# Administración y Despliegue de UnrealIRCd.


### *Nota sobre la obra:*



Respecto al autor

La obra está realizada por Kamal Majaiti y queda atribuida a kamal majaiti con el cual se puede contactar mediante twitter mediante el siguiente Nick @kamalsenior o a través de la url https://twitter.com/kamalsenior. Ya sea para resolver una duda o cualquier otro interés.

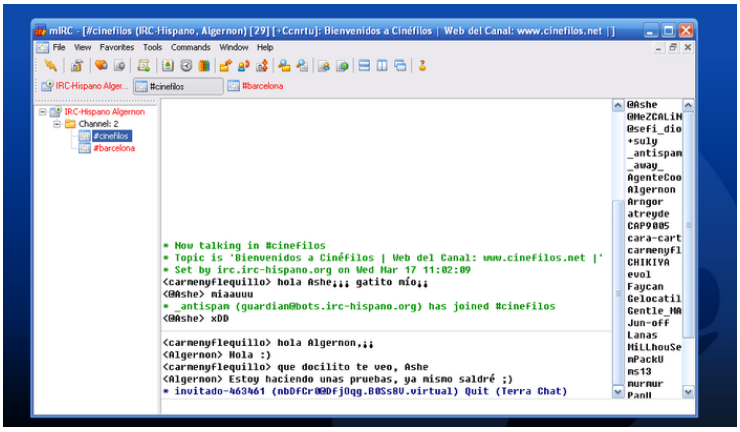
# 1.0 El protocolo IRC.

## *1.1 Introducción*

Es el protocolo de comunicación en tiempo real basado en texto, es el más antiguo de la red. Nació haya por el año 1988 y aun sigue en uso. En la actualidad es muy usado para controlar remotamente troyanos de forma anónima, pero eso no tiene nada que ver con este documento.

## *1.2 Funcionamiento*

Para comenzar una charla, se establece una conexión a un servidor IRC y se Junta a un debate (Canal/Sala) que trata sobre un tema determinado. Una vez dentro se dispone de una lista de usuarios conectados a esa sala. Con los cuales se puede comunicar mediante mensajes. Estos pueden ser de tipo privado que solo son leídos por un único usuario. O de tipo público que son leídos por todos los usuarios a través de la zona publica del canal.



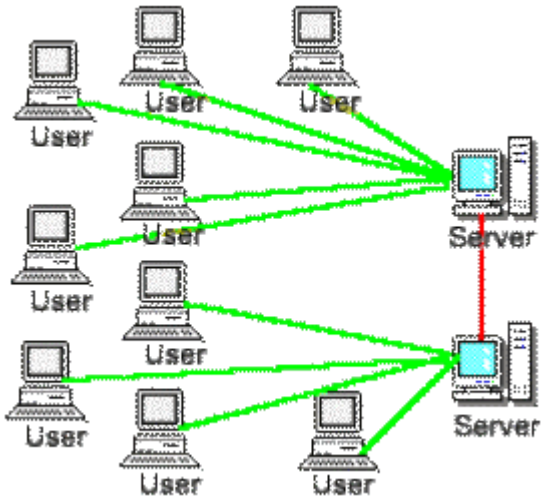

## *1.3 Características de IRC*

- Soporta SSL.
- Usa texto plano.
- Posibilita la compartición de archivos a través de DCC.
- Soporta juegos en línea (Basados en texto).
- Permite la interacción entre varias personas.
- Las comunicaciones quedan registradas a los usuarios en ficheros log.
- Es ligero y muy simple.
- Las operaciones se realizan con comandos en texto plano.
- Se puede modificar el comportamiento de un canal o un usuario mediante modos.
- Los modos se modifican mediante comandos.

## *1.4 Estructura una red IRC.*

- Todo servidor pertenece a una red IRC y una red IRC está formada por servidores.
- Los servidores de la misma red IRC se comunican entre ellos.

- Los canales y los usuarios están distribuidos entre los servidores IRC.



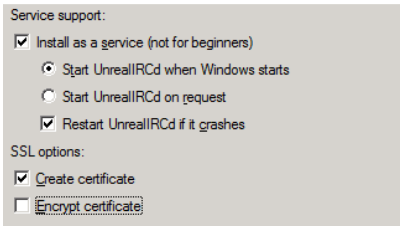

### *1.4 Tipos de usuarios más usados en servidores IRC.*

**Owner:** Es el usuario creador del canal.
**IRCop:** Es el usuario encargado de gestionar y mantener el servidor IRC.
**Clon:** Es aquel usuario que posee dos conexiones al servidor IRC desde una misma ip.
**Half-Op:** Es un usuario avanzado con más privilegios en el canal.
**Operador:** Suele ser el encargado de administrar el canal.
**Bot:** Es un robot normalmente realiza funciones específicas por ejemplo: Saludar a las personas que entran.
**Registrados:** Son usuarios que tienen un Nick reservado en el servidor.
**Anónimos:** Son usuarios que no tienen Nick registrado en el servidor.

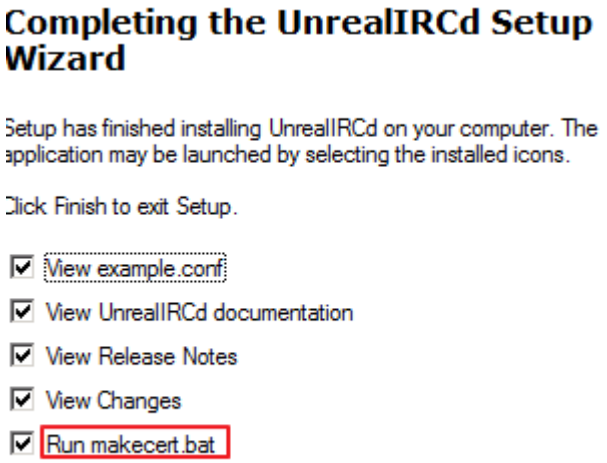
## 2.0 Despliegue del servidor IRC bajo Windows.

A continuación se muestran las tareas necesarias para desplegar un servidor IRC bajo un sistema Windows/Linux con soporte SSL. En este mi caso he escogido este servidor IRC llamado UnrealIRCd disponible en la web http://www.unrealircd.com el cual es Open Source y dispone versiones tanto para Windows como para Linux y la configuración es idéntica.

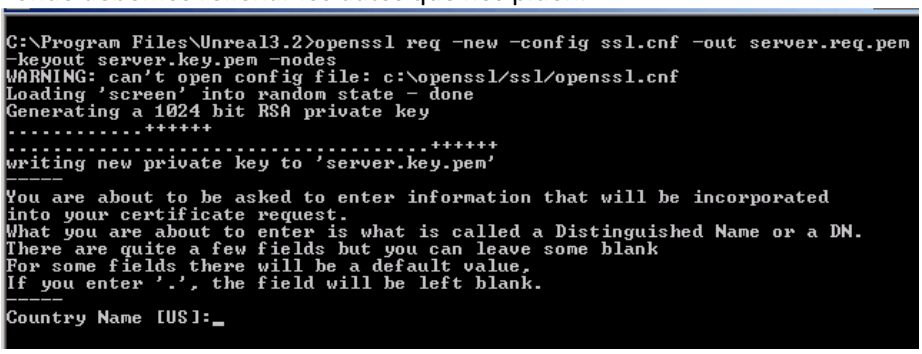
Una vez bajado comenzamos la instalación en la cual debemos marcar estas opciones.



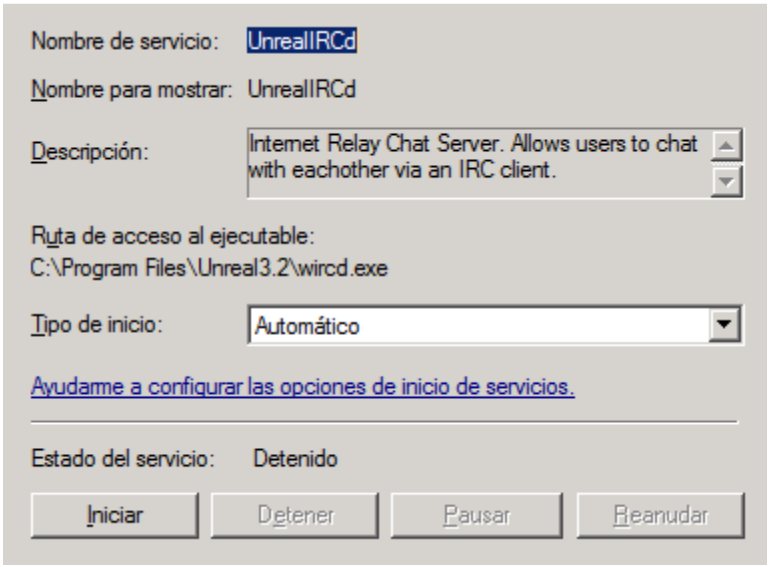

Una vez acabada la instalación, antes de finalizar marcamos la opción de generar certificado.



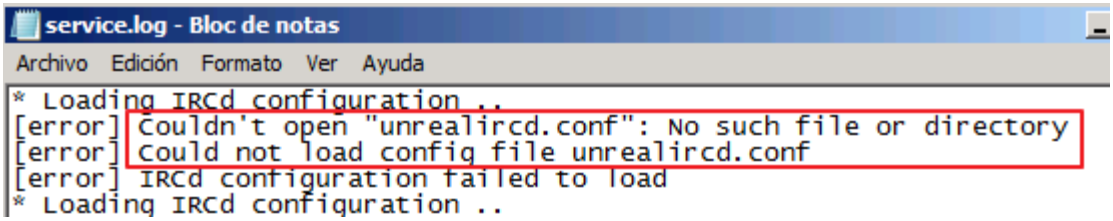

Donde debemos rellenar los datos que nos piden.




Una vez instalado se aprecia que se nos ha instalado un servicio para realizar la función de servidor IRC.




Le damos a iniciar y vemos que no se inicia así que buscamos el fichero log en la carpeta donde se ha instalado normalmente C:\Program Files\Unreal3.2 y una vez dentro miramos el fichero service.log donde apreciamos el primer error.




Ahora disponemos de dos opciones la primera consiste en copiarse el fichero de ejemplo situado en la carpeta doc llamado example.conf y lo renombramos a unrealircd.conf una vez hecho procedemos a editar el fichero.

A continuación se mostrara el contenido de un fichero creado por el autor de este documento y comentado indicando todas las opciones.

En el caso de Linux usted puede realizar la descarga del servidor desde el repositorio de su distribución.

En los sistemas basados en debían puede usted descargarlo con el siguiente comando.

## 3.0 Configuración del servidor IRC.

La primera parte más importante de la configuración es la definición de los módulos del servidor IRC. (Obligatoria)

```
#Carga de modulos Linux
#loadmodule "src/modules/commands.so";
#loadmodule "src/modules/cloak.so";

#Carga de modulos windows:
loadmodule "modules/commands.dll";
loadmodule "modules/cloak.dll";
```

Este servidor IRC dispone de extensiones que se añaden al programa mediante la directiva loadmodule. Estos dos módulos indicados en la imagen vienen por defecto y son obligatorios para el funcionamiento de dicho servidor.

El segundo bloque de código más importante es la sección me. **(Obligatorio)**

```
#Informacion del servidor IRC
me {
    name irc.localhost; #FQDN del servidor.
    info Chat de la Empresa; #Mensaje Informativo.
    numeric 0;
};
```

En esta sección se define una red formada por un servidor (name) el cual contiene la FQDN, después una descripción del servidor (info) y el número de orden del servidor (numeric).

Numeric indica un número para identificar el servidor en nuestra red IRC puesto que cada red puede tener múltiples servidores IRC.

La FQDN es mostrada al conectarse

```
Connected. Now logging in...
Welcome to the Red empresa IRC Network jetcode!jetcode@172.16.0.1
Your host is irc.localhost, running version Unreal3.2.9
This server was created Sat Nov  5 10:17:10 2011
irc.localhost Unreal3.2.9 iowghraAsORTVSxNCWqBzvdHtGp lvhopsmntikrRcaqOAL
UHNAMES NAMESX SAFELIST HCN MAXCHANNELS=5 CHANLIMIT=#:5 MAXLIST=b:60,e:60
```

El tercer bloque consisten en el bloque admin (Obligatorio).

```
admin {
        "Kamal Majaiti";
        "Contacto:";
        "kamal@localhost";
};
```

El bloque admin contiene cierto texto que tiene la función de facilitar el contacto de los usuarios con el administrador IRC.

Los usuarios pueden obtener esta información ejecutando el comando /admin y obtendrán este resultado.

```
Administrative info about irc.localhost
Kamal Majaiti
Contacto:
kamal@localhost
```

El cuarto lugar está el bloque set. (Obligatorio)

```
set {
    network-name "Red empresa"; #Define el nombre de la red
    default-server      "localhost";  #Servidor por defecto de la red.
    services-server     "localhost";  #Servidores de servicios de terceros.
    stats-server        "localhost"; #Servidor de estadisticas IRC.
    kline-address kamal@localhost; #Informacion de contacto del operador de red.
    auto-join "#Sop_Informatica"; #Indica que cuando se conecte un usuario sea autoconectado al canal indicado.
    level-on-join "none"; #Privilegios que tienen los usuarios cuando se unen a un canal por defecto son operadores.
    oper-only-stats "*";  #Evitamos que los usuarios puedan consultar el estado del servidor.
    hiddenhost-prefix "empresa"; #Ocultamos la ip.
    help-channel    "#Sop_Informatica"; #Definimos el canal de ayuda.
    modes-on-oper +gpWi; #Colocamos modos especiales a los Operadores IRC.
    maxchannelsperuser 5; #Definimos el numero maximo por usuario.
    restrict-usermodes AagNoprSWi; #Evitamos escaladas de privilegios a los usuarios.
        hosts {
        local       "locop.172.16.0.34"; #Host con privilegios locales.
        global      "ircop.172.16.0.34"; #Host con privilegios Globales.
        coadmin     "coadmin.172.16.0.34"; #Host del coadministrador.
        admin       "admin.172.16.0.34"; #Host del administrador.
        servicesadmin   "csops.172.16.0.34"; #Host del admin de servicios.
        netadmin    "netadmin.172.16.0.34"; #Host del admin de red.
        host-on-oper-up "no"; #Indica si recibe automaticamen los privilegios cuando se conecte.
        };
        cloak-keys { #Claves usadas para cifrar las ips de los usuarios.
        "a1s2d3f4gDVBNM5h6";
        "a2JO6fh3Q6w4oN3s7";
        "a2J16fh3Q6w2oB3z7";
        };

};
```

El bloque set define un grupo de directivas muy importantes. Todas están indicadas mediante comentarios. Por último los host que coincidan con los indicados en local, global, admin etc.… se les darán esos privilegios. Pero no de forma automática.

En quinto bloque. (Obligatorio) Contiene una clase para definir las características que comparten un grupo de objetos (en este caso los usuarios) del servidor.

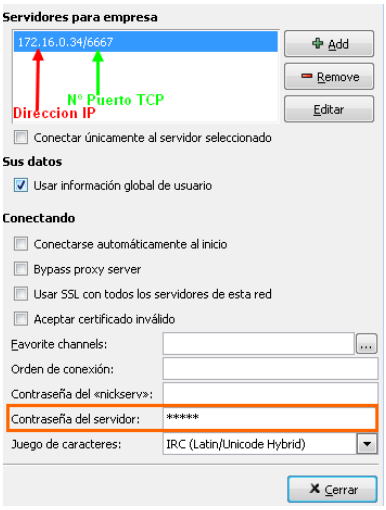
```
53  class usuarios { #Definimos una clase para los usuarios
54      pingfreq 90; #Frecuencia de ping (para comprobar la conexion)
55      maxclients 49; #máximo de clientes unidos al mismo tiempo.
56      sendq 100000;  #Tamaño del buffer de salida.
57      recvq 8000; #Tamaño del buffer de entrada.
58  };
```

Las directivas están comentadas indicando su función así que nada más que decir. Solo que la frecuencia esta en segundos, maxcliente es un numero decimal y en los dos últimos el tamaño es orientativo.

El sexto bloque de código (Obligatorio) es una ACL que controla el acceso al servidor y establece una contraseña a dicho servicio.

```
allow {
    ip *@172.16.0.*; #Restriccion por ip.
    hostname *; #Restriccion por dominion o FQDN.
    password 12345; #Contraseña de acceso al servidor irc.
    maxperip 5; #Numero de conexiones maximas por ip.
    class usuarios; #Indica que afectara a los usuarios.
    options {
        useip; #siempre usa la ip en vez de la FQDN.
        };
};
```

En este caso los usuarios deben conocer la clave de acceso así como la conexión debe provenir de la red definida en la ACL 172.16.0.0/24.
Para conectarse desde el cliente debemos especificar los siguientes datos.




- Contraseña de Acceso.
- Puerto TCP del servidor.
- Dirección IP del servidor.
- Ip de origen este incluida en la ACL.

La imagen pertenece al programa Open Source Xchat. Es un cliente IRC que se puede obtener de forma gratuita tanto los binarios como el código fuente de la página web.
http://xchat.org/

El séptimo bloque (obligatorio) trata sobre el puerto de escucha y sus diversas opciones las más básicas son 2. A continuación en la imagen se aprecia la configuración y la descripción de las directivas.

```
listen *:6667 { #Abrimos un puerto en todas las interfaces.
    options { #Indicamos las Opciones de escucha
        clientsonly; #Solo se pueden conectar clientes.
        #ssl; #Activamos el soporte SSL
        };
};
```

El octavo bloque (obligatorio) es usado para definir a un IRCOP con contraseña. Todo está indicado mediante comentarios.

```
oper kamal { #Definimos un operador llamado kamal.
    class usuarios; #Usamos la configuracion de buffer tipo usuarios
    from {
        userhost 172.16.0.1; #Solo podra entrer de esta ip.
    };
    password "kamal";  #Contraseña de acceso.
    flags { #Indicamos los privilegios del operador.
        netadmin; #Privilegios de red.
        can_gkline; #Puede usar el comando gkline
        can_gzline; #Puede usar el comando gZline
        can_zline; #Puede usar el comando Zline
        can_restart; #Puede reiniciar el servidor.
        can_die; #Puede apagar el servidor.
        global; #Privilegios globales
        admin; #Privilegios de administrador.
        services-admin; #Privilegios de servicios.
        can_dccdeny; #Privilegios sobre transferencias de archivos.
        can_override; #Posibilita saltarse baneos.
    };
    swhois "Example of a whois mask"; #Falsificamos la informacion sobre el usuario.
    snomask frebWqFv; #Mascara cifrar la IP.
};
```

## *Otras directivas para mejorar la seguridad.*

**Establecer contraseñas de reinicio/apagado para el servidor IRC.** (Opcional)

```
drpass {
    restart "12345";
    die "12345";
};
```

No mucho que decir 12345 es la contraseña. Y afecta al comando de la derecha. Donde **die** es apagar y **restart** reiniciar.

**Crear un log Personalizado.** (Opcional)

A veces es necesaria la creación de un registro personalizado con ciertas cosas interesantes como por ejemplo el siguiente.

```
log irc.log { #Registra en el fichero indicado.
    maxsize 15MB ; #Indica el tamaño del fichero.
    flags { #Guarda cierta informacion.
        errors; #Errores.
        server-connects; #Conexiones de servidor.
        oper; #Intentos de Logins de los IRCOP.
    };
};
```

Otras opciones de registro son:

**Flags disponibles :**

| | |
|---|---|
| errors | loguea errores |
| kills | loguea /kill |
| tkl | loguea información de *lines, shuns y spamfilters |
| connects | loguea conexiones/desconexiones de usuarios |
| server-connects | loguea conexiones/desconexiones de servidores |
| kline | loguea el uso /kline |
| oper | intento de oper |
| sadmin-commands | loguea el uso de los comandos /sa* (samode, sajoin, sapart, etc.) |
| chg-commands | loguea el uso de los comandos /chg* (chghost, chgname, chgident, etc.) |
| oper-override | loguea el uso de operoverride |
| spamfilter | loguea las coincidencias de spam |

**Ban permanentes.** (Opcional)

En diversas circunstancias tenemos la necesidad de realizar un ban que sean persistente. Generalmente se usan 2 tipos de ban permanentes que son:

- Por dirección ip.
- Por nombre de usuario.

La siguiente imagen muestra los dos tipos de ban.

```
ban nick { #Banear un nick determinado.
    mask "sex*"; #Nick a buscar para bannear.
    reason "Nick no permitido"; #Mensaje que se le indicara
};

ban ip { #Banear una direccion ip
    mask 192.168.*; #Direccion ip a bannear.
    reason "Este rango de ips no pertenece a la empresa"; #Mensaje
};
```

**Control de transferencias de ficheros.** (Opcional)

El protocolo IRC usa una extensión llamada DCC usada para transmitir ficheros punto a punto. O para comunicarse punto a punto Sin un servidor central.

```
deny dcc { #Denegamos el envio de un tipo archivos.
    filename "*.exe"; #Filtramos por nombre de archivo *exe
    reason "Archivo ejecutable"; #Indicamos porque este denegado.
    soft yes; #Tipo de moderacion.
};

allow dcc {
    filename "*.jpg"; #Permitimos envio de fotos.
    soft no; #Tipo de moderacion.
};
```

Si soft está fijado a yes permitirá al usuario aceptar manualmente el archivo.

Ejemplo de soft=yes.

```
End of /motd command.
* Herimax (Joni12Pack@172.16.0.1) tried to DCC SEND you a file named 'mirc.exe', the request has been blocked.
* Files like these might contain malicious content (viruses, trojans). Therefore, you must explicitly allow anyone that tries to send you such files.
* If you trust Herimax, and want him/her to send you this file, you may obtain more information on using the dccallow system by typing '/DCCALLOW HELP'
* Herimax (Joni12Pack@172.16.0.1) tried to DCC SEND you a file named 'mirc.exe', the request has been blocked.
```

Y si ponemos soft=no.

```
End of /motd command.
* *** Cannot DCC SEND file "uninstall.exe to Herimax (Archivo ejecutable)
* *** You have been blocked from sending files, reconnect to regain permission to send files
```

**Control de acceso a canales.** (Opcional)

Un ejemplo de cómo controlar el acceso a los canales es con la opción.

```
deny channel {
  channel "#help"; #Nombre del canal
  reason "El canal de ayuda es #Sop_Informatica"; #Mensaje
  redirect "#Sop_Informatica"; #Redireccion
};
```

En este caso lo que hacemos es redireccionar un canal a otro, pero también podemos bloquear canales con diversos nombres por ejemplo #Sexo. Y sin indicar redirect.

**Filtrado de palabras.** (Opcional)

```
badword all {
    word "sexo"; #Palabra mala a tratar.
    replace "censurado"; #Palabra por la cual la remplazamos.
    action replace; #Indicamos la accion a realizar.
};
```

All Indica a que afecta la censura por defecto all indica todo pero puede ser cambiado por:

**Channel** para Mensajes en el canal público.

**Message** para Mensajes en el canal Privado.

**Quit** para Mensajes de despedida.

**All** para Todos los mensajes.

Action puede tener los valores replace (para reemplazar) o block para bloquear).

Ejemplo:

```
badword all  {
    word "puta"; #Palabra mala a tratar.
    action block; #Indicamos la accion a realizar.
};
```

Donde da este resultado:

```
jetcode puta
      * #Sop_Informatica :Swearing is not permitted in this channel (#Sop_Informatica)
jetcode puta
      * #Sop_Informatica :Swearing is not permitted in this channel (#Sop_Informatica)
jetcode puta
      * #Sop_Informatica :Swearing is not permitted in this channel (#Sop_Informatica)
```

## *Filtrado de malas palabras para los usuarios.*

E la directiva set hay que añadir la opción
"modes-on-connect "+ixwG";"

```
23  set {
24      modes-on-connect "+ixwG";
25      network-name "Red empresa"; #Define el nombre de la red
26      default-server      "localhost";  #Servidor por defecto de la red.
27      services-server     "localhost";  #Servidores de servicios de terc
28      stats-server        "localhost"; #Servidor de estadisticas IRC
```

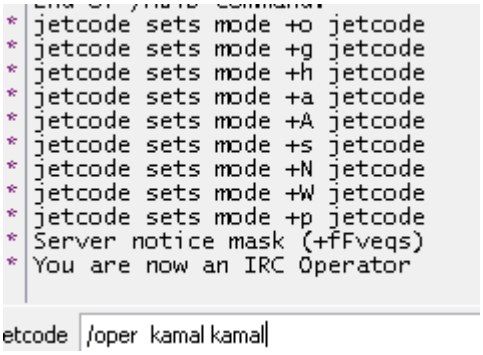
Una vez hecho se debe reiniciar al servidor.

Y el resultado es este.

```
*** OperOverride -- jetcode (jetcode@172.16.0.1) MODE #Sop_Informatica +G
*** Herimax (Joni12Pack@172.16.0.1) did a /whois on you.
*** Herimax (Joni12Pack@172.16.0.1) did a /whois on you.
Herimax does not accept private messages containing swearing.
```

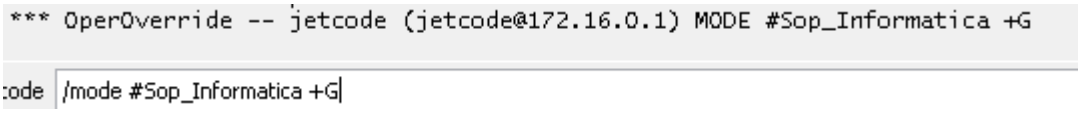
## *Filtrado de un canal*

Nos ponemos como Ircop.

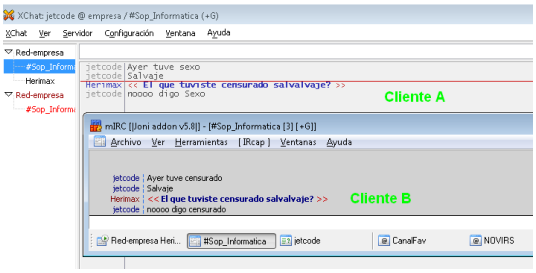



Y donde 1º va el usuario y luego la contraseña.

Ahora cambiamos el modo del canal a +G para que se filtren las palabras




Una vez hecho el resultado será este.




## *Canales predefinidos.*

Quizás sea buena idea la de crear canales predefinidos. Para ello usamos la directiva

official-*channels {*
*"#Nombre_canal" { topic "Un mensaje"; };*
*};*

EL nombre del canal indica el nombre que se le dará a este y el topic es un resumen del canal contiene información relativa al canal.

Ejemplo:

```
official-channels {
  "#Sop_Informatica" { topic "Pide ayuda a un informatico"; };
  "#Ventas" { topic "Departamento de Ventas"; };
  "#Produccion" { topic "Departamento de Produccion"; };
  "#Mantenimiento" { topic "Departamento de Matenimiento"; };
  "#Limpieza" { topic "Departamento de Limpiza"; };
};
```

Entonces cuando ejecute el comando /list vera esto:

```
Channel          Users   Topic
#Limpieza        0       Departamento de Limpiza
#Mantenimiento   0       Departamento de Matenimiento
#Produccion      0       Departamento de Produccion
#Ventas          0       Departamento de Ventas
#Sop_Informatica 3
End of /LIST
```

## *Filtros AntiSpam.*

Es muy habitual recibir ataques de robots que lo que hacen es mandar publicidad a los usuarios del IRC una contramedida muy rápida es colocar filtros AntiSpam. Para ello se dispone de la directiva.

```
spamfilter {
    regex "www"; #palabra a cazar.
    target { private; channel; }; #Donde buscar
    action gline; # Acccion a tomar
    reason "Publicidad no permitida"; #Mensaje a mostrar.
    ban-time 6h; #Tiempo de baneo.
};
```

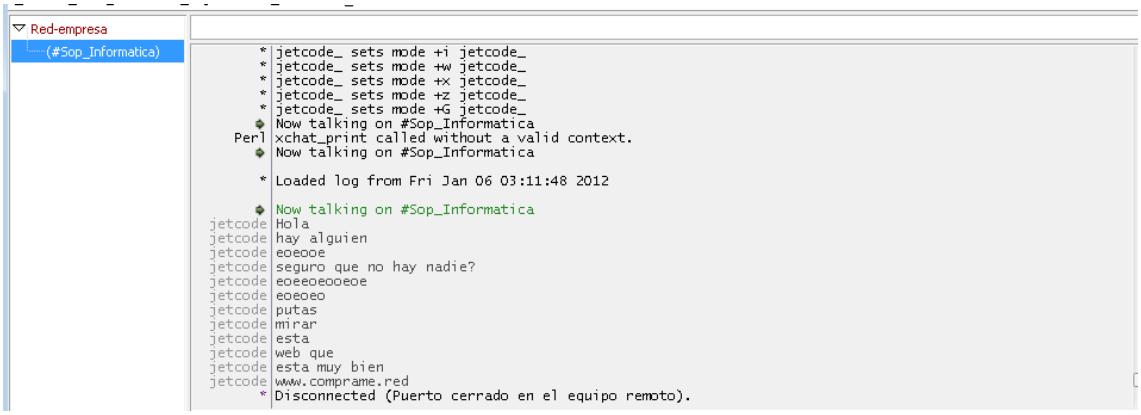
**Target** indica donde buscar que pueden ser:

**Channel:** Mensaje a canal.
**prívate**: Mensaje privado de usuario a usuario.
**user:** *Nombre de usuario.*
**away:** **En el** Mensaje temporal cuando está inactivo.
**user:** Nick de usuario.

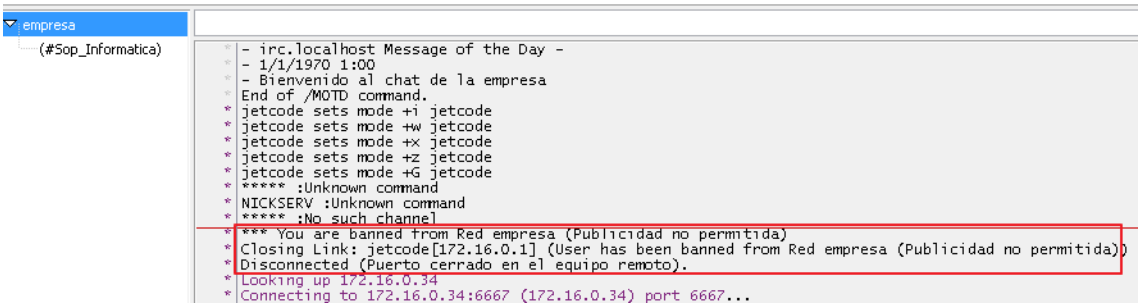
Acción indica la acción a tomar y puede tomar los valores.

**kill** desconecta al usuario
**tempshun** silencia la sesión actual del usuario (si reconecta el silencio se va)
**shun** pone un silencio al host
**kline** Banea la máscara host localmente.
**gline** Deniega el acceso al host en toda la red del IRC.
**zline** Banea una dirección IP del servidor local. No afecta a la red
**gzline** Deniega el acceso al host en toda la red del IRC.
**block** Sólo bloquea el mensaje.
**dccblock** Marca el usuario para no poder enviar más DCCs.
**viruschan** Fuerza su salida de todos los canales, y desactiva todos los comandos enviados.

El resultado es este:




Y si miramos el log del cliente se aprecia la razón.




## *Activar el cifrado SSL en el servidor.*

Para activar el soporte SSL debemos modificar el fichero de configuración y como primera medida debemos decirle al servidor que acepte conexiones SSL en dicho puerto ,es decir añadir la directiva SSL en la sección options del puerto deseado ejemplo:

```
listen *:6667 { #Abrimos un puerto en todas las interfaces.
    options { #Indicamos las Opciones de escucha
        clientsonly; #Solo se pueden conectar clientes.
        ssl; #Activamos el soporte SSL
        };
};
```

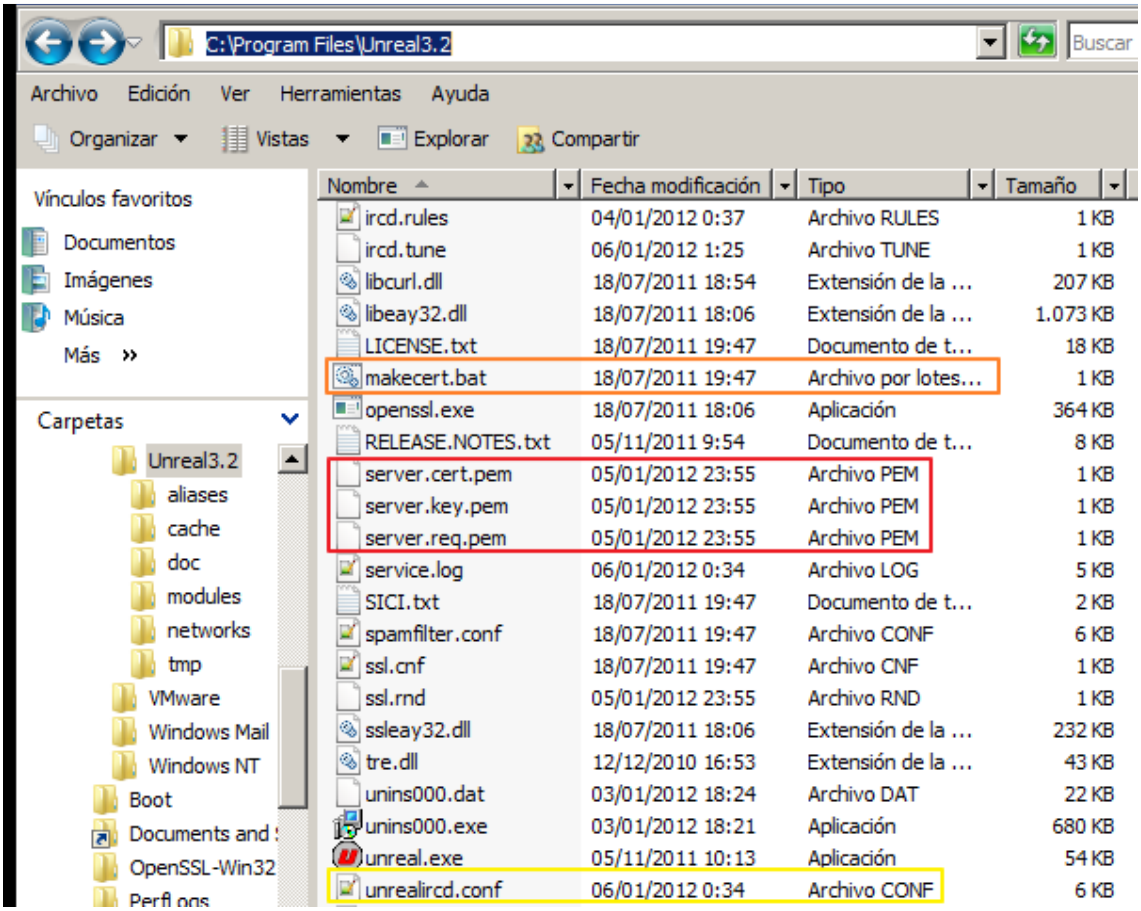
Una vez hecho debemos ir a la sección set y añadir la directiva SSL para indicarle la llave privada y la llave publica que se van a usar para firmar y cifrar los datos.

Quedando así:

```
23  set {
24  ssl {
25          certificate server.cert.pem;
26          key server.key.pem;
27          };
28      modes-on-connect "+ixwG";
```

Donde certifícate indica el fichero que contiene la llave pública y luego con key indicamos el fichero que contiene la llave privada. Opcionalmente podemos usar la opción trusted-ca-file para indicar el fichero de la autoridad certificadora.

Los archivos de la llave pública y la privada indicados en el fichero deben estar en la carpeta donde está el archivo de configuración. (En recuadro rojo las llaves) y en recuadro amarillo el fichero de configuración.




En caso de no disponer las llaves podemos usar el script makecert.bat situado en la carpeta del programa (recuadro naranja) para general las claves privada y pública para SSL.

Para crear un certificado SSL bajo Linux usamos los siguientes comandos.

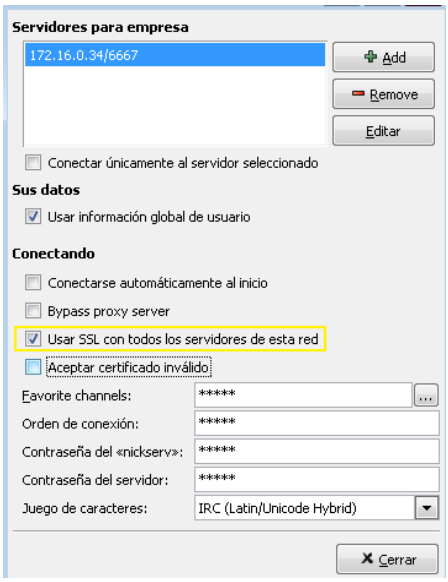
1º Generamos la llave privada.
Comando: **openssl genrsa 1024 > llave.key**
2º Generamos la llave pública.
Comando: **openssl req -new -x509 -key llave.key > publica.key**

Y rellenamos los datos.

```
Country Name (2 letter code) [AU]:sp
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:
Email Address []:
```

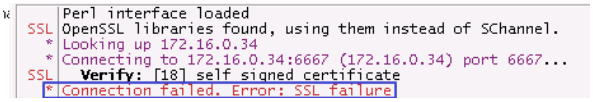
## *Activar el cifrado SSL en el cliente.*

Para Xchat.



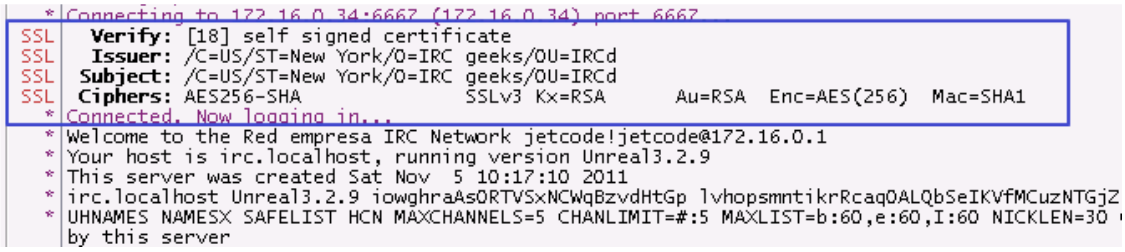

Una vez activa podemos conectarnos y en

En caso de conectarnos y ver este mensaje.




Debemos activar la opción aceptar certificado invalido, eso es debido a que no está firmado por una CA de pago.

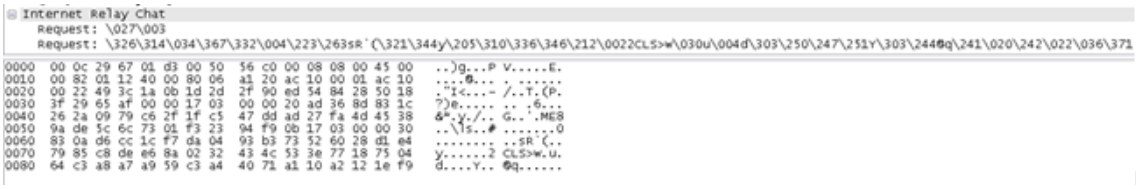
Una vez marcada la opción podemos conectarnos.




En el caso de Mirc se deberá descargarse la extensión openssl que está disponible en la página web. Véase la siguiente pagina web http://www.mirc.com/ssl.html

- Nota: Mirc no soporta IPV6 de forma nativa, el auto no tiene ganas de implementarlo pero tiene hay una extensión para funcionar sobre dicho protocolo. Llamada mircv6loader.

En la siguiente imagen se aprecia como realmente el trafico es cifrado bajo SSL.



## 4.0 Recursos y Bibliografía.

Documentación Oficial de UnrealIRCD
http://www.unrealircd.com/files/docs/unreal32docs.html

Lista de comandos IRC.
http://personales.mundivia.es/papi/comirc.html

Historia del protocolo IRC
http://www.sindominio.net/quique/Traducciones/historia_irc.html