

El resolver problemas de las fugas de memoria de MallocLite

Contenido

[Introducción](#)

[Antecedentes](#)

[Troubleshooting](#)

[Identifique la aplicación responsable del escape](#)

[Decodifique el distribuidor PC](#)

[Investigue las estadísticas de la memoria de MallocLite](#)

[Inhabilite MallocLite](#)

Introducción

Este documento describe cómo resolver problemas las fugas de memoria de MallocLite en las Plataformas del ^{@software del} Cisco IOS.

También especifica la información que usted debe recopilar antes de que usted abra un caso del Centro de Asistencia Técnica de Cisco (TAC) o recargue el dispositivo. Recoja las salidas mencionadas en este documento, y asócielas al caso TAC para ayudar a apresurar la solución de problemas.

Antecedentes

MallocLite es utilizado por el administrador de la memoria para afectar un aparato los pedazos pequeños, de tamaño fijo de memoria, conocidos como pedazos, para las asignaciones inferior o igual los bytes 128. Las pequeñas asignaciones de memoria no tienen los gastos indirectos de una encabezado de bloque para cada asignación. Esta característica se soporta para los pools de memoria del procesador solamente.

Cada encabezado de bloque de memoria toma cerca de 48 bytes de memoria, y el bloque más pequeño toma cerca de 24 bytes. Con un acercamiento tradicional en Cisco IOS Software para cada asignación, usted consumiría por lo menos 72 (48 + 24) bytes de memoria, incluso si usted necesita afectar un aparato solamente 8 bytes de los datos reales.

Con MallocLite, estos gastos indirectos se pueden reducir por el uso de los pedazos. Todavía hay un ciertos gastos indirectos, porque los pedazos tienen que ser manejados. Sin embargo, puesto que los pedazos son de tamaño fijo, se manejan de una manera diferente que los bloques, y los gastos indirectos son menos.

Es la responsabilidad de las aplicaciones que utilizan la memoria de MallocLite para liberarla correctamente. MallocLite enmascara al usuario de la memoria.

Troubleshooting

Nota: [El analizador del CLI de Cisco \(clientes registrados solamente\)](#) apoya los ciertos comandos show. Utilice el analizador del CLI de Cisco para ver una análisis de la salida del comando show.

Identifique la aplicación responsable del escape

Es generalmente difícil identificar un bug existente si usted busca solamente por la palabra clave del *malloclite*.

Este ejemplo muestra que el proceso del **MallocLite** está celebrando una cantidad de memoria anormal:

```
#show processes memory sorted
```

```
Processor Pool Total: 1614282720 Used: 1544726580 Free: 69556140
I/O Pool Total: 313524224 Used: 115564032 Free: 197960192
```

```
PID TTY Allocated Freed Holding Getbufs Retbufs Process
0 0 0 0 1476043512 0 0 *MallocLite*
```

Usted necesita identificar la aplicación exacta que es responsable del escape. Tres métodos posibles de la identificación son:

- Decodifique el distribuidor PC.
- Investigue las estadísticas de la memoria de MallocLite.
- Inhabilite MallocLite.

Decodifique el distribuidor PC

Incluso con MallocLite se giró, usted puede ver generalmente lo que funcionar preguntó la memoria. La salida del **comando show memory allocating-process totals** pudo mostrar diversos valores PC aunque el nombre señalado es MallocLite:

```
#show processes memory sorted
```

```
Processor Pool Total: 1614282720 Used: 1544726580 Free: 69556140
I/O Pool Total: 313524224 Used: 115564032 Free: 197960192
```

```
PID TTY Allocated Freed Holding Getbufs Retbufs Process
0 0 0 0 1476043512 0 0 *MallocLite*
```

Un ingeniero de Cisco TAC puede decodificar los valores PC desde arriba de la lista (con el total más alto). Esto ayuda a identificar la aplicación que tiene la fuga de memoria.

Investigue las estadísticas de la memoria de MallocLite

Entre las mejoras agregadas en el Cisco IOS Software Release 15.1T era un nuevo CLI que visualiza el resumen de memoria de MallocLite afectado un aparato por cada PC. El comando de los **lite-pedazos de la memoria de la demostración** puede ayudarle a identificar las aplicaciones que están utilizando una gran cantidad de bloques de MallocLite.

```
#show processes memory sorted
```

```
Processor Pool Total: 1614282720 Used: 1544726580 Free: 69556140
I/O Pool Total: 313524224 Used: 115564032 Free: 197960192
```

```
PID TTY Allocated      Freed   Holding   Getbufs   Retbufs Process
0  0          0          0  1476043512      0          0 *MallocLite*
```

Refiera a la [referencia de comandos](#) para los detalles del comando de los lite-pedazos de la memoria de la demostración.

```
#show processes memory sorted
```

```
Processor Pool Total: 1614282720 Used: 1544726580 Free: 69556140
I/O Pool Total: 313524224 Used: 115564032 Free: 197960192
```

```
PID TTY Allocated      Freed   Holding   Getbufs   Retbufs Process
0  0          0          0  1476043512      0          0 *MallocLite*
```

Ejemplos de salida de este comando incluye:

```
#show processes memory sorted
```

```
Processor Pool Total: 1614282720 Used: 1544726580 Free: 69556140
I/O Pool Total: 313524224 Used: 115564032 Free: 197960192
```

```
PID TTY Allocated      Freed   Holding   Getbufs   Retbufs Process
0  0          0          0  1476043512      0          0 *MallocLite*
```

Una vez más el ingeniero de TAC puede decodificar los valores PC con el total más alto e identificar la aplicación que se está escapando la memoria.

Neutralización MallocLite

La característica de MallocLite se habilita por abandono. Para investigar el escape de MallocLite, usted puede inhabilitar MallocLite:

```
#show processes memory sorted
```

```
Processor Pool Total: 1614282720 Used: 1544726580 Free: 69556140
I/O Pool Total: 313524224 Used: 115564032 Free: 197960192
```

```
PID TTY Allocated      Freed   Holding   Getbufs   Retbufs Process
0  0          0          0  1476043512      0          0 *MallocLite*
```

La memoria escapada todavía estará bajo MallocLite hasta la recarga siguiente; sin embargo, usted puede comenzar a monitorear otros escapes con la **memoria de los procesos de la demostración clasificada** y los **comandos show memory allocating-process totals**. Los escapes ahora aparecerán bajo proceso real.

Si el dispositivo se ejecuta muy bajo en la memoria, usted debe salvar la configuración y recargar el dispositivo para liberar la memoria:

```
#show processes memory sorted
```

```
Processor Pool Total: 1614282720 Used: 1544726580 Free: 69556140
I/O Pool Total: 313524224 Used: 115564032 Free: 197960192
```

```
PID TTY Allocated      Freed   Holding   Getbufs   Retbufs Process
0  0          0          0  1476043512      0          0 *MallocLite*
```

La memoria pudo agotar otra vez en un cierto plazo, así que utilice la **memoria de los procesos de la demostración clasificada** y los **comandos show memory allocating-process totals** para monitorear el uso de la memoria de esa punta adelante.

Nota: Si usted inhabilita con eficacia MallocLite con el **ningún** comando de **lite de la memoria**

y recarga el dispositivo, la salida del comando de los lite-pedazos de la memoria de la **demonstración** estará vacía.

Refiera a la [referencia de comandos](#) para los detalles del comando de **lite de la memoria**.