

Fehlerbehebung: Pufferlecks

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Konventionen](#)

[Leaks im Puffer für hochgezackte Schnittstellen](#)

[Systempuffer-Lecks](#)

[Tipps zur Fehlerbehebung](#)

[Zugehörige Informationen](#)

Einführung

Pufferleaks sind Cisco IOS[®] Softwarefehler. Es gibt zwei Arten von Pufferlecks:

- Puffer für Hochzeitschnittstellen sind undicht.
- Systempufferlecks.

Um Probleme mit Pufferlecks zu beheben, müssen Sie die Art des Pufferlecks identifizieren, auf den Sie stoßen. Die Befehle **show interfaces** und **show buffers** sind in dieser Situation sehr hilfreich.

Wenn Sie die Ausgabe von **Show-Schnittstellen** und **Show-Puffer**-Befehlen von Ihrem Cisco Gerät haben, können Sie [Cisco CLI Analyzer](#) zur Anzeige potenzieller Probleme und Fixes verwenden. Um den [Cisco CLI Analyzer](#) verwenden zu können, müssen Sie [registrierter Kunde sein, sich anmelden und JavaScript aktiviert haben](#).

Voraussetzungen

Anforderungen

Es gibt keine spezifischen Anforderungen für dieses Dokument.

Verwendete Komponenten

Dieses Dokument ist nicht auf bestimmte Software- und Hardware-Versionen beschränkt.

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netz Live ist, überprüfen Sie, ob Sie die mögliche Auswirkung jedes möglichen Befehls verstehen.

Konventionen

Weitere Informationen zu Dokumentkonventionen finden Sie unter [Cisco Technical Tips Conventions](#) (Technische Tipps von Cisco zu Konventionen).

Leaks im Puffer für hochgezackte Schnittstellen

Hochzeitspufferlecks verursachen, dass die Eingabewarteschlange einer Schnittstelle bis zu dem Punkt gefüllt ist, an dem sie keine Pakete mehr akzeptieren kann. Unter bestimmten Verkehrsbedingungen wird die Eingangswarteschlange einer Schnittstelle verkürzt, d. h. die Anzahl der Eingangswarteschlangen ist größer als die Warteschlangentiefe.

Hier ein Beispiel für die Ausgabe des Befehls **show interfaces**, der zeigt, dass die Schnittstelle hochgeladen ist:

```
Ethernet0/0 is up, line protocol is up
Output queue 0/40, 0 drops; input queue 76/75, 1250 drops
```

Das Symptom eines solchen Pufferlecks ist eine vollständige Eingabewarteschlange (76/75). Hier stellen die Werte 76 und 75 die Anzahl der Pakete in der Eingangswarteschlange bzw. die maximale Größe der Eingangswarteschlange dar. In diesem Fall ist die Anzahl der Pakete in der Eingabewarteschlange größer als die Warteschlangentiefe. Dies wird als "Hochzeitsschnittstelle" bezeichnet. Wenn eine Schnittstelle verbunden ist, leitet der Router keinen Datenverkehr mehr weiter, der von der betroffenen Schnittstelle stammt.

Laden Sie den Router neu, um die Eingangswarteschlange freizugeben, und stellen Sie den Datenverkehr wieder her, bis die Warteschlange wieder voll ist. Dies kann je nach Schweregrad des Lecks zwischen ein paar Sekunden und ein paar Wochen dauern.

Vorsicht: Bevor Sie den Router neu laden, stellen Sie sicher, dass Sie alle erforderlichen Informationen zur Identifizierung des Schuldigen sammeln.

Verwenden Sie diese Befehle, um die Quelle des Pufferlecks zu identifizieren:

- **show buffers pool [Poolname] [Paket/Header]**
- **show buffers old** (Verwenden Sie diesen Befehl nur, wenn **debug sanity** aktiviert ist. **Hinweis:** Der Befehl **debug sanity** ist in den meisten Cisco IOS-Softwareversionen verborgen. Wenn **Debugging-Integrität** aktiviert ist, wird jeder im System verwendete Puffer bei der Zuweisung und bei der Freigabe auf die Integrität überprüft. **Hinweis:** Sie müssen den Befehl **debug sanity** im privilegierten EXEC-Modus (Aktivierungsmodus) ausführen. Obwohl dieser Befehl eine gewisse CPU-Kapazität verwendet, hat er keine signifikanten Auswirkungen auf die Funktionalität des Routers. Wie andere Debugbefehle wird die **Fehlerprüfbarkeit** nicht in der Konfiguration gespeichert. Daher übersteht dieser Befehl keinen Neustart des Systems. **Hinweis:** Um die Überprüfung der Integrität zu deaktivieren, verwenden Sie den privilegierten EXEC-Befehl **undebug sanity**.)
- **Belegungspuffer anzeigen**

Systempuffer-Lecks

In diesem Abschnitt werden Pufferlecks des Systems behandelt.

Im Folgenden finden Sie ein Beispiel für die Ausgabe des Befehls **show buffers**, der auf ein

Pufferleck in einem der Systempufferpools hinweist:

```
Middle buffers, 600 bytes (total 20825, permanent 180):
 286 in free list (20 min, 400 max allowed)
 89122311 hits, 99597 misses, 133679 trims, 154324 created
 2247 failures (0 no memory)
```

Diese Ausgabe des Befehls **show buffers** gibt einen Pufferleck im mittleren Puffer-Pool an. Der Router verfügt über insgesamt 20825 Zwischenpuffer, und nur 286 befinden sich in der freien Liste. Dies impliziert, dass ein Prozess alle Puffer annimmt, aber nicht zurückgibt.

Weitere Symptome dieser Art von Pufferlecks sind "%SYS-2-MALLOCFAIL"-Fehlermeldungen für den Poolprozessor oder die Eingabe/Ausgabe (I/O), basierend auf der Plattform.

Verwenden Sie diese Befehle, um die Quelle des Pufferlecks zu identifizieren:

- **show buffers old** (Verwenden Sie diesen Befehl nur, wenn **debug sanity** aktiviert ist. **Hinweis:** Der Befehl **debug sanity** ist in den meisten Cisco IOS-Softwareversionen verborgen. Wenn **Debugging-Integrität** aktiviert ist, wird jeder im System verwendete Puffer bei der Zuweisung und bei der Freigabe auf die Integrität überprüft. **Hinweis:** Sie müssen den Befehl **debug sanity** im privilegierten EXEC-Modus (Aktivierungsmodus) ausführen. Obwohl dieser Befehl eine gewisse CPU-Kapazität verwendet, hat er keine signifikanten Auswirkungen auf die Funktionalität des Routers. Wie andere Debugbefehle wird die **Fehlerprüfbarkeit** nicht in der Konfiguration gespeichert. Daher übersteht dieser Befehl keinen Neustart des Systems. **Hinweis:** Um die Überprüfung der Integrität zu deaktivieren, verwenden Sie den privilegierten EXEC-Befehl **undebug sanity**.)
- **show buffers pool [Poolname] [Paket/Header]**
- **Belegungspuffer anzeigen**

Tipps zur Fehlerbehebung

Pufferleaks sind Cisco IOS Software-Bugs. Um bekannte Bug-Fehler zu beheben, aktualisieren Sie auf die neueste Version in Ihrem Release Train. Wenn Sie z. B. derzeit die Cisco IOS Software Release 11.2(14) verwenden, aktualisieren Sie auf das neueste 11.2(x)-Image. Falls dies nicht hilfreich ist oder ein Upgrade des Routers nicht möglich ist, wenden Sie sich an das Cisco TAC und stellen Sie dem Techniker die Ausgabe der entsprechenden **Show-Puffer-Befehle** sowie die Ausgabe des Befehls **show tech-support** zur Verfügung.

Hier einige Tipps, die Ihnen dabei helfen, die Pakete zu identifizieren, die das Pufferleck verursachen:

- Wenn Sie ein Pufferleck erkennen, verwenden Sie die zugehörigen Befehle **show buffers**, um ein Muster in Paketen zu finden, die so viele Puffer verwenden.
- Wenn Sie den Pakettyp identifizieren, versuchen Sie, eine Lösung zu entwickeln, die ein Leck verhindert (z. B. mithilfe einer Zugriffsliste, um diese Pakete zu filtern).

Im Folgenden finden Sie Ausgabebeispiele für zugeordnete **show**-Befehle:

```
Router#show interface ethernet 0/0
Ethernet0/0 is up, line protocol is up
Hardware is AmdP2, address is 0050.3ee8.4060 (bia 0050.3ee8.4060)
```

```

Internet address is 10.200.40.37/22
MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
Encapsulation ARPA, loopback not set, keepalive set (10 sec)
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:51, output 00:00:08, output hang never
Last clearing of "show interface" counters never
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 76/75, 1250 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  15686 packets input, 2872866 bytes, 0 no buffer
  Received 15342 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input packets with dribble condition detected
  10352 packets output, 1031158 bytes, 0 underruns
  0 output errors, 0 collisions, 3 interface resets
  0 babbles, 0 late collision, 2 deferred
  0 lost carrier, 0 no carrier
  0 output buffer failures, 0 output buffers swapped out

```

Router#**show buffers old**

Header	DataArea	Pool	Rcnt	Size	Link	Enc	Flags	Input	Output
80F09828	1A00084	Small	1	54	11	11	201	Et0/0	None
80F09A34	1A001C4	Small	1	54	11	11	201	Et0/0	None
80F09C40	1A00304	Small	1	54	11	11	201	Et0/0	None
80F09E4C	1A00444	Small	1	54	11	11	201	Et0/0	None
80F0A058	1A00584	Small	1	54	11	11	201	Et0/0	None
80F0A264	1A006C4	Small	1	54	11	11	201	Et0/0	None
80F0A470	1A00804	Small	1	54	11	11	201	Et0/0	None
80F0A67C	1A00944	Small	1	54	11	11	201	Et0/0	None
80F0A888	1A00A84	Small	1	54	11	11	201	Et0/0	None
80F0AA94	1A00BC4	Small	1	54	11	11	201	Et0/0	None
80F0ACA0	1A00D04	Small	1	54	11	11	201	Et0/0	None
80F0AEAC	1A00E44	Small	1	54	11	11	201	Et0/0	None
80F0B0B8	1A00F84	Small	1	54	11	11	201	Et0/0	None
80F0B2C4	1A010C4	Small	1	54	11	11	201	Et0/0	None
80F0B4D0	1A01204	Small	1	54	11	11	201	Et0/0	None
80F0B6DC	1A01344	Small	1	54	11	11	201	Et0/0	None
80F0B8E8	1A01484	Small	1	54	11	11	201	Et0/0	None
80F0BAF4	1A015C4	Small	1	54	11	11	201	Et0/0	None
80F0BD00	1A01704	Small	1	54	11	11	201	Et0/0	None
80F0BF0C	1A01844	Small	1	54	11	11	201	Et0/0	None
80F0C118	1A01984	Small	1	54	11	11	201	Et0/0	None
80F0C324	1A01AC4	Small	1	54	11	11	201	Et0/0	None
80F0C530	1A01C04	Small	1	54	11	11	201	Et0/0	None
80F0C73C	1A01D44	Small	1	54	11	11	201	Et0/0	None
80F5F644	1B9B0A4	Small	1	54	11	11	201	Et0/0	None
80FDF118	1B78604	Small	1	54	11	11	201	Et0/0	None
80FDF324	1B78744	Small	1	54	11	11	201	Et0/0	None
80FDF530	1B78884	Small	1	54	11	11	201	Et0/0	None
80FDF73C	1B789C4	Small	1	54	11	11	201	Et0/0	None
80FDF948	1B78B04	Small	1	54	11	11	201	Et0/0	None
80FDFB54	1B78C44	Small	1	54	11	11	201	Et0/0	None
80FDFD60	1B78D84	Small	1	54	11	11	201	Et0/0	None
80FDFE6C	1B78EC4	Small	1	54	11	11	201	Et0/0	None
80FE0178	1B79004	Small	1	54	11	11	201	Et0/0	None
80FE0384	1B79144	Small	1	54	11	11	201	Et0/0	None
80FE0590	1B79284	Small	1	54	11	11	201	Et0/0	None
80FE079C	1B793C4	Small	1	54	11	11	201	Et0/0	None
80FE09A8	1B79504	Small	1	54	11	11	201	Et0/0	None
80FE0BB4	1B79644	Small	1	54	11	11	201	Et0/0	None
80FE0DC0	1B79784	Small	1	54	11	11	201	Et0/0	None

80FE0FCC	1B798C4	Small	1	54	11	11	201	Et0/0	None
80FE11D8	1B79A04	Small	1	54	11	11	201	Et0/0	None
80FE13E4	1B79B44	Small	1	54	11	11	201	Et0/0	None
80FE15F0	1B79C84	Small	1	54	11	11	201	Et0/0	None
80FE17FC	1B79DC4	Small	1	54	11	11	201	Et0/0	None
80FE1A08	1B79F04	Small	1	54	11	11	201	Et0/0	None
80FE1C14	1B7A044	Small	1	54	11	11	201	Et0/0	None
80FE1E20	1B7A184	Small	1	54	11	11	201	Et0/0	None
80FE202C	1B7A2C4	Small	1	54	11	11	201	Et0/0	None
80FE2238	1B7A404	Small	1	54	11	11	201	Et0/0	None
81107F40	1B9B1E4	Small	1	54	11	11	201	Et0/0	None
8110814C	1B9B324	Small	1	54	11	11	201	Et0/0	None
81108358	1B9B464	Small	1	54	11	11	201	Et0/0	None
81108564	1B9B5A4	Small	1	54	11	11	201	Et0/0	None
8110897C	1B9B824	Small	1	54	11	11	201	Et0/0	None
81108B88	1B9B964	Small	1	54	11	11	201	Et0/0	None
81108D94	1B9BAA4	Small	1	54	11	11	201	Et0/0	None
81108FA0	1B9BBE4	Small	1	54	11	11	201	Et0/0	None
811093B8	1B9BE64	Small	1	54	11	11	201	Et0/0	None
811095C4	1B9BFA4	Small	1	54	11	11	201	Et0/0	None
811097D0	1B9C0E4	Small	1	54	11	11	201	Et0/0	None
811099DC	1B9C224	Small	1	54	11	11	201	Et0/0	None
81109DF4	1B9C4A4	Small	1	54	11	11	201	Et0/0	None
8110A000	1B9C5E4	Small	1	54	11	11	201	Et0/0	None
8110A20C	1B9C724	Small	1	54	11	11	201	Et0/0	None
8110A418	1B9C864	Small	1	54	11	11	201	Et0/0	None
81121364	1B9CC24	Small	1	54	11	11	201	Et0/0	None
81121570	1B9CD64	Small	1	54	11	11	201	Et0/0	None
81121988	1B9CFE4	Small	1	54	11	11	201	Et0/0	None
81121B94	1B9D124	Small	1	54	11	11	201	Et0/0	None
81121FAC	1B9D3A4	Small	1	54	11	11	201	Et0/0	None
811221B8	1B9D4E4	Small	1	54	11	11	201	Et0/0	None
811225D0	1B9D764	Small	1	54	11	11	201	Et0/0	None
811227DC	1B9D8A4	Small	1	54	11	11	201	Et0/0	None
811229E8	1B9D9E4	Small	1	54	11	11	201	Et0/0	None
81122BF4	1B9DB24	Small	1	54	11	11	201	Et0/0	None

Router#**show buffers old header**

Buffer information for Small buffer at 0x80F09828

```

data_area 0x1A00084, refcount 1, next 0x0, flags 0x201
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
if_input 0x80F57BE0 (Ethernet0/0), if_output 0x0 (None)
inputtime 0x4CDFC58, outputtime 0x0, oqnumber 65535
datagramstart 0x1A000CA, datagramsize 54, maximum size 260
mac_start 0x1A000CA, addr_start 0x1A000CA, info_start 0x0
network_start 0x1A000D8, transport_start 0x0

```

source:BE200040.0060.09c3.f9fe socket 0453

destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01

Buffer information for Small buffer at 0x80F09A34

```

data_area 0x1A001C4, refcount 1, next 0x0, flags 0x201
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
if_input 0x80F57BE0 (Ethernet0/0), if_output 0x0 (None)
inputtime 0x4CDFAA0, outputtime 0x0, oqnumber 65535
datagramstart 0x1A0020A, datagramsize 54, maximum size 260
mac_start 0x1A0020A, addr_start 0x1A0020A, info_start 0x0
network_start 0x1A00218, transport_start 0x0

```

source:BE200040.0060.09c3.f9fe socket 0453

destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01

Buffer information for Small buffer at 0x80F09C40

```

data_area 0x1A00304, refcount 1, next 0x0, flags 0x201
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
if_input 0x80F57BE0 (Ethernet0/0), if_output 0x0 (None)
inputtime 0x4CDF8D7, outputtime 0x0, oqnumber 65535
datagramstart 0x1A0034A, datagramsize 54, maximum size 260
mac_start 0x1A0034A, addr_start 0x1A0034A, info_start 0x0
network_start 0x1A00358, transport_start 0x0

```

```

source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01

```

....

Router#show buffers input-interface ethernet 0/0

Header	DataArea	Pool	Rcnt	Size	Link	Enc	Flags	Input	Output
80F09828	1A00084	Small	1	54	11	11	201	Et0/0	None
80F09A34	1A001C4	Small	1	54	11	11	201	Et0/0	None
80F09C40	1A00304	Small	1	54	11	11	201	Et0/0	None
80F09E4C	1A00444	Small	1	54	11	11	201	Et0/0	None
80F0A058	1A00584	Small	1	54	11	11	201	Et0/0	None
80F0A264	1A006C4	Small	1	54	11	11	201	Et0/0	None
80F0A470	1A00804	Small	1	54	11	11	201	Et0/0	None
80F0A67C	1A00944	Small	1	54	11	11	201	Et0/0	None
80F0A888	1A00A84	Small	1	54	11	11	201	Et0/0	None
80F0AA94	1A00BC4	Small	1	54	11	11	201	Et0/0	None
80F0ACA0	1A00D04	Small	1	54	11	11	201	Et0/0	None
80F0AEAC	1A00E44	Small	1	54	11	11	201	Et0/0	None
80F0B0B8	1A00F84	Small	1	54	11	11	201	Et0/0	None
80F0B2C4	1A010C4	Small	1	54	11	11	201	Et0/0	None
80F0B4D0	1A01204	Small	1	54	11	11	201	Et0/0	None
80F0B6DC	1A01344	Small	1	54	11	11	201	Et0/0	None
80F0B8E8	1A01484	Small	1	54	11	11	201	Et0/0	None
80F0BAF4	1A015C4	Small	1	54	11	11	201	Et0/0	None
80F0BD00	1A01704	Small	1	54	11	11	201	Et0/0	None
80F0BF0C	1A01844	Small	1	54	11	11	201	Et0/0	None
80F0C118	1A01984	Small	1	54	11	11	201	Et0/0	None
80F0C324	1A01AC4	Small	1	54	11	11	201	Et0/0	None
80F0C530	1A01C04	Small	1	54	11	11	201	Et0/0	None
80F0C73C	1A01D44	Small	1	54	11	11	201	Et0/0	None
80F5F644	1B9B0A4	Small	1	54	11	11	201	Et0/0	None
80FDF118	1B78604	Small	1	54	11	11	201	Et0/0	None
80FDF324	1B78744	Small	1	54	11	11	201	Et0/0	None
80FDF530	1B78884	Small	1	54	11	11	201	Et0/0	None
80FDF73C	1B789C4	Small	1	54	11	11	201	Et0/0	None
80FDF948	1B78B04	Small	1	54	11	11	201	Et0/0	None
80FDFB54	1B78C44	Small	1	54	11	11	201	Et0/0	None
80FDFD60	1B78D84	Small	1	54	11	11	201	Et0/0	None
80FDFE6C	1B78EC4	Small	1	54	11	11	201	Et0/0	None
80FE0178	1B79004	Small	1	54	11	11	201	Et0/0	None
80FE0384	1B79144	Small	1	54	11	11	201	Et0/0	None
80FE0590	1B79284	Small	1	54	11	11	201	Et0/0	None
80FE079C	1B793C4	Small	1	54	11	11	201	Et0/0	None
80FE09A8	1B79504	Small	1	54	11	11	201	Et0/0	None
80FE0BB4	1B79644	Small	1	54	11	11	201	Et0/0	None
80FE0DC0	1B79784	Small	1	54	11	11	201	Et0/0	None
80FE0FCC	1B798C4	Small	1	54	11	11	201	Et0/0	None
80FE11D8	1B79A04	Small	1	54	11	11	201	Et0/0	None
80FE13E4	1B79B44	Small	1	54	11	11	201	Et0/0	None
80FE15F0	1B79C84	Small	1	54	11	11	201	Et0/0	None
80FE17FC	1B79DC4	Small	1	54	11	11	201	Et0/0	None
80FE1A08	1B79F04	Small	1	54	11	11	201	Et0/0	None
80FE1C14	1B7A044	Small	1	54	11	11	201	Et0/0	None

80FE1E20	1B7A184	Small	1	54	11	11	201	Et0/0	None
80FE202C	1B7A2C4	Small	1	54	11	11	201	Et0/0	None
80FE2238	1B7A404	Small	1	54	11	11	201	Et0/0	None
81107F40	1B9B1E4	Small	1	54	11	11	201	Et0/0	None
8110814C	1B9B324	Small	1	54	11	11	201	Et0/0	None
81108358	1B9B464	Small	1	54	11	11	201	Et0/0	None
81108564	1B9B5A4	Small	1	54	11	11	201	Et0/0	None
8110897C	1B9B824	Small	1	54	11	11	201	Et0/0	None
81108B88	1B9B964	Small	1	54	11	11	201	Et0/0	None
81108D94	1B9BAA4	Small	1	54	11	11	201	Et0/0	None
81108FA0	1B9BBE4	Small	1	54	11	11	201	Et0/0	None
811093B8	1B9BE64	Small	1	54	11	11	201	Et0/0	None
811095C4	1B9BFA4	Small	1	54	11	11	201	Et0/0	None
811097D0	1B9C0E4	Small	1	54	11	11	201	Et0/0	None
811099DC	1B9C224	Small	1	54	11	11	201	Et0/0	None
81109DF4	1B9C4A4	Small	1	54	11	11	201	Et0/0	None
8110A000	1B9C5E4	Small	1	54	11	11	201	Et0/0	None
8110A20C	1B9C724	Small	1	54	11	11	201	Et0/0	None
8110A418	1B9C864	Small	1	54	11	11	201	Et0/0	None
81121364	1B9CC24	Small	1	54	11	11	201	Et0/0	None
81121570	1B9CD64	Small	1	54	11	11	201	Et0/0	None
81121988	1B9CFE4	Small	1	54	11	11	201	Et0/0	None
81121B94	1B9D124	Small	1	54	11	11	201	Et0/0	None
81121FAC	1B9D3A4	Small	1	54	11	11	201	Et0/0	None
811221B8	1B9D4E4	Small	1	54	11	11	201	Et0/0	None
811225D0	1B9D764	Small	1	54	11	11	201	Et0/0	None
811227DC	1B9D8A4	Small	1	54	11	11	201	Et0/0	None
811229E8	1B9D9E4	Small	1	54	11	11	201	Et0/0	None
81122BF4	1B9DB24	Small	1	54	11	11	201	Et0/0	None

Router#show buffers address 81122BF4 dump

```

Buffer information for Small buffer at 0x81122BF4
  data_area 0x1B9DB24, refcount 1, next 0x0, flags 0x201
  linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
  if_input 0x80F57BE0 (Ethernet0/0), if_output 0x0 (None)
  inputtime 0x4CE2BFC, outputtime 0x0, oqnumber 65535
  datagramstart 0x1B9DB6A, datagramsize 54, maximum size 260
  mac_start 0x1B9DB6A, addr_start 0x1B9DB6A, info_start 0x0
  network_start 0x1B9DB78, transport_start 0x0

source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01

```

```

01B9DB20: 00000000 00000000 00000000 00000000 .....
01B9DB30: 00000000 00000000 00000000 00000000 .....
01B9DB40: 00000000 00000000 00000000 00000000 .....
01B9DB50: 00000000 00000000 00000000 00000000 .....
01B9DB60: 00000000 00000000 0000FFFF FFFFFFFF .....
01B9DB70: 006009C3 F9FE0028 FFFF0028 0001BE20 .`.Cy~.(...(>
01B9DB80: 0040FFFF FFFFFFFF 0453BE20 00400060 .@.....S> .@.`
01B9DB90: 09C3F9FE 04530001 00000040 06000200 .Cy~.S.....@....
01B9DBA0: 00000000 00000000 00000000 00000000 .....
01B9DBB0: 00000000 00000000 00000000 00000000 .....
01B9DBC0: 00000000 00000000 00000000 00000000 .....
01B9DBD0: 00000000 00000000 00000000 00000000 .....
01B9DBE0: 00000000 00000000 00000000 00000000 .....
01B9DBF0: 00000000 00000000 00000000 00000000 .....
01B9DC00: 00000000 00000000 00000000 00000000 .....
01B9DC10: 00000000 00000000 00000000 00000000 .....
01B9DC20: 00000000 00 .....

```

Router#

Wenn Sie ein Muster in den Puffern nicht identifizieren können, erfassen Sie die Ausgabe der **show**-Befehle (z. B. **show buffers old**), und speichern Sie sie in einer Datei (z. B. buffers.log). Versuchen Sie dann, das Muster mithilfe des UNIX-Dienstprogramms "grep" oder etwas Ähnliches zu isolieren.

```
grep linktype buffers.log
```

```
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 0 (None), enctype 0 (None), encsize 0, rxtype 0
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
linktype 11 (NOVELL), enctype 11 (NOVELL-ETHER), encsize 14, rxtype 7
```

```
...
```

```
!--- Here you can clearly see a lot of NOVELL-related buffers.
```

```
!--- The problem seems to be with the IPX packets.
```

```
!--- You can check this through the wc -l (to count lines) command on a UNIX system.
```

```
grep linktype buffers.log | wc -l
```

```
175
```

```
grep linktype buffers.log | grep NOVELL-ETHER | wc -l
```

```
153
```

```
!--- 153 out of 175 old buffers are IPX packets. Try to find out what
```

```
!--- type of packets they are with another grep command:
```

```
grep socket buffers.log
```

```
source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01
source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01
source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01
source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01
source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01
source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01
source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01
source:BE200040.0060.09c3.f9fe socket 0453
destination: BE200040.ffff.ffff.ffff socket 0453 protocol 01
source:BE200040.0060.09c3.f9fe socket 0453
```

```
...
```

```
!--- There are Broadcasts to socket 453, protocol 01...
```

```
!--- Those are IPX RIP packets.
```

```
!--- Disable IPX RIP, or use IPX EIGRP instead, until a bug fix is available.
```

Zusammenfassung:

- Überprüfen Sie, ob ein Pufferleck vorliegt. Pufferüberläufe werden häufig falsch interpretiert als ein sprunghafter Datenverkehrsaufkommen (bei dem viele Pakete aufgrund einer falschen Konfiguration oder einer nicht unterstützten Funktion an Prozess-Switching gehen) oder als Angriff.
- Pufferleaks sind Cisco IOS Software-Bugs. Die beste Lösung für dieses Problem ist ein Upgrade der Cisco IOS Software auf die neueste Version.
- Falls dies fehlschlägt, wenden Sie sich an das Cisco TAC und stellen Sie dem Techniker die Ausgabe der relevanten **Show-Puffer** und **zeigen die Befehle für den technischen Support** an.

Zugehörige Informationen

- [Puffer-Tuning](#)
- [Fehlerbehebung bei Speicherfehlern](#)
- [Technischer Support – Cisco Systems](#)