache que es apache2.conf

nano /etc/apache2/apache2.conf

Para mayor claridad, copiaremos y pegaremos las líneas siguientes al final del archivo apache2.conf y guardaremos

```
<Directorio "/ var / www / html">
        Opciones Índices FollowSymLinks
        AuthType Basic
        AuthName "Acceso restringido"
        AuthUserFile /etc/apache2/.htpasswd
        Requiere usuario válido
</Directorio>
```

Reiniciemos nuestro servidor Apache con el comando habitual:

systemctl reiniciar apache2

Comprobamos que todo está bien con

systemctl estado apache2

La próxima vez que inicie sesión en su servidor, se le pedirá el nombre de usuario y la contraseña que ha elegido.

8-Creación de un OPENBRIDGE con un servidor asociado

Vimos anteriormente cómo crear un enlace. **MIRAR** a un servidor asociado. Como recordatorio, un enlace de este tipo se comporta como la conexión de un hotspot simple y a nivel de servidor,**será necesario escribir una línea en el archivo rules.py.** También necesitará escribir una línea de opciones en su PISTAR (pero volveremos)

OPENBRIDGE (OPB) es la mejor solución para conectar dos servidores asociados. De hecho, hay un canal dedicado abierto entre los 2 servidores y potencialmente puede intercambiar sus TG sin limitación. El protocolo de transmisión OPENBRIDGE está optimizado para este tipo de conexión.

Aparte de los aspectos técnicos, una conexión OPENBRIDGE es socialmente respetuosa porque requiere que te pongas en contacto con el administrador de un servidor asociado para intercambiar tus direcciones IP, tus TG, puertos de envío y recepción, contraseña y especialmente tu **Identificación de red**

Un aspecto importante a recordar es que con FreeDMR, una conexión Openbridge, no es necesario escribir una línea de "reglas" (todo es automático)

Si es un radioaficionado con licencia, tiene 2 posibilidades para configurar un servidor Freedmr:

1- **Para un servidor público**
   Siendo parte integral de esta red y referenciada como tal. Por lo tanto, deberá solicitar un número de red único (ID de red) que lo identificará como servidor oficial de Freedmr.
   Este número será esencial para que funcione la conexión Openbridge.

2- **Para un servidor privado que no sea FreeDMR**
   Deseando beneficiarse de los TG de la red Freedmr pero sin estar afiliado y referenciado oficialmente.

   Para hacer esto, todo lo que necesita hacer es "inventar" una ID de red fuera de las distribuidas oficialmente por FreeDMR. Los "ID de red" oficiales comienzan con 208 (para el país, 208 es Francia) +1 o 2 dígitos.
   Si, por ejemplo, inventa un ID de 10 dígitos como 4568214975, puede estar casi seguro de que nunca entrará en conflicto con otro servidor declarado oficialmente.

Dicho esto, veamos cómo establecer una puerta de enlace OPENBRIDGE (OPB) con un servidor asociado.

- Solicite una identificación de red en Telegram contactando a Simon @hacknix https://t.me/FreeDMR_Building_Server_Support

- Busque un servidor asociado que ofrezca los TG que desea utilizar en su propio servidor. Si desea crear un nuevo TG y ponerlo a disposición de todo el mundo, deberá solicitarlo a Norman @NormanFreeDMR, quien le enviará un formulario para completar.

http://www.freedmr.uk/index.php/world-wide-talk-groups/

Ps: Los servidores FreeDMR pueden potencialmente distribuirle todos los TG globales a menos que el administrador del servidor haya decidido (filtrando) mantener solo aquellos que son útiles. Sin embargo, es recomendable (sin que tu VPS a 5 € esté demasiado ocupado) establecer solo 3 Openbridges como máximo (redundancia o para recuperar en otro servidor, TG que tu primer socio no te ofrece).

Una vez que haya encontrado su servidor asociado, comuníquese con su administrador y solicite un enlace de Openbridge. No conozco un "directorio" de administradores del servidor FreeDMR, pero si miras a tu alrededor, ¡lo encontrarás!
Le enviará una sección de Openbridge para instalar en su archivo hblink.cfg o freedmr.cfg (versión de Docker). Este archivo se verá así:

[OBP-PARTNER]
MODO: PUENTE ABIERTO
HABILITADO: Verdadero
IP: PUERTO: 62100 (la IP: PORT y TARGET_PORT pueden ser idénticos o diferentes) * El ID de
IDENTIFICACIÓN DE RED: red que el administrador de su socio le comunicará
CONTRASEÑA: La contraseña que le dará el administrador de su socio La IP o
TARGET_IP: DNS de su servidor socio
TARGET_PORT: 62100 El puerto IP ofrecido por su servidor asociado USE_ACL:
Verdadero
SUB_ACL: DENY: 1
TGID_ACL: PERMISO: La lista de TG que desea recuperar; los demás serán rechazados **
RELAX_CHECKS: Verdadero
ENHANCED_OBP: verdadero

*Si desea mantener todos los TG del servidor asociado: TGID_ACL: PERMISO: TODOS
* *Cada vez que se conecte un nuevo OPB, será necesario aumentar su **IP: PUERTO**

Debe comunicar su ID de red y la dirección IP de su servidor al administrador del socio.

Importante: No olvide registrar su **ID de red en la sección GLOBAL** desde su archivo hblink.cfg **SERVER_ID: su ID de red**
Sin él, estará bien conectado, pero no se transmitirá ninguna transmisión. Tu archivo**hblink.cfg** por lo tanto, evolucionará así:

[GLOBAL]
SENDERO: ./
PING_TIME: 10
MAX_FALTADOS: 3
USE_ACL: Verdadero
REG_ACL: PERMIT: TODOS
SUB_ACL: DENY: 1
TGID_TS1_ACL: PERMIT: TODOS
TGID_TS2_ACL: PERMIT: TODOS
GEN_STAT_BRIDGES: Verdadero
ALLOW_NULL_PASSPHRASE: Verdadero
ANNOUNCEMENT_LANGUAGES: en_GB, en_US, es_ES, fr_FR, de_DE, dk_DK, it_IT, no_NO, pl_PL, se_SE, pt_PT, cy_GB, el_GR, CW SERVER_ID:
0000 **- reemplácelo aquí con su ID de red**

[INFORMES]
INFORME: Verdadero
INFORME_INTERVAL: 60
INFORME_PORTE: 4321
REPORT_CLIENTS: 127.0.0.1

[LOGGER]
LOG_FILE: freedmr.log
LOG_HANDLERS: archivo
cronometrado LOG_LEVEL: INFO
LOG_NAME: FreeDMR

[ALIAS]
TRY_DOWNLOAD: Verdadero
SENDERO: ./
PEER_FILE: peer_ids.json
SUBSCRIBER_FILE: suscriptor_ids.json
TGID_FILE: talkgroup_ids.json
PEER_URL: https://www.radioid.net/static/rptrs.json SUBSCRIBER_URL:
https://www.radioid.net/static/users.json TGID_URL: http: //
downloads.freedmr.uk/downloads/talkgroup_ids.json STALE_DAYS: 7

########################
# #
Mysql ha sido abandonado
# #
########################

[MYSQL]
USE_MYSQL: Falso
USUARIO: hblink
PASS: mypassword
DB: hblink
SERVIDOR: 127.0.0.1
PUERTO: 3306
TABLA: repetidores

########################
# #
Sección OPENBRIDGE
# #
########################

[OBP-PARTNER]
MODO: PUENTE ABIERTO
HABILITADO: Verdadero **Si es falso, su OPB está deshabilitado**
IP: PUERTO: 62044 **(la IP: PORT y TARGET_PORT pueden ser idénticos o diferentes) El ID de red**
IDENTIFICACIÓN DE RED: **que el administrador de su socio le comunicará**
CONTRASEÑA: **La contraseña que le dará el administrador de su socio La IP o DNS de su**
TARGET_IP: **servidor socio**
TARGET_PORT: 62044 **El puerto IP ofrecido por su servidor asociado**
USE_ACL: Verdadero
SUB_ACL: DENY: 1
TGID_ACL: PERMISO: El **lista de TG que desea recuperar separados por comas**. El otro TG será rechazado o**TODO** para mantener todos los TG
RELAX_CHECKS: Verdadero
ENHANCED_OBP: verdadero

```
##############################################
#                                            #
# Generador de zona de hotspot #
#                                            #
##############################################
```

[SISTEMA]
MODO: MAESTRO
HABILITADO: Verdadero
REPETIR: Verdadero
MAX_PEERS: 1
EXPORT_AMBE: Falso
IP: 127.0.0.1
PUERTO: 54000
CONTRASEÑA:
GROUP_HANGTIME: 5
USE_ACL: Verdadero
REG_ACL: DENY: 1
SUB_ACL: DENY: 1
TGID_TS1_ACL: PERMISO: TODOS
TGID_TS2_ACL: PERMISO: TODOS
DEFAULT_UA_TIMER: 60
SINGLE_MODE: verdadero
VOICE_IDENT: verdadero
TS1_STATIC:
TS2_STATIC:
DEFAULT_REFLECTOR: 0
ANNOUNCEMENT_LANGUAGE: **en_FR**
GENERADOR: 100

```
########################
#                      #
# Sección PEER         #
#                      #
########################
```

[**THEGATE**]
MODO: PAREJA
ACTIVADO: **Cierto**
SUELTO: Falso
EXPORT_AMBE: Falso
IP:
PUERTO: 54301
MASTER_IP: **thegate.hblink.net**
MASTER_PORT**: 62031**
CONTRASEÑA: **passw0rd**
SEÑAL DE LLAMADA: **Tu indicativo**
RADIO_ID: **Su ID de DMR CCS7 o CCS7 + sufijo de 01 a 99** RX_FREQ:
449000000
TX_FREQ: 444000000
TX_POWER: 25
CÓDIGO DE COLOR: 1
RANURAS: 2
LATITUD: 46.000000 (para cambiar)
LONGITUD: 5.000000 (para cambiar)
ALTURA: 320
ALQUILER: **Su ciudad, país o localizador de QRA**
DESCRIPCIÓN: URL del punto de acceso virtual:
www.google.fr
SOFTWARE_ID: 20170620
PACKAGE_ID: MMDVM_HBlink
GROUP_HANGTIME: 5
OPCIONES:
USE_ACL: Verdadero
SUB_ACL: DENY: 1
TGID_TS1_ACL: DENY: TODOS
TGID_TS2_ACL: PERMIT: **9379**
ANNOUNCEMENT_LANGUAGE: **fr_FR**

9-Creación de una conexión XLXPEER en una puerta de enlace XLX

Hay servidores que actúan como pasarelas entre los diferentes modos digitales. Estos son servidores XLX.
Puerta de enlace C4FM <-> DMR, DSTAR <-> DMR

Ejemplo: la puerta de enlace XLX933C http:// xlx933.dstarfrance.fr/index.php?show=modules

Su IP es: 152.228.170.6 o su DNS: xlx933.dstar-france.fr
y en la tabla de la derecha encontrará una columna llamada DMR en la que hay nombres de módulos. Estos nombres de módulo se pueden mostrar como un número de 4 dígitos o el nombre del servidor seguido de la letra del alfabeto correspondiente a los 2 últimos dígitos del módulo.

Ejemplo: xlx933-Y será equivalente al módulo xlx933 4025 25 porque Y es la 25ª letra del alfabeto.

Una conexión en el servidor xlx933 módulo 4025 pondrá su servidor DMR en contacto con la red YSF-France que opera en C4FM.

Asimismo, una conexión al servidor xlx933 módulo 4003 conectará su servidor DMR y la red DSTAR-France.

Una conexión XLXPEER es una variante de la conexión PEER pero tiene 5 particularidades:

Moda : **XLXPEER** en lugar de PEER
SUELTO: **Cierto** mientras que es Falso para una conexión PEER
MASTER_PORT: **62030**
RANURAS: **1**
Deberá crear un número TG para esta conexión en la sección de reglas

Recuerdo :

RANURAS: **0** **TS1 solamente**
RANURAS: **1** **TS2 solamente**
RANURAS: **2** **TS1 o TS2**

Utilice TG9 / TS2 para escribir la línea de reglas

[**XLX933-DSTAR**]
MODA: **XLXPEER**
HABILITADO: Verdadero
SUELTO: **Cierto**
EXPORT_AMBE: Falso
IP:
PUERTO: 54607 **Un puerto no utilizado en su servidor** MASTER_IP:
152.228.170.6 MASTER_PORT**: 62030**

CONTRASEÑA: **passw0rd**
SEÑAL DE LLAMADA: **Tu indicativo** RADIO_ID:
208xxxx **Su ID de DMR CCS7** RX_FREQ:
44900000
TX_FREQ: 444000000
TX_POWER: 25
CÓDIGO DE COLOR: 1
RANURAS: **1**
LATITUD: **46.000000 (para cambiar)** LONGITUD: **5.000000 (se modificará)** ALTURA: 320
ALQUILER: **Tu ciudad**
DESCRIPCIÓN:
URL: http: //www.google; fr
SOFTWARE_ID: 20170620
PACKAGE_ID: MMDVM_FreeDMR
GROUP_HANGTIME: 5
4000 + la posición numérica del módulo en el alfabeto, por ejemplo, A = 4001
XLXMODULE: **4003**
USE_ACL: Verdadero
SUB_ACL: DENY: 1
TGID_TS1_ACL: PERMISO: TODOS
TGID_TS2_ACL: PERMISO: TODOS
ANNOUNCEMENT_LANGUAGE: fr_FR


```
PUENTES = {
                '9379': [{'SYSTEM': 'THEGATE', 'TS': 2, 'TGID': 9379, 'ACTIVE': True, 'TIMEOUT': 10, 'TO_TYPE': 'NONE', 'ON': [9379], 'APAGADO': [], 'RESTABLECER': []},],
               'TG No.': [{' SISTEMA ': 'XLX933-DSTAR ', 'TS': 2, 'TGID': 9, 'ACTIVE': Verdadero, 'TIMEOUT': 2, 'TO_TYPE': 'NONE', 'ON': [TG No.], 'APAGADO': [], 'RESTABLECER': []},],
                 }
if __name__ == '__main__':
    de pprint importar pprint
    pprint (PUENTES)
```

10-Creación de uno o más TG en su servidor

Seguramente también le gustaría poder crear su propio TG.

Nada podría ser más simple Como se indicó anteriormente, todo lo que tiene que hacer es completar los siguientes campos en el área [SISTEMA]:

```
############################################
#                                          #
# Generador de zona de hotspot #
#                                          #
############################################
```

[SISTEMA]
MODO: MAESTRO
HABILITADO: Verdadero
REPETIR: Verdadero
MAX_PEERS: 1
EXPORT_AMBE: Falso
IP: 127.0.0.1
PUERTO: 54000
CONTRASEÑA:
GROUP_HANGTIME: 5
USE_ACL: Verdadero
REG_ACL: DENY: 1
SUB_ACL: DENY: 1
TGID_TS1_ACL: PERMISO: TODOS
TGID_TS2_ACL: PERMISO: TODOS
DEFAULT_UA_TIMER: 60
SINGLE_MODE: verdadero
VOICE_IDENT: verdadero
TS1_STATIC: **Aquí los TG que desea crear y poner a disposición para TS1** TS2_STATIC: **Aquí los TG que desea crear y poner a disposición para TS2** DEFAULT_REFLECTOR: 0

ANNOUNCEMENT_LANGUAGE: **en_FR**
GENERADOR: 100

Los TG que se crearán deben estar separados por comas

Tenga cuidado a diferencia de HBLINK en el archivo rules.py:

- El **Campo TG** (ejemplo de TG 9379) solo debe **digital** ! (sin etiqueta TG9379)

- Cada PEER o XLXPEER debe corresponder a una nueva línea de reglas
- No se debe escribir ninguna línea de "reglas" para las conexiones OPB

11- Instalación de un loro

¡No, ese no!

El loro (Parrot) es un servicio que te permite enviarte el eco de tu modulación en un TG en particular (TG9990).

Para construir esto necesitamos:

- Instalar una zona **MIRAR** que nombraremos [**ECO**] en nuestro archivo **hblink.cfg**
- Crea un archivo **reproducción.cfg** al que se conectará nuestro PEER [ECHO]
- Para crear un **servicio de loros** para que el loro se inicie automáticamente con el servidor.

Normalmente, durante la instalación de FreeDMR, el script **install.sh** ya ha instalado todas las dependencias de software necesarias para el correcto funcionamiento de tu loro.

Sin embargo, y si es necesario, estas dependencias están en el archivo requirements.txt **/ opt / FreeDMR / requirements.txt** y, si es necesario, se puede instalar.

Ahora creemos la zona **reproducción.cfg** :

nano /opt/FreeDMR/playback.cfg con el siguiente contenido:

[GLOBAL]
SENDERO: ./
PING_TIME: 10
MÁXIMOS PERDIDOS: 5
USE_ACL: Verdadero
REG_ACL: PERMIT: TODOS
SUB_ACL: DENY: 1
TGID_TS1_ACL: PERMIT: TODOS TGID_TS2_ACL: PERMIT: TODOS GEN_STAT_BRIDGES: Falso
ALLOW_NULL_PASSPHRASE: Falso ANNOUNCEMENT_LANGUAGES: en_GB, en_US, es_ES, fr_FR, de_DE, dk_DK, it_GB, plDSE, it, plIT_GB, plDSE_GB dk_DK, it_0000, 0000, pl_PT_GB, p_t_, 000

[INFORMES]
INFORME: Falso
INFORME_INTERVAL: 60
INFORME_PORTE: 4322
REPORT_CLIENTS: 127.0.0.1

[LOGGER]
LOG_FILE: /var/log/parrot.log
LOG_HANDLERS: archivo
cronometrado LOG_LEVEL: INFO
LOG_NAME: loro

[ALIAS]
TRY_DOWNLOAD: Falso
SENDERO: ./
PEER_FILE: peer_ids.json
SUBSCRIBER_FILE: suscriptor_ids.json
TGID_FILE: talkgroup_ids.json
TGID_URL: http://downloads.freedmr.uk/downloads/talkgroup_ids.json
PEER_URL: https://database.radioid.net/static/rptrs.json SUBSCRIBER_URL: https://database.radioid.net/static/users.json STALE_DAYS: 1

[LORO]
MODO: MAESTRO
HABILITADO: Verdadero
REPETIR: Verdadero
MAX_PEERS: 1
EXPORT_AMBE: Falso
IP: **127.0.0.1**
PUERTO: **54915**
CONTRASEÑA: passw0rd
GROUP_HANGTIME: 5
USE_ACL: Verdadero
REG_ACL: DENY: 1
SUB_ACL: DENY: 1
TGID_TS1_ACL: DENY: TODOS
TGID_TS2_ACL: PERMIT:**9990**
DEFAULT_UA_TIMER: 10
SINGLE_MODE: verdadero
VOICE_IDENT: Falso

Luego edita el área **ECO** de nuestro archivo **hblink.cfg**

[**ECO**]
MODO: PAREJA
HABILITADO: Verdadero
SUELTO: Falso
EXPORT_AMBE: Falso
IP:
PUERTO: **54916**
MASTER_IP: **127.0.0.1**
MASTER_PORT: **54915** **(aquí el puerto debe ser idéntico al de la zona PARROT del archivo playback.cfg)**
CONTRASEÑA: passw0rd
SEÑAL DE LLAMADA: **ECO**
RADIO_ID: **9990** **(Imperativo)**
RX_FREQ: 430500000
TX_FREQ: 439900000
TX_POWER: 25
CÓDIGO DE COLOR: 1
RANURAS: **1 (Recordatorio 0 =TS1 solamente - 1 = TS2 solamente - 2 = TS1 o TS2)** LATITUD: 45.000000

LONGITUD: 004.00000
ALTURA: 320
UBICACIÓN: LYON (69)
DESCRIPCIÓN: ECHO
URL:
SOFTWARE_ID: 20170620
PACKAGE_ID: MMDVM_FreeDMR
GROUP_HANGTIME: 5
OPCIONES:
USE_ACL: Verdadero
SUB_ACL: DENY: 1
TGID_TS1_ACL: PERMISO: TODOS
TGID_TS2_ACL: PERMISO: TODOS
ANNOUNCEMENT_LANGUAGE: fr_FR

Agreguemos la siguiente línea a nuestro archivo **rules.py**

```
'9990': [
        {'SYSTEM': 'ECHO', 'TS': 2, 'TGID': 9990, 'ACTIVE': True, 'TIMEOUT': 2, 'TO_TYPE': 'NONE', 'ON': [9990], ' APAGADO ': [],' RESTABLECER ': []},],
```

Ahora creemos el servicio para lanzar el "Parrot" automáticamente:

Editemos el archivo **loro.servicio:**

nano /lib/systemd/system/parrot.service e inserte las siguientes líneas:

[Unidad]
Descripción = Puente de HB todo Servicio después de
= network-online.target syslog.target Wants =
network-online.target

[Servicio]
StandardOutput = nulo
WorkingDirectory = / opt / FreeDMR
RestartSec = 3
ExecStart = / usr / bin / python3 /opt/FreeDMR/playback.py -c /opt/FreeDMR/playback.cfg
Reiniciar = al abortar
[Instalar en pc]

WantedBy = multi-user.target

Iniciar el servicio PARROT:

Activar el servicio PARROT:
systemctl habilitar loro

Inicie el servicio PARROT:
systemctl start parrot

Probar el funcionamiento del servicio PARROT (que no debería devolver errores)
loro de estado systemctl

No olvide reiniciar su servidor FreeDMR para tener en cuenta la modificación de sus archivos **hblink.cfr y rules.py**

systemctl reiniciar freedmr
systemctl status freedmr

Después de reiniciar, si su zona ECHO está verde en su Tablero, coloque su TX en el **TG9900 / TS2**, hable y escuche sus comentarios.