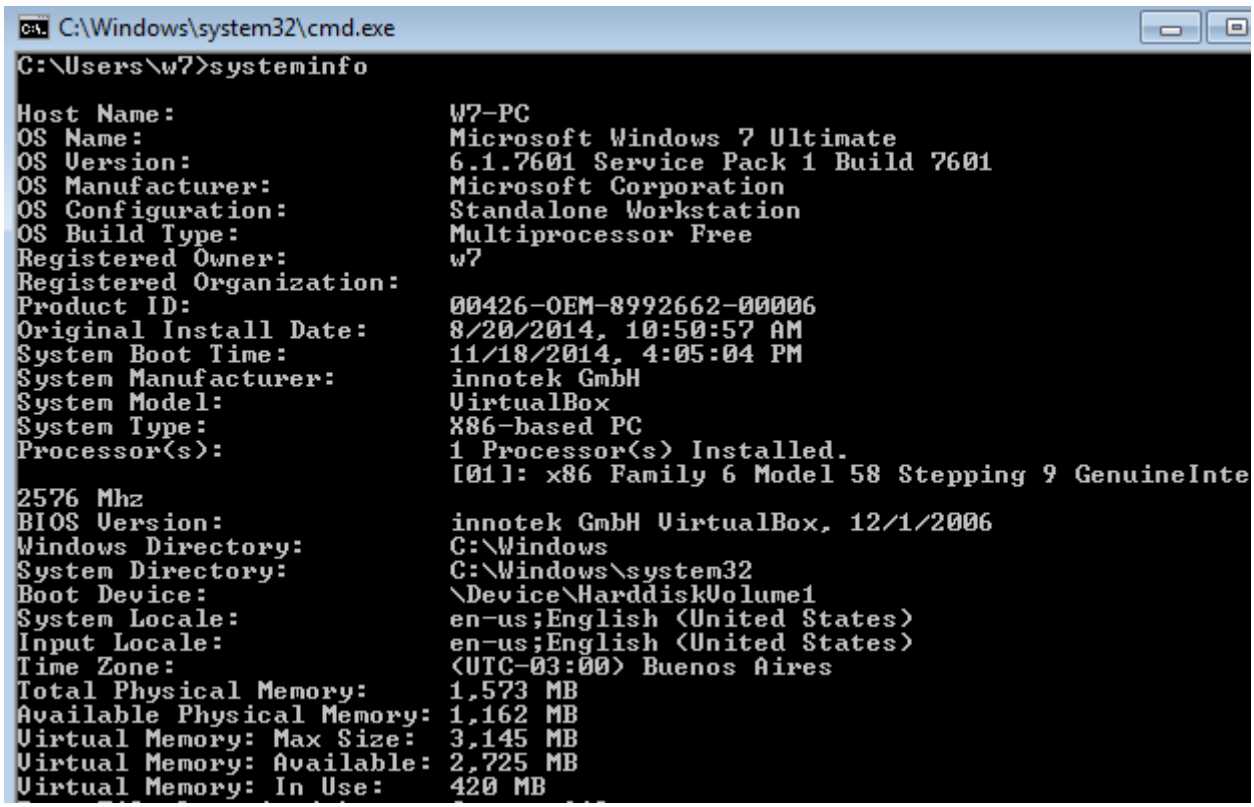
Llegamos a la segunda entrega de esta serie de comandos. Me di cuenta que no voy a poder hacer todos los que quisiera, pero sí bastantes. Ampliaré con más clases porque será necesario:)

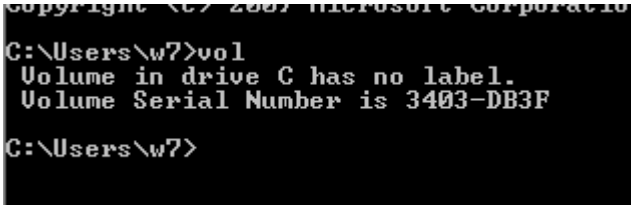
El primer comando que veremos la clase de hoy es ***systeminfo***. Si lo tecleamos en la consola, nos mostrará toda **información del sistema** tales como el sistema operativo, cosas de hardware, una lista de parches y configuraciones de la red.



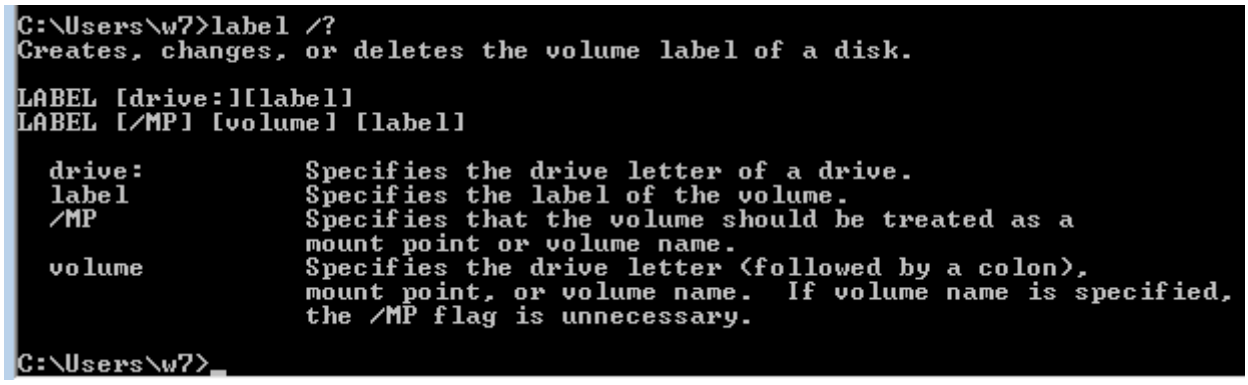
Siguiendo con el tipo de instrucciones informativas, podemos encontrar ***hostname*** que no hace más que devolver el **nombre** de nuestro **host**. Es decir de la máquina en la que estamos sentados.

Útil para ciertas herramientas de red, y no tener que recordar una IP que, además, puede ser dinámica.

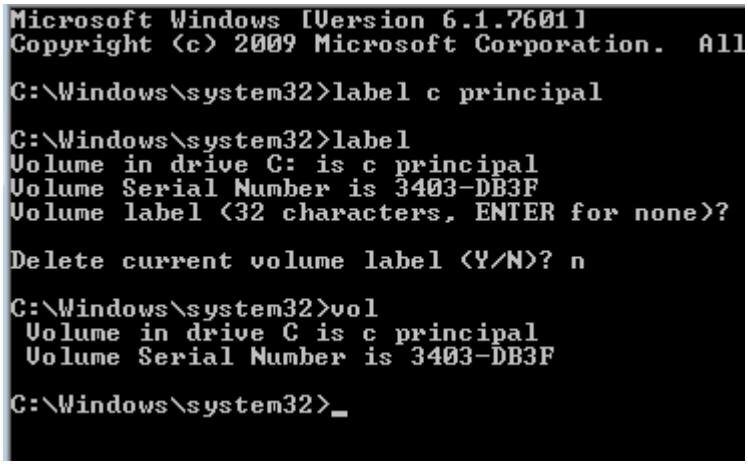
Si hacemos ***vol <parametro>*** donde el parámetro corresponde a una partición (C, D, E, etc), podemos obtener pequeña información. El **label** es un **nombre** que se le puede dar a una partición -me siento recursivo- para el simple hecho de facilidad del usuario.



¿Y cómo cambiamos el nombre para ponerle el que se nos cante? Usemos ***label***. Pero no recuerdo exactamente como usar la instrucción, así que como parámetro uso **/?** para imprimir en pantalla la ayuda. Pensemos en cada **comando** como un programa, o un **software** y que como todo software tiene **modos de uso** y **ayuda**. Nada más que en vez de tener un entorno gráfico, viene por texto.

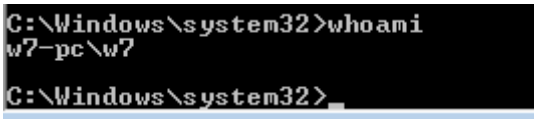


Sí, en **inglés**. Ya sé, hay muchos que se rehúsan a aprender éso pero para este trabajo es necesario. Además piensen como un nuevo aprendizaje. Y si no, aprendan ruso o chino que también hay cosas muy interesantes. Sobre todo el malware ruso. ¿Qué tendrán contra nosotros?
Entonces "***label [drive:] [label]***" (entre corchetes son los parámetros). El **drive** será nuestro disco, o C en mi caso, y **label** será el nombre.



Vemos que si despues usamos ***label*** sin parámetros, nos dice para poner el disco sin este nombre. Elegí que no para que vean que cuando volvemos a usar el comando ***vol*** nos aparece **principal**.

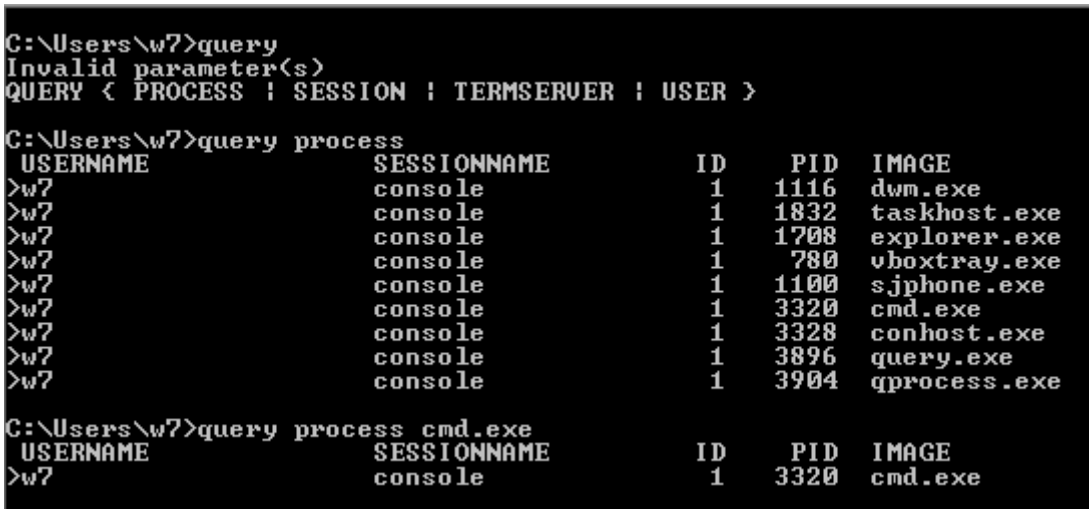
El ***whoami*** (por inglés "Who Am I?" o "¿Quién soy?") nos dirá nuestro usuario. Cuando hagan alguna prueba de penetración a una máquina, van a querer llegar a ser **Administrador** o **System** que son los usuarios con más privilegios.



A parte, si queremos **cerrar la sesión** desde la consola, podemos usar ***logoff*** sin parámetros. No se los puedo mostrar con una imagen y no tengo manera de poner un .gif ^^.

Ahora les voy a mostrar un comando muy interesante que anteriormente eran muchos hasta que se unificó en uno solo. ***Query*** es el utilizado para sacar información del sistema de procesos, usuarios, sesiones o los conectados al terminal server -ahora no nos enfoquemos en qué es cada cosa en profundidad-.

Si lo hacemos sin parámetros nos aparece un poco de ayuda. Nuestro parámetro sera **process** (no lo intenté, pero creo que con o sin mayúsculas funciona igual) donde nos listará los procesos con el identificador o **PID** (process id) por si lo necesitamos. Si ponemos también el nombre del proceso, sólo nos listará ése o también podemos poner el número de id en vez del nombre.



Con ***query session*** podemos ver las sesiones activas con sus servicios y con ***query user*** podemos ver los usuarios activos, y así con las cuatro opciones que tenemos (hagan la prueba en sus casas). Perdón si no les explico demasiado pero no se preocupen que cuando sea el momento lo van a saber.

```
C:\Users\w7>query session
 SESSIONNAME       USERNAME                 ID  STATE   TYPE        DEVICE
 services                                    0  Disc
>console           w7                        1  Active

C:\Users\w7>
```

Otro de los comandos que podemos usar es **runas** ("run as"). Éste nos facilita la idea de correr un programa cualquiera desde cualquier sesión, por otro usuario con otros **privilegios**. Muy bueno para usar una sesión de invitado o de **pocos privilegios** por **seguridad** (no es lo mismo correr un virus con privilegios altos que bajos) y correr los programas que nosotros queramos con una cuenta de **administrador**.

La instrucción a hacer sería "**runas /user:nombredeusuario rutacompletadelejecutable**"
En caso de que el ejecutable tenga espacios en los nombres de alguna carpeta o en si mismo, es obligatorio encerrar todo entre comillas

```
C:\Users\Guest>runas /user:w7 c:/Windows/notepad.exe
Enter the password for w7:
Attempting to start c:/Windows/notepad.exe as user "W7-PC\w7" ...

C:\Users\Guest>_
```

Cuando les pida la contraseña del usuario, se darán cuenta que no hay asteriscos ni nada. Esto es así para que si alguien está mirando, no sepa cual es la longitud de esta clave. Digamos que se quedará en negro como si no estuvieran tipeando nada, pero quédense tranquilos.

Ahora pasemos un poco a los comandos que nos posibilitan la información de la red.
Primero, el comando ***getmac***, que nos muestra la dirección MAC de nuestra/s tarjeta/s de red.

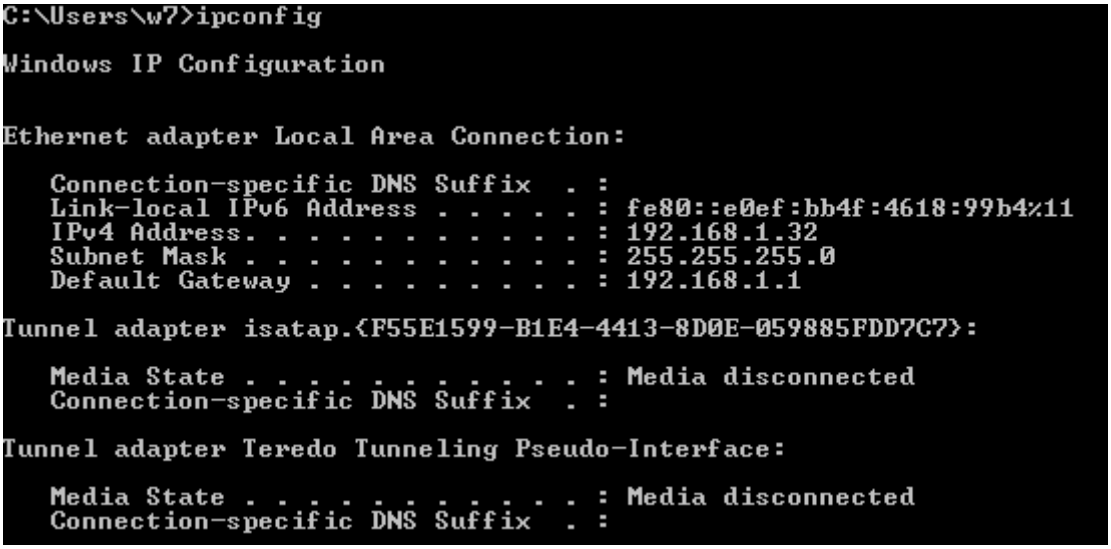
```
C:\Users\w7>getmac

Physical Address    Tra
=================== ===
08-00-27-9B-BC-90   \De
```

Pero no sólo podemos conseguir nuestra dirección, sino también la de un ordenador remoto que esté en la misma red. No puedo hacer la prueba ahora mismo por problemas técnicos, pero el comando sería "**getmac /S Ipdehostremoto**". Si no sabemos que tarjeta de red es, podemos usar el parámetro **/V** (por Verbose, hay muchos programas que usan la V como **verbose** para entregar más información).

```
C:\Users\w7>getmac /V

Connection Name Network Adapter Physical Address

=============== =============== ===================
============================
Local Area Conn Intel(R) PRO/10 08-00-27-9B-BC-90
-4413-8D0E-059885FDD7C7}
```

La próxima es ***ipconfig*** que nos imprimirá en pantalla mucha información local de la red a la que estamos conectados.

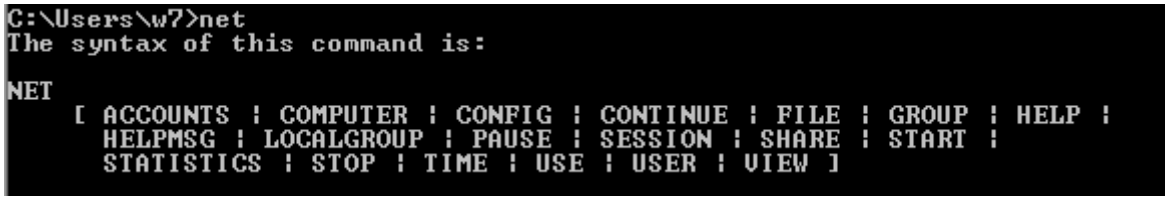
Fíjense que me nota 2 medios desconectados, y 1 ethernet (por red cableada) conectado. En la última nos da mucha info. Tenemos la dirección **IPv6**, **IPv4**, la **máscara** de red y la dirección de **gateway**.



Si usamos el parámetro "***/release"*** nos limpiará la información que está puesta. En caso de tener un **dhcp**, no estoy muy seguro pero creo que luego de un tiempo le pedirá que le entregue una IP. En caso de que no lo haga, podemos usar el parámetro "***/renew***" así lo hacemos manualmente. Algunas veces es necesario para saber si el **dhcp** está funcionando correctamente o si hubo una colisión de IP's entre dos dispositivos.

Si tuviésemos un **servidor** conectado a una red de computadoras y pudiésemos administrarla desde allí, tenemos el comando ***net*** que en conjunto con parámetros, podemos configurarla.

Siempre irá acompañado de alguno de ellos. Miremos cuáles hay.



Muchas opciones interesantes, pero veamos una por una :)

net accounts

Con ésta, podemos ver opciones de las cuentas. Algo así como cuánto **tiempo** puede estar conectado, cual es el **mínimo largo** de la contraseña, cuánto puede durar sin ser cambiada esa contraseña, etc. Si hacemos ***net accounts /?*** obtendremos más info.

```
C:\Users\w7>net accounts
Force user logoff how long after time expires?:       Never
Minimum password age (days):                          0
Maximum password age (days):                          42
Minimum password length:                              0
Length of password history maintained:                None
Lockout threshold:                                    Never
Lockout duration (minutes):                           30
Lockout observation window (minutes):                 30
Computer role:                                        WORKSTATION
The command completed successfully.
```

Net computer

Para agregar o quitar **equipos** de la red. Si entrtamos y tenemos los máximos privilegios, podríamos obtener información de todos los equipos conectados. Hace falta un poquito más de teoría para comprender al cien este comando pero ya lo veremos.

```
C:\Users\w7>net computer
The syntax of this command is:

NET COMPUTER
\\computername {/ADD | /DEL}
```

Net config

Con éste podemos ver los **servicios configurables** que están en **ejecución** o configurarlos en el caso de que se pueda.

```
C:\Users\w7>net config
The following running services can be controlled:

   Server
   Workstation

The command completed successfully.


C:\Users\w7>net config Workstation
Computer name                        \\W7-PC
Full Computer name                   w7-PC
User name                            w7

Workstation active on
        NetBT_Tcpip_{F55E1599-B1E4-4413-8D0E-059885FDD7C7} (0800279BBC90)
```

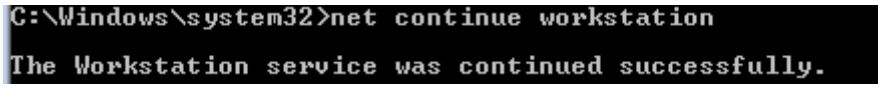
Net pause

Es para **pausar servicios** que están corriendo. Fíjense que para poder hacer esto hay que correr la terminal de comandos con permisos de **administrador**. De otra manera nos dira que es acceso denegado. Intentemos con el servicio **workstation**, poniendo éste como último parámetro.

```
C:\Windows\system32>net pause workstation

The Workstation service was paused successfully.
```

Net continue

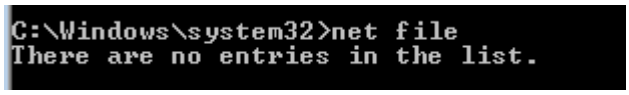
Exactamente para "despausar" o **recontinuar** el proceso que estaba corriendo ante de pausado.



Net file

Como estamos trabajando en la red, nos daremos cuenta que se usa para poder monitorear los **archivos compartidos** que permanecen **abiertos** o si existe alguno bloqueado. También se pueden cerrar los archivos desde aquí.

Yo no tengo ninguno y por eso en la imagen no aparecen.

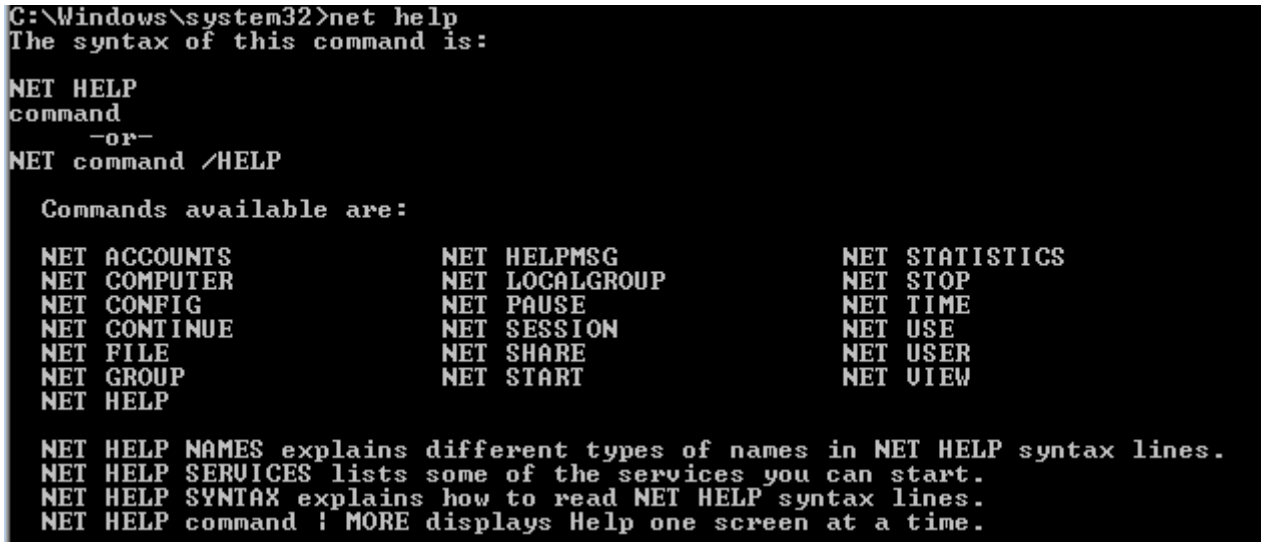


Net group

Administra grupos de **dominio**. **Los dominios son grupos de red que confían en un servidor principal** (como si fuese su dios de los privilegios y la administración). Sólo para Windows Server.

Net help

Como sabemos, nos mostrará ayuda y como podemos consultar por msa documentación por cada parámetro.



Net helpmsg

Bueno, llegamos a un comando que a mi me gusta mucho personalmente. Sirve para que nos de **información** de un número de **error**. Eso sí, está muy simplificado y me gustaría que venga con más data, pero la idea es encantadora.

```
C:\Windows\system32>net helpmsg 123

The filename, directory name, or volume label syntax is incorrect.
```

Net localgroup

Llega al mismo término que **group** pero en el equipo **local**. Por defecto nos vienen varios, así que tendran una buena lista.

```
C:\Windows\system32>net localgroup

Aliases for \\W7-PC

-------------------------------------------

*Administrators
*Backup Operators
*Cryptographic Operators
*Distributed COM Users
*Event Log Readers
*Guests
*IIS_IUSRS
*Network Configuration Operators
*Performance Log Users
*Performance Monitor Users
*Power Users
*Remote Desktop Users
*Replicator
*Users
The command completed successfully.
```

Net session

Muestra una **lista** (también puede **desconectar**) de **usuarios** remotos **conectados** a un **equipo local**. Como yo no tengo ninguno conectado, no aparecerá ninguno.

```
C:\Windows\system32>net session
There are no entries in the list.
```

Net share

Este comando es el que maneja los **recursos compartidos**. En la columna izquierda tenemos el nombre, el el medio qué es lo que se esta compartiendo y en la derecha aparece un comentario que puede ser cualquier cosa.

```
C:\Windows\system32>net share

Share name   Resource                        Remark

-------------------------------------------------------------------------------
C$           C:\                             Default share
IPC$                                         Remote IPC
ADMIN$       C:\Windows                      Remote Admin
The command completed successfully.
```

Net start

Ya suponen, bien seguramente, que éste puede **mostrar** los procesos que estan **empezados** (sin parámetros) o **empezar** alguno que este **detenido** (dándole el nombre del servicio).

```
C:\Windows\system32>net start
These Windows services are started:

   Application Information
   Base Filtering Engine
   COM+ Event System
   Cryptographic Services
   DCOM Server Process Launcher
   Desktop Window Manager Session Manager
   DHCP Client
   Diagnostic Policy Service
   Diagnostic Service Host
   Distributed Link Tracking Client
   DNS Client
   Foxit Cloud Safe Update Service
   Group Policy Client
   IKE and AuthIP IPsec Keying Modules
   IP Helper
   IPsec Policy Agent
   Network Connections
```

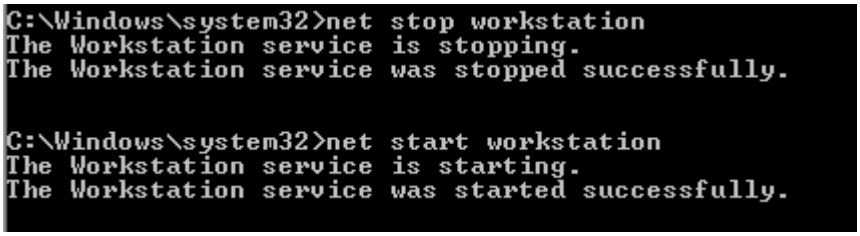
Net statistics

Aquí podemos ver **estadísticas** de los **servicios server** o **workstation.**

```
C:\Windows\system32>net statistics workstation
Workstation Statistics for \\W7-PC


Statistics since 11/28/2014 8:35:42 AM


  Bytes received                            1922
  Server Message Blocks (SMBs) received     16
  Bytes transmitted                         1600
  Server Message Blocks (SMBs) transmitted  15
  Read operations                           0
  Write operations                          0
  Raw reads denied                          0
  Raw writes denied                         0

  Network errors                            0
  Connections made                          2
  Reconnections made                        0
  Server disconnects                        0

  Sessions started                          0
```

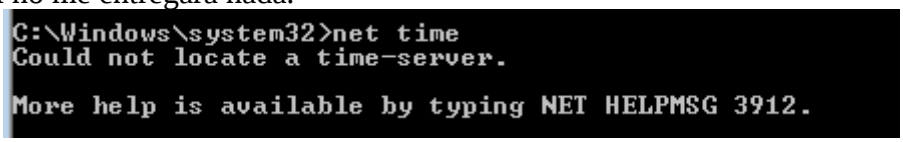
Net stop

El antítesis de **start**. Detiene el servicio que sea. En el ejemplo detuve y reinicie el servicio de **workstation**.



Net time

En caso de que tengamos un **servidor** que nos entregue la **hora**, podemos **sincronizarla** desde este comando. A mi no me entregará nada.



Net use

Administra los recursos compartidos. Si lo hacemos sin parámetros nos muestra info. Yo tengo una carpeta compartida por la máquina host.

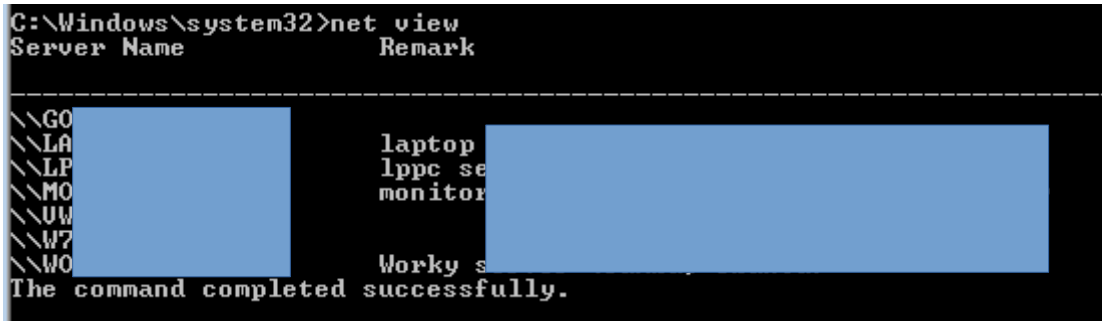
Net user

Se puede **agregar**, **modificar** o ver **información** de **usuario**.

```
C:\Windows\system32>net user

User accounts for \\W7-PC

-------------------------------------------------------------------------------
Administrator            Guest                    password
w7
The command completed successfully.


C:\Windows\system32>net user /?
The syntax of this command is:

NET USER
[username [password | *] [options]] [/DOMAIN]
         username {password | *} /ADD [options] [/DOMAIN]
         username [/DELETE] [/DOMAIN]
         username [/TIMES:{times | ALL}]
```

Net view

Muestra una **lista** de **equipos** en la red que **comparten recursos**. Tapo lo que hay en la red porque los conozco. ¡Sin verguenzas! :P



Hasta aquí le dimos al comando **net**, pero vamos a seguir con los próximos. Aún queda bastante camino por recorrer.

Uno de los comandos que más nos interesan a nosotros es ***netstat***. ¿Qué es lo que hace? Simple. Muestra información de las conexiones de un equipo. Tiene muchos parámetros para poder dar información de ésto. Vamos a investigarlo a fondo:). Recuerden que con el comando y **/?** pueden ver la ayuda de éste.

```
C:\Users\w7>netstat /?

Displays protocol statistics and current TCP/IP network connections.

NETSTAT [-a] [-b] [-e] [-f] [-n] [-o] [-p proto] [-r] [-s] [-t] [interval]

  -a            Displays all connections and listening ports.
  -b            Displays the executable involved in creating each connection or
                listening port. In some cases well-known executables host
                multiple independent components, and in these cases the
                sequence of components involved in creating the connection
                or listening port is displayed. In this case the executable
                name is in [] at the bottom, on top is the component it called,
                and so forth until TCP/IP was reached. Note that this option
                can be time-consuming and will fail unless you have sufficient
                permissions.
  -e            Displays Ethernet statistics. This may be combined with the -s
                option.
  -f            Displays Fully Qualified Domain Names (FQDN) for foreign
                addresses.
  -n            Displays addresses and port numbers in numerical form.
  -o            Displays the owning process ID associated with each connection.
  -p proto      Shows connections for the protocol specified by proto; proto
                may be any of: TCP, UDP, TCPv6, or UDPv6.  If used with the -s
                option to display per-protocol statistics, proto may be any of:
                IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, or UDPv6.
  -r            Displays the routing table.
  -s            Displays per-protocol statistics.  By default, statistics are
                shown for IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, and UDPv6;
                the -p option may be used to specify a subset of the default.
  -t            Displays the current connection offload state.
  interval      Redisplays selected statistics, pausing interval seconds
                between each display.  Press CTRL+C to stop redisplaying
                statistics.  If omitted, netstat will print the current
                configuration information once.
```

Suponiendo que lo usamos a secas **sin** poner **parámetros**, nos saldrán las **conexiones activas** que existen en este momento.

```
C:\Users\w7>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.1.32:49163     192.168.1.10:52323     TIME_WAIT
  TCP    192.168.1.32:49165     192.168.1.10:52323     TIME_WAIT
  TCP    192.168.1.32:49167     192.168.1.10:52323     TIME_WAIT
  TCP    192.168.1.32:49169     192.168.1.10:52323     TIME_WAIT
  TCP    192.168.1.32:49170     192.168.1.10:52323     TIME_WAIT
  TCP    192.168.1.32:49171     192.168.1.10:52323     TIME_WAIT
```

Analizemos un poco el resultado. En la **primer columna** nos sale el **protocolo** de comunicación que existe en ese puerto (tcp, udp. ¿Recuerdan?); en la **segunda**, nos aparecera la IP y el puerto **locales** de la **comuncación**; en la **tercera** tenemos las IP y puertos **destino**. Por último, en la última columna tenemos el estado de la comunicación. Generalmente sin parámetros muestra muchas conexiones establecidas o **established**, pero aquí nos encontramos con "**TIME_WAIT**", que significa que el host **local** está **cerrando** la **conexión** (recuerden que para cerrar la conexión tenemos que mandar tramas con ciertas características. Si dejamos de mandar datos la conexión seguirá establecida hasta que digan lo contrario o alguno de los dos se quede sin contacto).

Con el parámetro -***b***, la instrucción **analiza** los **binarios** o programas que hicieron el intercambio de datos. En este caso, abrí el navegador Firefox y entré a la página de Google. Miren el resultado.

```
C:\Windows\system32>netstat -b

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:49172        w7-PC:49173            ESTABLISHED
 [firefox.exe]
  TCP    127.0.0.1:49173        w7-PC:49172            ESTABLISHED
 [firefox.exe]
  TCP    192.168.1.32:49174     ec2-54-213-3-31:https  ESTABLISHED
 [firefox.exe]
  TCP    192.168.1.32:49175     eze03s05-in-f16:https  TIME_WAIT
  TCP    192.168.1.32:49176     eze03s05-in-f7:http    ESTABLISHED
 [firefox.exe]
  TCP    192.168.1.32:49177     snippets:https         TIME_WAIT
  TCP    192.168.1.32:49178     72.21.91.29:http       TIME_WAIT
  TCP    192.168.1.32:49179     72.21.91.29:http       ESTABLISHED
 [firefox.exe]
  TCP    192.168.1.32:49180     server-54-192-56-136:https  ESTABLISHED
 [firefox.exe]
```

Con el parámetro -***e*** vemos **estadísticas** de la conexión **ethernet**.

```
C:\Windows\system32>netstat -e
Interface Statistics

                           Received            Sent

Bytes                      9539589           553549
Unicast packets               8202             5216
Non-unicast packets           2112              981
Discards                         0                0
Errors                           0                0
Unknown protocols                0
```

El -***n*** hará que la parte de puertos sea representación en modo **númerico** y no en modo texto.
He aquí las diferencias, en este caso en la columna de **foreign address**:

```
C:\Windows\system32>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:49172        w7-PC:49173            ESTABLISHED
  TCP    127.0.0.1:49173        w7-PC:49172            ESTABLISHED

C:\Windows\system32>netstat -n

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:49172        127.0.0.1:49173        ESTABLISHED
  TCP    127.0.0.1:49173        127.0.0.1:49172        ESTABLISHED
```

También podemos ver los **PID**, agregando el parámetro -***o***. Simplemente agrega la columna. Fíjense como un proceso (nos damos cuenta porque más de una fila comparten el valor de PID) puede tener varias conexiones.

```
C:\Windows\system32>netstat -o

Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    127.0.0.1:49172        w7-PC:49173            ESTABLISHED     2936
  TCP    127.0.0.1:49173        w7-PC:49172            ESTABLISHED     2936
```

Podemos **filtrar** la salida, agregando el parámetro ***-p <protocolo>*** por los **protocolos** UDP, TCP o IP.

```
C:\Windows\system32>netstat -p TCP

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:49172        w7-PC:49173            ESTABLISHED
  TCP    127.0.0.1:49173        w7-PC:49172            ESTABLISHED
```

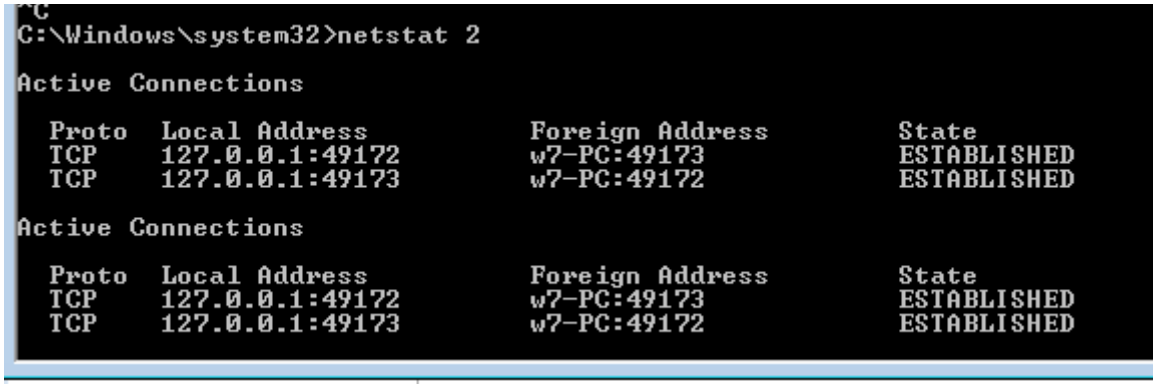
Con ***-r*** visualizaremos las **tablas de enrutamiento**. Ésto es una **base de datos** que **almacena** el dispositivo para saber qué **camino** tomar **hasta** cierto **otro nodo** de la red. Y si están pensando ya como un atacante, les diré que sí. Se pueden cambiar estas rutas **a mano**. Quizas alguien se quiera meter en el medio entre tú y el router de salida. ;)

```
C:\Windows\system32>netstat -r
===========================================================================
Interface List
 11...08 00 27 9b bc 90 ......Intel(R) PRO/1000 MT Desktop Adapter
  1...........................Software Loopback Interface 1
 12...00 00 00 00 00 00 00 e0 Microsoft ISATAP Adapter
 13...00 00 00 00 00 00 00 e0 Teredo Tunneling Pseudo-Interface
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0    192.168.1.1    192.168.1.32     10
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    306
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    306
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    306
      192.168.1.0    255.255.255.0         On-link    192.168.1.32    266
     192.168.1.32  255.255.255.255         On-link    192.168.1.32    266
    192.168.1.255  255.255.255.255         On-link    192.168.1.32    266
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    306
        224.0.0.0        240.0.0.0         On-link    192.168.1.32    266
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    306
```

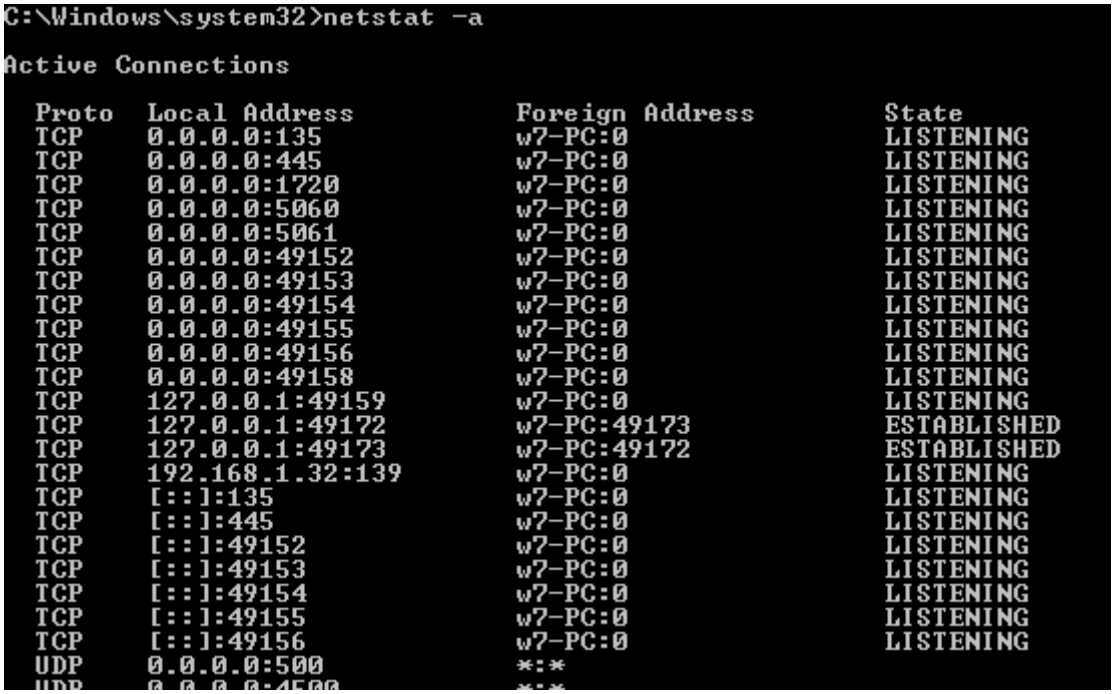
Pero no profundizemos tanto. Sigamos con el comando ***netstat***. Para ver **estadísticas** de la comunicación de los **protocolos**, usemos ***-s***. Es algo extenso y detallista.

```
C:\Windows\system32>netstat -s

IPv4 Statistics

  Packets Received                   = 3653
  Received Header Errors             = 0
  Received Address Errors            = 1
  Datagrams Forwarded                = 0
  Unknown Protocols Received         = 0
  Received Packets Discarded         = 129
  Received Packets Delivered         = 3899
  Output Requests                    = 2247
  Routing Discards                   = 0
  Discarded Output Packets           = 0
  Output Packet No Route             = 6
  Reassembly Required                = 0
  Reassembly Successful              = 0
  Reassembly Failures                = 0
  Datagrams Successfully Fragmented  = 0
  Datagrams Failing Fragmentation    = 0
  Fragments Created                  = 0

IPv6 Statistics
```

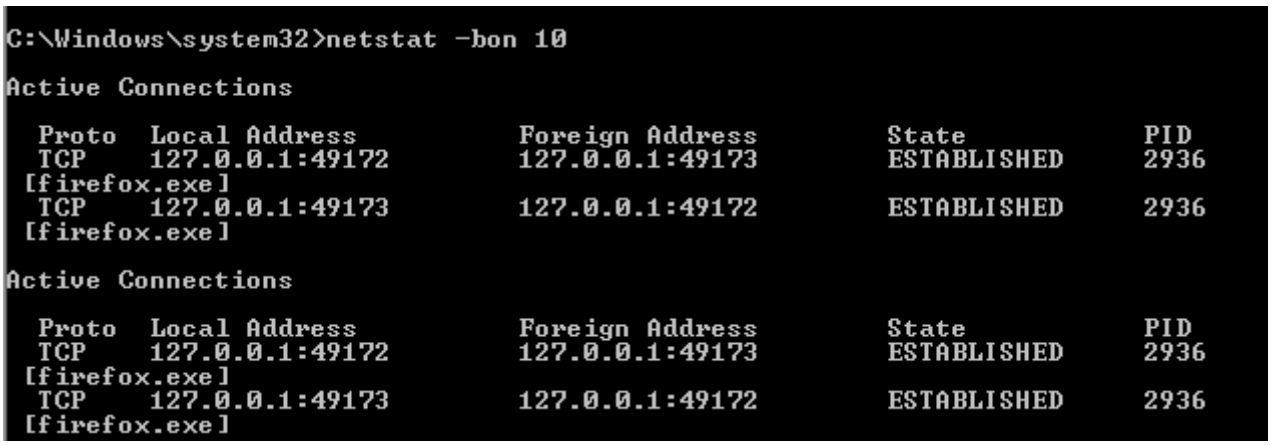
La opción *-**f*** la veremos más adelante que todavía no es necesario. Luego de todos los parámetros, podemos poner un número, que estará representado en **segundos**, y que le dirá al comando que se vuelva a **repetir** cada cierta cantidad de tiempo. Se denomina **intervalo**. Se puede parar con las teclas Ctrl+C.



Si usan *-**a*** pueden ver todas las conexiones.



Y también pueden **combinar** los **parámetros** poniendo un solo guión y varias letras, más el intervalo. Por ejemplo quiero ver qué programas lo usan, los PID, que lo muestre de manera numérica y que lo haga cada 10 segundos.



Esto puede hacerse para mostrar toda la info que uno necesita:) La idea es que a uno le sirva lo que va a mostrar.

Todavía faltan cosas pero voy a dejar aquí y seguir en la próxima para ya poder subir esta clase. Saludos futuros hackers!

Pueden seguirme en Twitter: @RoaddHDC

Cualquier cosa pueden mandarme mail a: r0add@hotmail.com

Para donaciones, pueden hacerlo en bitcoin en la dirección siguiente: 1HqpPJbbWJ9H2hAZTmpXnVuoLKkP7RFSvw

Roadd.
