Manipulación de logs

Contenido

Manipulación de Logs

La información de salida que muestran los comandos **"debug"** varían según su tipo. Algunos comandos generan solo una sola línea de salida por paquete mientras que otros generan múltiples líneas de salida, existen otros que generan líneas ocasionalmente, por lo tanto, la forma en la que la salida del comando debug es documentada también varia.

Nota: La salida del debug que genera solo una línea de texto usualmente se conoce como line by line y la salida del comando debug que genera una salida en múltiples líneas se le conoce como formato de campo y comúnmente se describe como tablas.

Por defecto, los dispositivos de red mandan la salida del debug a la consola. Enviar la salida del debug a una terminal virtual o lineas VTY produce menos gastos que enviarlo a la consola. Es importante tener en cuenta que él envió de logs o salidas de los comandos **"debug"** hacia la consola genera mayor demanda de CPU es recomendable deshabilitar esta función con el comando **"no logging console"** a nivel de configuración de terminal.

```
!-------------------------------------------------------------------------------
! Para habilitar él envió de las salidas de los comandos "debug" a la consola
!-------------------------------------------------------------------------------

# configure terminal
    no logging console
```

Cuando estas conectado en un equipo activo de red vía una sesión de Telnet o SSH esto se le conoce como VTY o Terminal emulada, para visualizar las salidas de los comandos **"debug"** usa el comando **"terminal monitor"** que se aplica en modo privilegiado (Enable) y además necesitas aplicar el comando **"logging monitor"** en modo de configuración de terminal.
Si lo que quieres es deshabilitar él envió de información a la terminal o VTY, aplica el comando **"no logging monitor"** en modo de configuración de terminal.

```
!-------------------------------------------------------------------------------
! Para habilitar él envió de las salidas de los comandos "debug" a las VTYs
!-------------------------------------------------------------------------------

# terminal monitor
# configure terminal
# logging monitor
```

Para visualizar a hacia donde se están enviando los logs; si es a la consola, a las VTYs, buffers o incluso a syslog server esto lo puedes visualizar con el comando **"show log"**.

Con el comando **"show log"** podemos verificar hacia donde se están enviando los logs de salida de los comandos **"debug"**, para el efecto de este documento solo pondremos atención en la Console, VTYs (Monitoring Logging), Buffers y Syslogs Server (Trap logging).

```
#show log
Syslog logging:enabled (0 messages dropped, 0 messages rate-limited, 0 flushes, 0 overruns, xml disabled, filtering disabled)

No Active Message Discriminator.

No Inactive Message Discriminator.

    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level debugging, 679 messages logged, xml disabled,
                     filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    File logging: disabled
    Persistent logging: disabled
    Trap logging: level informational, 638 message lines logged

Log Buffer (4096 bytes):
```

Otro punto importante es sincronizar las salidas de información generada por el comando **"debug"** hacia la consola así como en las terminales virtuales o VTYs, esto se hace con el comando **"logging synchronous"**, así como el tiempo de cierre de la sesión. Con el comando **"exec-timeout <Min> <Seg>"**, estos comandos se configuran a nivel de **"line con 0"** o **"line vty 0 5"**, respectivamente.

El comando **"exec-timeout <Min> <Seg>"** que se ejecuta a nivel de **"line con 0"** o **"line vty 0 5"**, si le indicamos ambos parámetros con los valores de **"0"** esto quiere decir que la sesión en la terminal virtual (VTY) o consola no se cierra.

```
line  con  0
    exec-timeout  0  0
    logging synchronous
line aux 0
line  vty  0 5
    exec-timeout  0 0
    logging synchronous
```

Para la manipulación del formato de cómo se envían las salidas de los comandos **"debugs"** a la consola, buffers o terminales virtuales (VTY) se utilizan los comandos **"service timestamps <debug|logs> <datetime|msec|localtime|how-timezone|year>"**, asi podemos definir el formato de la fecha, la hora en milisegundos, zona horaria o timezone que tendrá el log de salida; esto es importante ya que al definir el formato de salida de los log de los comandos **"debugs"** en milisegundos ayuda a encontrar llamadas tomando como referencia el tiempo.

```
!-------------------------------------------------------------------------------
!  Para dar formato a la salida del debug o log en el IOS Device
!-------------------------------------------------------------------------------

# service timestamps debug datetime local msec
# service timestamps log datetime local msec
```

El comando **"service sequence-numbers"** escribe el número de secuencia del debug en la línea. Esta característica es muy útil cuando se envían la salida del debug a un syslog server, pues permite identificar si un mensaje del debug no fue enviado al the syslog server. El número de secuencia será el primer elemento que se muestre en el debug antes del timestamp y el mensaje actual.

Es importante verificar que el servicio de NTP y los comandos **"Clock"** estén configurados adecuadamente en el IOS Device.

```
# show  running-config  |  include  ntp|clock
  clock timezone MX -6 0
  clock summer-time MX recurring
  ntp source GigabitEthernet0/0.3
  ntp master 2
  ntp update-calendar                    !  Este es solo en Gateways
```

Para ver que debugs están corriendo o ejecutándose actualmente en el equipo activo o IOS Device, es con el comando **"show debug"** y también es importante revisar el performance con el comando **"show process cpu history"**. En caso de querer deshabilitar todo tipo de comando **"debug"** activo en el IOS Device se hace con el comando **"undebug all"** o **"un all"** este último es para mayor velocidad y se aplica a nivel de modo Enable.

```
!-------------------------------------------------------------------------------
!  Para visualizar los debugs activos en el IOS Device
!-------------------------------------------------------------------------------

# show  debugging

CCSIP SPI: SIP Call Message tracing is enabled  (filter is OFF)
```

Nota: Los comandos **"debug"** son acumulativos, por lo que hay que tener cuidado cuando los aplicas ya que pueden estar corriendo más de uno a la vez, para deshabilitar un solo **"debug"** ejecución, antepone la palabra **"no"** al comando **"debug"** que deseas deshabilitar de la lista de debugs en ejecución.

Para controlar él envió de los Logs generados por las salidas de los comandos **"debug"** a una entidad final ya sea los buffers del mismo IOS Device o un servidor de Logs, hay que aplicar el comando **"default logging rate-limit"**. Por default los IOS Devices no controlan la velocidad con la que se envían los Logs generados por los comandos **"debug"**. Si un ingeniero del TAC sospecha que el router no está enviando los mensajes de debug al syslog server o al buffer de registro de un router, entonces puede solicitar que se incremente el rate o deshabilitarlo.

Por default el router hace encolamiento de paquetes. Hay una cantidad limitada de memoria que el enrutador almacenará en la cola mientras espera que se escriba en el búfer de registro o envíe el mensaje a un servidor de syslog. El comando **"default logging queue-limit"** por lo general, se recomienda dejarlo encendido para garantizar la estabilidad del enrutador. Si un ingeniero de TAC sospecha que el enrutador está eliminando depuraciones antes de llegar al búfer de registro o al servidor de registro del enrutador, puede solicitar que se aumente a un valor mayor o que se deshabilite.

Nota: Cambiar el **"default logging rate-limit"** y el **"default logging queue-limit"** en entornos con alto volumen de trafico puede causar inestabilidad en la CPU, ya que asegurará que cada mensaje de **"debug"** llegue al búfer de registro o al servidor syslog.

Recolección de Logs en Buffers

Una buena práctica es mandar la salida de los comandos **"debug"** a los buffers del equipo activo (IOS Device), esto se hace con el comando **"logging buffer <tamaño_buffer_bytes> <type_log>"**. El ultimo parámetro de este comando **"Type Log"** es para definir el nivel de log que se quiera enviar a los buffers.

```
!-------------------------------------------------------------------------
!  Envió de logs a Buffers y definiendo una tamaño de 1 M
!-------------------------------------------------------------------------

(config)# logging   buffered   1000000   ?

  <0-7>          Logging severity level
  alerts         Immediate action needed           (severity=1)
  critical       Critical conditions               (severity=2)
  debugging      Debugging messages                (severity=7)
  emergencies    System is unusable                (severity=0)
  errors         Error conditions                  (severity=3)
  informational  Informational messages            (severity=6)
  notifications  Normal but significant conditions (severity=5)
  warnings       Warning conditions                (severity=4)
  <cr>
```

Nota: si tienes dudas en cuanta memoria definir para el espacio en buffer, se hace con el comando **"show memory summary"**, en la columna con el nombre de "Largest (b)" nos indica en cuanta memoria hay disponible ara el buffer. Para este ejemplo tenemos como memoria disponible 61 Megas (61546852).

```
C3560-SW-Lab#show  memory  summary
                Head    Total(b)     Used(b)     Free(b)   Lowest(b)  Largest(b)
Processor   26A92F0    85290256    19811248    65479008    63350604    61546852
      I/O    7800000     8380416     3620584     4759832     4712132     4753824
Driver  te   1900000     1048576          44     1048532     1048532     1048532
```

Nota: Cuando envías la salida de los comandos **"debugs"** a los Buffers, cuando estos alcanzan el limite de memoria definido previamente, van borrando la información mas vieja, para asi permitir a las nuevas líneas integrarse.

Antes de aplicar él envió de los Logs generados por los comandos **"debug"** hacia los buffers del equipo activo (IOS Device) es muy recomendable limpiarlos con el comando **"clear logging"**.

```
!-------------------------------------------------------------------------
!  Limpiamos los Buffers con el comando "clear logging"
!-------------------------------------------------------------------------

# clear logging
```

El script para enviar la salida de los comandos **"debug"** hacia los buffers quedaría de la siguiente manera.

```
!-------------------------------------------------------------------------
!  Script de envió de logs hacia los buffers
!-------------------------------------------------------------------------

# configure  terminal
    service timestamps debug datetime local msec
    service timestamps log datetime local msec
    service sequence
    no logging console
    no logging monitor
    no logging rate-limit
    no logging queue-limit
    logging buffer 10000000 debug
    end
```

Ya que indicamos al equipo activo o IOS device que envié toda la salida del comando **"debug"** al buffer y no a la consola o terminal (VTY), para visualizar la captura que está en el buffer de equipo, se hace aplicando el comando **"show logging"**. Una nota importante aquí es que para visualizar todos los logs capturados en el buffer evitando una paginación de 20 líneas hay que aplicar los comandos **"terminal length 0"** y **"terminal default length"** antes y después de mostrar los logs capturados en el buffer.

```
!-------------------------------------------------------------------------
!  Para visualizar los logs capturados en el buffers
!-------------------------------------------------------------------------

# terminal length 0
# show  logging
# terminal default length
```

Para volver a mandar los logs a las terminales y consola es con los comandos **"logging monitor"** y **"logging console"** a nivel de privileged EXEC mode respectivamente.

```
!-------------------------------------------------------------------------------
!  Para restaurar el envió de los logs a la consola y terminales VTY
!-------------------------------------------------------------------------------

# configure terminal
    logging monitor
    logging console
    logging on
    end
```

Recolección de Logs en un Syslog server

Los dispositivos Cisco carecen de grandes cantidades de almacenamiento para guardar logs. Para solucionar esta limitante, Cisco ofrece el uso de un protocolo SYSLOG de estilo UNIX para enviar mensajes a un dispositivo externo para su almacenamiento. El tamaño de almacenamiento no depende de los recursos del enrutador y está limitado solo por el espacio disponible en disco en el servidor syslog externo. Esta opción no está habilitada por defecto.

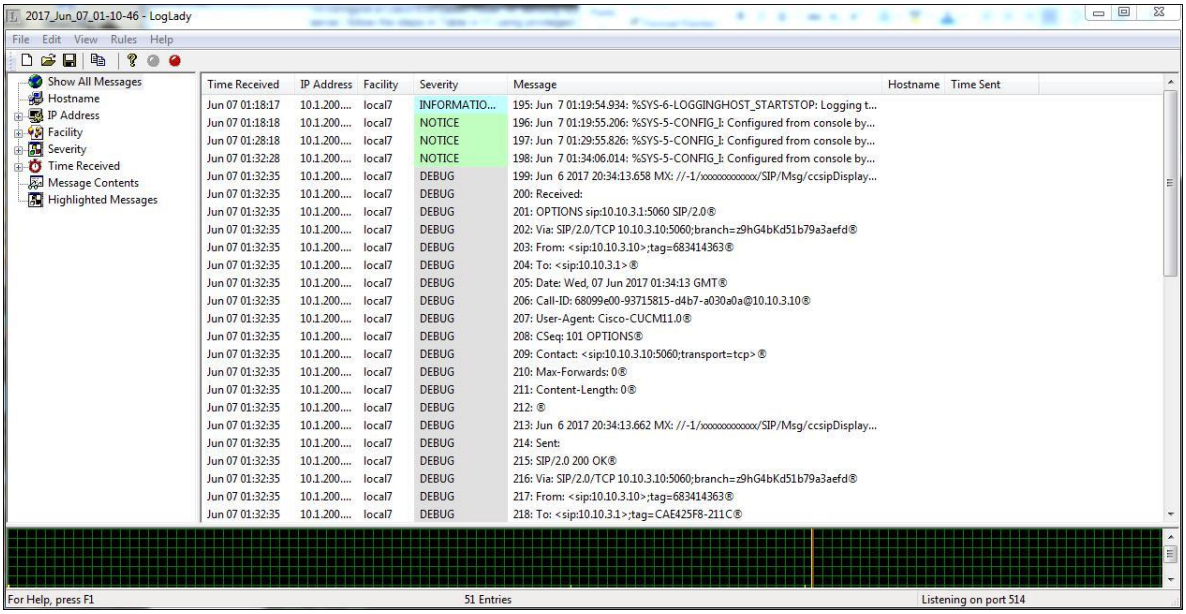
Para configurar el envió de las salidas de los comandos **"debug"** hacia un servidor de Logs se hace con el comando **"logging host <IP_SYSLOG_SRV>"** con el cual indicas la IP address del servidor de logs, así también es necesario indicar el **"Type Log"** con el comando **"logging trap <type_log >"**.

A continuación tenemos un script el cual se puede usar para mandar la salida de un comando **"debug"** hacia un servidor de Logs.

```
!-------------------------------------------------------------------
! Script de envió de logs hacia un servidor de logs
!-------------------------------------------------------------------

# configuration terminal
    service timestamps debug datetime local msec
    service timestamps log datetime local msec
    service sequence
    no logging console
    no logging monitor
    no logging rate-limit
    no logging queue-limit
    logging host <IP_SYSLOG_SRV>
    logging trap <Type_Log>
    end
```

Existen diferentes tipos de aplicaciones que vuelven a tu PC un servidor de logs como por ejemplo el uso del uno llamado Lady Log es una aplicación para windows que habilita este tipo de servicio en una maquina Windows.



Filtrado de depuración de llamadas de voz

A veces es importante filtrar los logs que se generar por parte de los comandos **"debug"**, para habilitar este tipo de filtro es por medio de comando **"call filter match-list <tag> voice"** el cual se ejecuta a nivel de configuración de terminal y después este comando te introduce en un sub-modo de configuración del mismo filtro, donde puedes definir el ANI o DNIS a filtrar, puerto, IP ya sea para llamadas de entrada o de salida.

```
!-------------------------------------------------------------------------------
! Customizacion de filtros para salida de logs en comandos debugs
!-------------------------------------------------------------------------------

# configure terminal
```

```
call filter match-list <tag> voice
    incoming calling-number <pattern>
    incoming called-number <pattern>
    incoming dialpeer <tag>
    incoming media local ipv4 <ip address>
    incoming media remote ipv4 <ip address>
    incoming port
    incoming signaling local ipv4 <ip address>
    incoming signaling remote ipv4 <ip address>
    incoming secondary-called-number <pattern>
    incoming uri <pattern>
    outcoming calling-number <pattern>
    outcoming called-number <pattern>
    outcoming dialpeer <tag>
    outcoming media local ipv4 <ip address>
    outcoming media remote ipv4 <ip address>
    outcoming port
    outcoming signaling local ipv4 <ip address>
    outcoming signaling remote ipv4 <ip address>
    outcoming secondary-called-number
    end
```

Para ver si se tiene habilitado un filtro en los comandos **"debug"** ejecutados en el IOS Devices, el siguiente comando **"show debugging"**.

```
!-------------------------------------------------------------------------
!  Para ver si tenemos un filtro activado en un debug
!-------------------------------------------------------------------------

#show   debugging

CCSIP SPI: SIP Call Message tracing is enabled  (filter is ON)
```

Para visualizar que reglas están configuradas dentro del filtro es con el comando **"show call filter match-list <tag>"**.

```
!-------------------------------------------------------------------------
!  Para ver que filtros existen en el IOS Device
!-------------------------------------------------------------------------

# show  call  filter  match-list  1
     outgoing calling-number 1004
debug condition match-list is set to EXACT_MATCH
```

Nota: Para deshabilitar estos filtros es ante poniendo la palabra **"no"** al comando **"show call filter match-list <tag>"** a nivel de configuración de terminal.

A continuación mostramos un ejemplo de cómo filtramos todas las llamadas que se hacen desde la LAN hacia la PSTN teniendo como ANI la extensión con el numero 1004.

```
# configure terminal
    call  filter  match-list  1  voice
       outgoing calling-number 1004
       end
# debug condition match-list 1 exact-match
# debug voip dialpeer inout
# terminal monitor
```

Lista de comandos Logging

- logging host { *ip* | *host-name*} : Para registrar mensajes en un host del servidor syslog.
- Logging trap <type_log> : Para limitar el número de mensajes enviados a los servidores de syslog, use el comando de configuración del enrutador de captura de registro. La sintaxis completa de este comando es "logging trap level".
- logging buffered <tamaño_buffer_bytes> <type_log> : Le dice al enrutador que envíe debugs a su registro de buffer local en la memoria del sistema. El tamaño del búfer se establece en bytes, y es de 10 MB aquí. El tamaño del búfer que pueda necesitar depende del volumen de la llamada, la duración del tiempo que el búfer necesita almacenar, la memoria del sistema aún está disponible (aproveche 'mostrar historial de estadísticas de memoria' y 'mostrar resumen de memoria' para esto).
- no logging console : Por defecto, el enrutador envía debugs a la consola. En iOS, la consola tiene la prioridad más alta de cualquier proceso. También se ejecuta a velocidades muy lentas (comúnmente 9600bps). Debido a esto, si las fallas se envían al enrutador más rápido que la velocidad de la consola, puede privar a la entrada de la consola y / o hacer que la CPU pase al 100%. Para aliviar este comportamiento, al ejecutar cualquier depuración en IOS, es imprescindible el envío de depuraciones a la consola se desactiva al ingresar este comando.
- no logging monitor : este comando evita que el enrutador envíe depuraciones en tiempo real a la sesión VTY (telnet / SSH) del enrutador. Como estaremos eliminando errores de forma reactiva, no queremos que nada se desplace en tiempo real. Además, el monitor de la terminal tiene el hábito de soltar mensajes si llegan en ráfagas, como hacen la mayoría de las depuraciones de voz.
- default logging rate-limit : e manera predeterminada, el enrutador sí envía mensajes de límite de velocidad. Por lo general, se recomienda dejarlo encendido para garantizar la estabilidad del enrutador. Si un ingeniero de TAC sospecha que el enrutador está eliminando depuraciones antes de llegar al búfer de registro del enrutador, puede solicitar que esto se incremente a un valor mayor o que se desactive. Tenga en cuenta que cambiar esto en entornos con alto volumen de tráfico puede causar inestabilidad de la CPU, ya que asegurará que cada mensaje de depuración llegue al búfer de registro.
- default logging queue-limit : e manera predeterminada, el enrutador sí envía mensajes de límite de velocidad. Por lo general, se recomienda dejarlo encendido para garantizar la estabilidad del enrutador. Si un ingeniero de TAC sospecha que el enrutador está eliminando depuraciones antes de llegar al búfer de registro del enrutador, puede solicitar que esto se incremente a un valor mayor o que se desactive. Tenga en cuenta que cambiar esto en entornos con alto volumen de tráfico puede causar inestabilidad de la CPU, ya que asegurará que cada mensaje de depuración llegue al búfer de registro.
- service sequence-numbers: este comando escribe el número de secuencia de la depuración en la línea. Esto es útil (esencialmente requerido) cuando se envía a un servidor de syslog, para identificar si algún mensaje de depuración al servidor de syslog ha sido eliminado en la red. El número de secuencia será el primer elemento en la depuración, antes de la marca de tiempo y el mensaje real. Tenga en cuenta que esto es diferente del número de marca de tiempo / secuencia que puede escribir en los archivos de registro de syslog, si corresponde.