

Fehlerbehebung bei MallocLite-Speicherlecks

Inhalt

[Einführung](#)

[Hintergrundinformationen](#)

[Fehlerbehebung](#)

[Identifizieren der für Lecks verantwortlichen Anwendung](#)

[Decode Allocator-PC](#)

[Analyse von MallocLite-Speicherstatistiken](#)

[MallocLite deaktivieren](#)

Einführung

In diesem Dokument wird beschrieben, wie MallocLite-Speicherlecks auf Cisco IOS®-Softwareplattformen behoben werden.

Außerdem wird festgelegt, welche Informationen Sie sammeln sollten, bevor Sie ein Cisco Technical Assistance Center-Ticket (TAC) öffnen oder das Gerät neu laden. Erfassen Sie die in diesem Dokument erwähnten Ergebnisse, und fügen Sie sie dem TAC-Fall bei, um die Problemlösung zu beschleunigen.

Hintergrundinformationen

MallocLite wird vom Speichermanager verwendet, um kleine, feste Speichereinheiten, die als Chunks bezeichnet werden, für Zuweisungen von weniger als oder gleich 128 Byte zuzuweisen. Kleine Speicherzuweisungen haben nicht den Overhead eines Block-Headers für jede Zuweisung. Diese Funktion wird nur für Speicherpools von Prozessoren unterstützt.

Jeder Header des Speicherblocks benötigt etwa 48 Byte Arbeitsspeicher, und der kleinste Block benötigt etwa 24 Byte. Bei einem herkömmlichen Ansatz in der Cisco IOS-Software für jede Zuweisung würden Sie mindestens 72 (48 + 24) Byte Arbeitsspeicher benötigen, selbst wenn Sie nur 8 Byte tatsächliche Daten zuweisen müssen.

Mit MallocLite lässt sich dieser Overhead durch den Einsatz von Chunks reduzieren. Es entstehen noch einige Gemeinkosten, da die Chunks verwaltet werden müssen. Da die Chunks jedoch eine feste Größe haben, werden sie auf eine andere Weise als Blöcke verwaltet, und der Overhead ist geringer.

Es liegt in der Verantwortung der Anwendungen, die den MallocLite-Speicher verwenden, ihn richtig freizugeben. MallocLite maskiert den Benutzer des Speichers.

Fehlerbehebung

Hinweis: Der [Cisco CLI Analyzer](#) (nur registrierte Kunden) unterstützt bestimmte **show**-Befehle. Verwenden Sie den Cisco CLI Analyzer, um eine Analyse der **Ausgabe des Befehls**

show anzuzeigen.

Identifizieren der für Lecks verantwortlichen Anwendung

In der Regel ist es schwierig, einen vorhandenen Fehler zu identifizieren, wenn Sie nur das *mallocite* Schlüsselwort suchen.

Dieses Beispiel zeigt, dass der **MallocLite**-Prozess eine ungewöhnliche Speichergröße aufweist:

```
#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*
```

Sie müssen die exakte Anwendung identifizieren, die für Lecks verantwortlich ist. Drei mögliche Identifizierungsmethoden sind:

- Decodier-Zuweiser-PC.
- Analysieren Sie MallocLite-Speicherstatistiken.
- Deaktivieren Sie MallocLite.

Decode Allocator-PC

Selbst wenn MallocLite eingeschaltet ist, können Sie in der Regel sehen, welche Funktion für den Speicher benötigt wurde. Die Ausgabe des Befehls **show memory allocations-process totals** kann verschiedene PC-Werte anzeigen, obwohl der gemeldete Name MallocLite ist:

```
#show memory allocating-process totals  
<snip>
```

```
Allocator PC Summary for: Processor  
Displayed first 2048 Allocator PCs only  
PC Total Count Name  
0x620BE3C4 42807572 594 MallocLite  
0x620ADDD4 13597308 193 MallocLite  
0x60738BB0 8909824 122 MallocLite  
0x620AE0E0 2060716 31 MallocLite  
0x620AE10C 1982780 30 MallocLite
```

Ein Cisco TAC-Techniker kann die PC-Werte ganz oben in der Liste decodieren (mit der höchsten Gesamtzahl). Dies hilft bei der Identifizierung der Anwendung, die Speicherlecks aufweist.

Analyse von MallocLite-Speicherstatistiken

Zu den Verbesserungen in der Cisco IOS-Softwareversion 15.1T gehört eine neue CLI, die die Zusammenfassung des von jedem PC zugewiesenen MallocLite-Speichers anzeigt. Der Befehl **show memory lite-chunks** kann Ihnen dabei helfen, Anwendungen zu identifizieren, die eine große Anzahl von MallocLite-Blöcken verwenden.

```
show memory lite-chunks { statistics | totals } { summary { pool | { all | pool } } }
```

Weitere Informationen zum Befehl **show memory lite-chunks** finden Sie in der [Befehlsreferenz](#).

CLI : show memory lite-chunks totals
DESC : Summary of all pools, based on alloc pc.
This cli can be used to find the alloc_pc which is using large amount memory allocated from all mlite pools

CLI : show memory lite-chunks statistics
DESC : Displays number of allocated & free mlite chunks

CLI : show memory lite-chunks summary pool *pool*
DESC : Show summary of particular mlite pool
This cli can be used to find the alloc_pc which is using large amount of memory in individual mlite pool

CLI : show memory lite-chunks summary pool all
DESC : Show individual summary of all mlite pools

CLI : show memory lite-chunks pool *pool*
DESC : Show All chunk elements in the specified pool

CLI : show memory lite-chunks pool all
DESC : show all chunk elements in all mlite pools

Beispiele für die Ausgabe dieses Befehls:

```
#show memory lite-chunks ?
pool          Malloc lite pool
statistics    Malloc lite statistics
summary       Malloc Lite summary
totals        Malloc Lite Allocating totals
```

```
#show memory lite-chunks statistics
```

Pool	Inuse	Free
8-Bytes	140	1904
20-Bytes	173	1313
44-Bytes	171	791
68-Bytes	24	687
96-Bytes	26	519
128-Bytes	20	410

```
#show memory lite-chunks totals
```

PC	Total	Count
26067AE0	2112	33
2269E68C	1932	29
2269FACC	1664	29
2269F964	1664	26
2269FA9C	1580	29
26067FB4	1360	34
23CD2A0C	1036	7

```
#show memory lite-chunks pool ?
```

```
128-Bytes 128 bytes pool
20-Bytes  20 bytes pool
44-Bytes  44 bytes pool
68-Bytes  68 bytes pool
8-Bytes   8 bytes pool
96-Bytes  96 bytes pool
```

```
all          all pools
```

```
#show memory lite-chunks summary pool 8
```

```
      8 bytes pool
      PC          Total      Count
2269FB10          812         29
23612084           700         25
2269F9F8           700         25
2269F9EC           700         25
```

Auch hier kann der TAC-Techniker PC-Werte mit der höchsten Gesamtanzahl decodieren und die Anwendung identifizieren, die den Speicher ausläuft.

MallocLite deaktivieren

Die MallocLite-Funktion ist standardmäßig aktiviert. Um das MallocLite-Leck zu untersuchen, können Sie MallocLite deaktivieren:

```
(config)#no memory lite
```

Der gesickerte Speicher verbleibt bis zum nächsten Neuladen weiterhin unter MallocLite. Sie können jedoch weitere Datenlecks überwachen, indem Sie den **Arbeitsspeicher der Anzeigeprozesse sortiert** und die Befehle zur **Speicherzuordnung** anzeigen. Die Lecks erscheinen nun unter dem eigentlichen Prozess.

Wenn das Gerät einen sehr niedrigen Arbeitsspeicher belegt, müssen Sie die Konfiguration speichern und das Gerät neu laden, um den Speicher freizugeben:

```
#wr
#reload
```

Der Speicher kann mit der Zeit wieder voll sein. Verwenden Sie daher den **sortierten Arbeitsspeicher show** und zeigen Sie den **Arbeitsspeicherzuweisungsprozess als Gesamtbefehle** an, um die Speichernutzung von diesem Zeitpunkt an zu überwachen.

Hinweis: Wenn Sie MallocLite mit dem Befehl **no memory lite** effektiv deaktivieren und das Gerät neu laden, ist die Ausgabe des Befehls **show memory lite-chunks** leer.

Weitere Informationen zum Befehl **Memory Lite** finden Sie in der [Befehlsreferenz](#).