

Bitcoin.conf

Bitcoin.conf es un archivo de texto plano que contiene la configuración para **bitcoin core**. Cuando este se instala viene por defecto tres cosas distintas:

1. **bitcoin-qt**

Crea un cliente wallet con un frontend para interactuar con el core de Bitcoin.

2. **bitcoind**

Provee un punto completo que se recibe solicitudes que usan comandos RPC Remote Procedure Call (por el puerto 8332 mainnet y 18332 testnet por defecto).

3. **bitcoin-cli**

Sirve para mandar comandos RPC al bitcoind desde un command line o terminal.

Todas las opciones que se dan en comando de línea (excepto `-?`, `-help`, `-version` y `-conf`) se pueden especificar en este archivo.

La prioridad para que tome una configuración es:
directamente en el Command Line > **bitcoin.conf** > Versión GUI

Los usuarios nunca deben realizar cambios de configuración que no entiendan. Además, los usuarios siempre deben tener cuidado de aceptar cualquier cambio de configuración que les proporcione otra fuente (incluso si creen que los entienden).

Se aconseja que el archivo **bitcoin.conf** tenga los siguientes permisos:

owner: Escribir y leer el archivo. 👁 Ojo! no es un archivo ejecutable. Por lo cual no tiene permisos de ejecución.

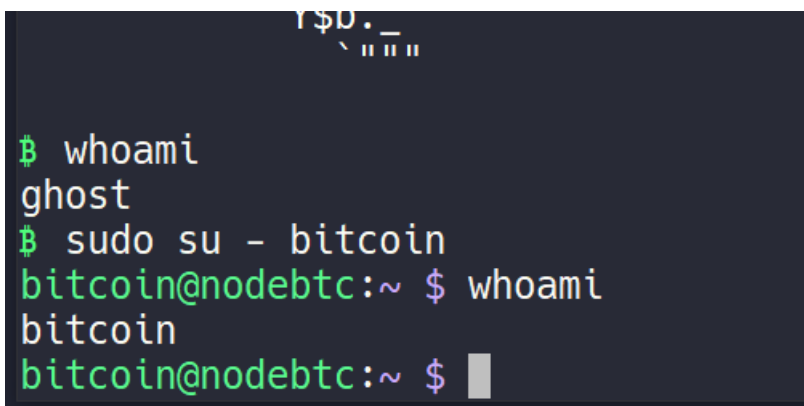
```
group: ninguno
othres: ninguno
```

```
chmod 0600 bitcoin.conf
```

Esta preferencia tiene una razón: El archivo `.conf` puede almacenar variables que son sensibles como:

- Las palabras clave de acceso user y password para acceder al RPC de Bitcoin Core aunque existen formas de ofuscar esta información como `rpcauth.py` que genera un hash de ambas palabras evitando dejar explícitamente el user y password en el archivo de configuración (De esta manera valida solo el hash de user, password proporcionado por el cliente con el hash del archivo de configuración).
- La dirección de una `wallet.dat` u otra info para alguna configuración muy específica (no se recomienda pero es una opción).

En el caso de [Bitcoin/Nodo Bitcoin](#) y [Nodo Testnet](#), por ejemplo, el archivo de configuración es editado desde un user específicamente creado para correr el Bitcoin Core.

A terminal window with a dark background. The text shows a user named 'ghost' running 'whoami' and then 'sudo su - bitcoin'. After the switch, the prompt changes to 'bitcoin@nodebtc:~' and 'whoami' is run again, returning 'bitcoin'.

```
$ whoami
ghost
$ sudo su - bitcoin
bitcoin@nodebtc:~ $ whoami
bitcoin
bitcoin@nodebtc:~ $
```

Formato del Archivo

El config file tiene las siguientes características:

- El archivo es de texto sin formato y consta de líneas **opcion=valor** (una por línea sin espacios)
- Los comentarios son líneas que NO se toman en cuenta al ejecutarse la configuración. Se reconocen al inicio de una línea con el símbolo #
- Se puede ordenar por "Redes específicas" (mainnet, testnet o regnet) donde al ser ejecutada cierta red solo toma las características de configuración que se detallan en su sección.

TXT

```
# Options only for mainnet
[main]

# Options only for testnet
[test]

# Options only for regtest
[regtest]

# tambien toma en cuenta el comando
# que antecede con el nombre de la cadena
regtest.maxmempool=100
```

- La red toma los valores específicos para cada red sobre los valores sin sección. Si múltiples valores se expresan para una misma opción, se toma generalmente la primera.

```
regtest=1
rpcport=2000
regtest.rpcport=3000

[regtest]
rpcport=4000
```

En el ejemplo el valor que toma `rpcport` es 3000.

Configuración del 'path file'

El archivo debe empezar con una ruta, dependiendo del sistema operativo se expresa distinto.

Os	Directorio	Ejemplo
Linux	<code>\$HOME/.bitcoin/</code>	<code>/home/username/.bitcoin/bitcoin.conf</code>
Windows	<code>%APPDATA%\Bitcoin\</code>	<code>C:\Users\username\AppData\Roaming\Bit</code>

Ejemplo:

```
# This config should be placed in following path:  
# ~/.bitcoin/bitcoin.conf
```

En el archivo de configuración específica del nodo se tenía esto:

```
# RaspiBolt: bitcoind configuration  
# /home/bitcoin/.bitcoin/bitcoin.conf  
...
```

Template basic

Las configuraciones más básicas que se pueden encontrar en foros es algo similar a este ejemplo.

```
rpcuser=someusername  
rpcpassword=somepassword  
rpccallowip=localhost  
daemon=1  
prune=600  
minrelaytxfee=2500  
maxconnections=20  
maxuploadtarget=250
```

A continuación hacemos una breve explicación de configuraciones básicas

RPC server

```
-rpcuser=<user>    Username for JSON-RPC connections  
-rpcpassword=<pw>   Password for JSON-RPC connections  
-rpccallowip=<ip>   Allow JSON-RPC connections from  
specified source. Valid for <ip> are a single IP (e.g.  
1.2.3.4), a network/netmask (e.g. 1.2.3.4/255.255.255.0)  
or a network/CIDR (e.g. 1.2.3.4/24). This option can be  
specified multiple times
```

Otros

```
-prune Reduce storage requirements by pruning  
(deleting) old blocks. This mode is incompatible with -  
txindex and -rescan. Warning: Reverting this setting  
requires re-downloading the entire blockchain.  
(default: 0 = disable pruning blocks, >550 = target  
size in MiB to use for block files)
```

`-daemon` Ejecutar en segundo plano como `_daemon_` y aceptar comandos

`minrelaytxfee=2500`

`-minrelaytxfee=<amt>` Fees (in BTC/kB) smaller than this are considered zero fee for relaying, mining and transaction creation (default: 0.00001)

`maxconnections=20`

`-maxconnections=<n>` Maintain at most `<n>` connections to peers (default: 125)

`maxuploadtarget=250`

`-maxuploadtarget=<n>` Tries to keep outbound traffic under the given target (in MiB per 24h), 0 = no limit (default: 0)

Template del Proyecto

Para llevar la gestión del nodo mainnet y testnet que trabajan en distintos ordenadores proponemos un archivo base.

```
#####
# Este archivo de configuración para Bitcoin Core
# contiene los apartados básicos para correr un testnet
#####
# dirección .conf
# /home/bitcoin/.bitcoin/bitcoin.conf

server=1
txindex=1
listen=1
bind=127.0.0.1
rpcauth=ghost:6737b963a783290cc97485931e01b09e$a6d3fc7
ca711426e314a1abdc164f7b55fbc06f1eab314b3322c7884b5f4c
697
zmqpubrawblock=tcp://127.0.0.1:18332
zmqpubrawtx=tcp://127.0.0.1:18333
whitelist=download@127.0.0.1
testnet=1

[test]
rpcbind=0.0.0.0
rpallowip=192.168.1.0/24

[core]
rpcbind=0.0.0.0
rpallowip=192.168.1.0/24
```

Errores

Se listan algunos errores que me saltaron al probar configuraciones distintas en el `bitcoin.conf` y su estado.

-  se solucionó y se explica el porque.

- **✗** no se solucionó y no se desarrolla una explicación completa aún.

Autenticación

! Error: No se puede acceder al comando `bitcoin-cli getblockchaininfo` desde mi usuario por un problema de autenticación con cookies.

Existen dos formas de autenticarse para interactuar con el bitcoin core: declarando user/pass o usando rpcauth.py para no dejar explícitamente el password.

Cuando se usa esta última forma, es decir, sin declarar en el archivo de bitcoin.conf ni `rpcuser rpcpassword` para la autenticación el sistema genera un archivo `.cookie` en la misma dirección que esta el archivo `.conf` e.g.

`/home/bitcoin/.bitcoin/`.

Este archivo que se genera automáticamente tiene permisos de lectura/escritura solo para el user bitcoin. Esto significa que para usar un comando en terminal `bitcoin-cli` debe ejecutarse desde el usuario 'bitcoin' `sudo su - bitcoin` para que funcione correctamente.

```
$ pwd
/home/bitcoin/.bitcoin
$ ls -la .cookie
.rw----- bitcoin bitcoin 75 B Wed Jan 18 10:14:53 2023 .cookie
```

Pero al ejecutarse desde otro usuario 'admin', nos lanzará un error:

```
$ whoami
ghost
$ bitcoin-cli getblockchaininfo
error: Could not locate RPC credentials. No authentication cookie could be f
, and RPC password is not set. See -rpcpassword and -stdinrpcpass. Configu
on file: (/home/ghost/.bitcoin/bitcoin.conf)
```

Para solucionarlo basta con otorgar un permiso de lectura para otros usuarios. Para pulir un poco la seguridad se sugiere

añadir al usuario admin en el grupo 'bitcoin' y darle permisos al grupo.

```
$ sudo chmod g+r /home/bitcoin/.bitcoin/.cookie  
$ ls -la /home/bitcoin/.bitcoin/.cookie  
.rw-r----- bitcoin bitcoin 75 B Wed Jan 18 10:14:53 2023 /home/bitcoin/.bitco  
n/.cookie  
$ bitcoin-cli getblockchaininfo  
{  
  "chain": "main",  
  "blocks": 772539,  
  "headers": 772539,  
  "bestblockhash": "0000000000000000000000004c40764c770c88fea6663759c8c53e71d31e47959  
0be7",  
  "difficulty": 37590453655497.09,  
  "time": 1674052319,  
  "mediantime": 1674049732,  
  "verificationprogress": 0.9999994715873346,  
  "initialblockdownload": false,  
  "chainwork": "00000000000000000000000000000000000000000000000000000000000000003e05d9395b15f61c056c12b8  
",  
  "size_on_disk": 510658986441,  
  "pruned": false,  
  "warnings": ""  
}
```

De esta manera se puede acceder desde otros usuarios.

Cuando se configura una automatización con `systemctl` para que la ejecución de bitcoind tenga lugar luego de un error o reinicio inesperado (como un corte de luz) se toma en cuenta el cambio en los permisos de `.cookie`.

En el archivo `/etc/systemd/system/bitcoind.service` se añade.

```
[Service]
ExecStart=/usr/local/bin/bitcoind -daemon \
        -pid=/run/bitcoind/bitcoind.pid \
        -
conf=/home/bitcoin/.bitcoin/bitcoin.conf \
        -datadir=/home/bitcoin/.bitcoin \
        -startupnotify="chmod g+r
/home/bitcoin/.bitcoin/.cookie"
```

Acceso externo

⚠ Se busca acceder a los comandos `bitcoin-cli` desde otro ordenador en la misma red.

El Bitcoin Core por defecto hace disponible su acceso solo al mismo ordenador o localhost. Pero si se puede configurar para que ciertas ip's puedan acceder.

En el `bitcoin.conf` se agregan:

```
rpccallowip=
# 0.0.0.0/24 means **Class C** network with IP address
range 0.0.0.1 - 0.0.0.254

rpcbind=
# This one tells bitcoind to bind RCP server to
_127.0.0.1_(localhost). That means it will be available
from the same host only.
```

Siendo una solución general para `bitcoin.conf`:

```
rpcallowip=192.168.1.0/24  
rpcbind=0.0.0.0
```

o una configuración más segura si se puede gestionar ips:

```
rpcallowip=192.168.1.155 rpcbind=192.168.1.132
```
