



トレース管理

この章の内容は、次のとおりです。

- [トレース管理](#) (1 ページ)
- [トレースの機能](#) (1 ページ)
- [トレースレベル](#) (5 ページ)
- [トレースレベルの表示](#) (6 ページ)
- [トレースレベルの設定](#) (8 ページ)
- [トレースバッファの内容の表示](#) (8 ページ)
- [例：パケットトレースの使用](#) (8 ページ)

トレース管理

トレースは、内部イベントをログする機能です。トレースメッセージを含むトレースファイルが自動的に作成され、ルータの `hard disk`: ファイルシステムの `tracelog` ディレクトリに保存されます（ブートフラッシュにトレースファイルが保存されます）。

トレースファイルのデータは、次の処理を行う場合に役立ちます。

- **トラブルシューティング**：ルータの問題を特定して解決するのに役立ちます。システムで他の問題が同時に発生している場合でも、診断モードでトレースファイルにアクセスできます。
- **デバッグ**：システムアクションと操作の詳細を取得するのに役立ちます。

トレースの機能

トレースは、ルータの内部イベントの内容を記録します。モジュールに関するすべてのトレース出力を含むトレースファイルが定期的に作成および更新され、`tracelog` ディレクトリに保存されます。トレースファイルは、システムパフォーマンスに影響を及ぼすことなく、このディレクトリから消去して、ファイルシステムのスペースを回復することができます。ファイル転送機能（FTP、TFTPなど）を使用してこれらのファイルを他の宛先にコピーできます。また、プレーンテキストエディタで開くことができます。



(注) ルータでトレースをディセーブルにすることはできません。

トレース情報を表示し、トレースレベルを設定するには、次のコマンドを使用します。

- **show logging process module** : 特定のモジュールに関する最新のトレース情報を表示します。このコマンドは特権 EXEC モードおよび診断モードで使用可能です。診断モードでこのコマンドを使用すると、Cisco IOS XE の障害発生時にトレース ログ情報を収集できます。
- **set platform software trace** : 出力に保存されるメッセージのタイプを決定するトレースレベルを設定します。トレース レベルの詳細については、[トレースレベル \(5 ページ\)](#) を参照してください。

UDF オフセットを使用したパケットトレーサの設定

オフセットを使用してパケットトレース UDF を設定するには、次の手順を実行します。

手順

ステップ 1 enable

例 :

```
Device> enable
```

特権 EXEC モードを有効にします。

- パスワードを入力します (要求された場合)。

ステップ 2 configure terminal

例 :

```
Device# configure terminal
```

グローバル コンフィギュレーション モードを開始します。

ステップ 3 udf udf name header {inner | outer} {13|14} offset offset-in-bytes length length-in-bytes

例 :

```
Router(config)# udf TEST_UDF_NAME_1 header inner 13 64 1
```

```
Router(config)# udf TEST_UDF_NAME_2 header inner 14 77 2
```

```
Router(config)# udf TEST_UDF_NAME_3 header outer 13 65 1
```

```
Router(config)# udf TEST_UDF_NAME_4 header outer 14 67 1
```

個々の UDF 定義を設定します。UDF の名前、オフセット元のネットワーキングヘッダー、抽出するデータの長さを指定できます。

inner キーワードまたは **outer** キーワードは、カプセル化されていないレイヤ 3 またはレイヤ 4 のヘッダーからのオフセットの開始を指定するか、またはカプセル化されたパケットがある場合は内部 L3/L4 からのオフセットの開始を指定します。

length キーワードはオフセットからの長さをバイト単位で指定します。有効な範囲は 1 ~ 2 です。

ステップ 4 **udf udf name {header | packet-start} offset-base offset length**

例 :

```
Router(config)# udf TEST_UDF_NAME_5 packet-start 120 1
```

- **header** : オフセットの基本設定を指定します。
- **packet-start** : **packet-start** からのオフセットベースを指定します。 **packet-start** は、パケットトレーサがインバウンドパケット用かアウトバウンドパケット用かによって異なります。パケットトレーサがインバウンドパケット用である場合、パケット開始はレイヤ 2 になります。アウトバウンドの場合、**packet-start** はレイヤ 3 になります。
- **offset** : オフセットベースからオフセットさせるバイト数を指定します。オフセットベース (レイヤ 3/レイヤ 4 ヘッダー) からの先頭バイトに一致させるには、オフセットを 0 に設定します。
- **length** : オフセットからのバイト数を指定します。1 バイトまたは 2 バイトだけがサポートされます。追加のバイト数に一致させるには、複数の UDF の定義が必要です。

ステップ 5 **ip access-list extended {acl-name [acl-num]}**

例 :

```
Router(config)# ip access-list extended acl2
```

拡張 ACL コンフィギュレーション モードを有効にします。CLI は拡張 ACL コンフィギュレーション モードを開始します。このモードでは、後続のすべてのコマンドが現在の拡張アクセスリストに適用されます。拡張 ACL は、IP パケットの送信元アドレスおよび宛先アドレスを ACL に設定されているアドレスと比較して、トラフィックを制御します。

ステップ 6 **ip access-list extended {deny | permit } udf udf-name value mask**

例 :

```
Router(config-acl)# permit ip any any udf TEST_UDF_NAME_5 0xD3 0xFF
```

現在のアクセス制御エントリ (ACE) と併せて、UDF で一致するように ACL を設定します。ACL で定義されているバイトは 0xD3 です。マスクは、許可および拒否するトラフィックを指定するように、IP ACL で IP アドレスとともに使用します。

ステップ 7 **debug platform condition [ipv4 | ipv6] [interface interface] [access-list access-list -name | ipv4-address / subnet-mask | ipv6-address / subnet-mask] [ingress | egress |both]**

例 :

```
Router# debug platform condition interface gi0/0/0 ipv4 access-list acl2 both
```

パケットをトレースするための一致基準を指定します。プロトコル、IP アドレスおよびサブネットマスク、アクセス制御リスト（ACL）、インターフェイス、方向によるフィルタリング機能を提供します。

ステップ 8 **debug platform condition start**

例：

```
Router# debug platform condition start
```

指定した位置基準を有効にしてパケットトレースを開始します。

ステップ 9 **debug platform packet-trace packet *pkt-num* [**fia-trace** | **summary-only**] [**circular**] [**data-size** *data-size*]**

例：

```
Router# debug platform packet-trace packet 1024 fia-trace data-size 2048
```

指定した数のパケットのサマリーデータを収集します。デフォルトでは機能パスデータをキャプチャし、必要に応じて FIA トレースを実行します。

pkt-num：所定の時間に維持されるパケットの最大数を指定します。

fia-trace：サマリーデータ、機能固有のデータなど、詳細なレベルのデータキャプチャを実行します。また、パケット処理中にアクセスされた各機能エントリも表示します。

summary-only：詳細情報を最小限にしたサマリーデータのキャプチャを有効にします。

circular：最近トレースされたパケットのデータを保存します。

data-size：各パケットの機能データと FIA トレースデータを保存するデータバッファのサイズをバイト単位で指定します。パケットで非常に重いパケット処理が実行された場合、ユーザーは必要に応じてデータバッファのサイズを増やすことができます。デフォルト値は 2048 です。

ステップ 10 **debug platform packet-trace {**punt** | **inject**|**copy** | **drop** |**packet** | **statistics**}**

例：

```
Router# debug platform packet-trace punt
```

データからコントロールプレーンへパントされたパケットのトレースを有効にします。

ステップ 11 **debug platform condition stop**

例：

```
Router# debug platform condition start
```

条件を非アクティブにして、パケットのトレースを停止します。

ステップ 12 **exit**

例：

```
Router# exit
```

特権 EXEC モードを終了します。

トレースレベル

トレースレベルは、トレースバッファまたはトレースファイルに保存する必要のあるモジュール情報の量を決定します。

次の表に、使用可能なすべてのトレースレベルと、各トレースレベルで表示されるメッセージのタイプについて説明します。

表 1: トレースレベルとその内容

トレースレベル	レベル番号	説明
緊急	0	システムが使用不能になる問題のメッセージです。
アラート	1	ただちに対応する必要がある動作についてのメッセージです。
重要	2	クリティカルな状態についてのメッセージです。これは、ルータ上のすべてのモジュールに関するデフォルト設定です。
エラー	3	システムエラーについてのメッセージです。
警告	4	システム警告についてのメッセージです。
通知	5	重大な問題に関するメッセージです。ただし、ルータは通常どおり動作しています。
情報	6	単に情報を提供するだけのメッセージです。
デバッグ	7	デバッグレベルの出力を提供するメッセージです。
詳細	8	生成可能なすべてのトレースメッセージが送信されます。

トレースレベル	レベル番号	説明
ノイズ	—	モジュールについて生成可能なすべてのトレースメッセージが記録されます。 ノイズレベルは常に最上位のトレースレベルに相当します。トレース機能の今後の拡張によって、 Verbose レベルよりも高いトレースレベルが導入される場合でも、 Noise レベルは新規に導入されるトレースレベルと同等になります。

トレースレベルが設定されている場合、設定されているトレースレベル自体と、それより低いすべてのトレースレベルの両方のメッセージが収集されます。

たとえば、トレースレベルを3（エラー）に設定すると、トレースファイルにはレベル0（緊急）、1（アラート）、2（重要）、および3（エラー）のメッセージが出力されます。

トレースレベルを4（警告）に設定すると、レベル0（緊急）、1（アラート）、2（重要）、3（エラー）、および4（警告）のメッセージが出力されます。

ルータのすべてのモジュールのデフォルトトレースレベルは5（通知）です。

トレースレベルは、コンフィギュレーションモードでは設定されません。このため、ルータのリロード後にトレースレベル設定がデフォルト値に戻ります。



注意 モジュールのトレースレベルをデバッグレベル以上に設定すると、パフォーマンスに悪影響を及ぼす可能性があります。



注意 多数のモジュールで高いトレースレベルを設定すると、パフォーマンスが大幅に低下する可能性があります。特定の状況で高いトレースレベルが必要な場合は、複数のモジュールで高いレベルを設定する代わりに、常に1つのモジュールのトレースレベルを高く設定することをお勧めします。

トレースレベルの表示

デフォルトでは、ルータ上のすべてのモジュールが5（通知）に設定されます。ユーザが変更しないかぎり、この設定はそのまま維持されます。

ルータのモジュールのトレースレベルを表示するには、特権EXECモードまたは診断モードで **show logging process** コマンドを入力します。

次の例では、**show logging process** コマンドを使用して、アクティブな RP 上のフォワーディング マネージャ プロセスのトレースレベルを表示します。

```
Router# show logging process forwarding-manager rp active
Module Name                               Trace Level
-----
acl                                         Notice
binos                                       Notice
binos/brand                               Notice
bipc                                        Notice
bsignal                                    Notice
btrace                                     Notice
cce                                         Notice
cdllib                                     Notice
cef                                         Notice
chasfs                                     Notice
chasutil                                   Notice
erspan                                     Notice
ess                                         Notice
ether-channel                             Notice
evlib                                       Notice
evutil                                     Notice
file_alloc                                 Notice
fman_rp                                    Notice
fpm                                         Notice
fw                                          Notice
icmp                                        Notice
interfaces                                Notice
iosd                                       Notice
ipc                                         Notice
ipclog                                     Notice
iphc                                       Notice
IPsec                                      Notice
mgmte-acl                                  Notice
mlp                                         Notice
mqipc                                      Notice
nat                                         Notice
nbar                                        Notice
netflow                                    Notice
om                                          Notice
peer                                       Notice
qos                                         Notice
route-map                                  Notice
sbc                                         Notice
services                                   Notice
sw_wdog                                    Notice
tdl_acl_config_type                       Notice
tdl_acl_db_type                           Notice
tdl_cdlcore_message                       Notice
tdl_cef_config_common_type                Notice
tdl_cef_config_type                       Notice
tdl_dpiddb_config_type                    Notice
tdl_fman_rp_comm_type                     Notice
tdl_fman_rp_message                       Notice
tdl_fw_config_type                         Notice
tdl_hapi_tdl_type                          Notice
tdl_icmp_type                              Notice
tdl_ip_options_type                       Notice
tdl_ipc_ack_type                          Notice
tdl_IPsec_db_type                         Notice
tdl_mcp_comm_type                         Notice
tdl_mlp_config_type                       Notice
tdl_mlp_db_type                           Notice
tdl_om_type                                Notice
```

tdl_ui_message	Notice
tdl_ui_type	Notice
tdl_urpf_config_type	Notice
tdllib	Notice
trans_avl	Notice
uihandler	Notice
uipeer	Notice
uistatus	Notice
urpf	Notice
vista	Notice
wccp	Notice

トレースレベルの設定

ルータに含まれる1つのモジュールのトレースレベル、またはルータにおける特定プロセスに含まれるすべてのモジュールのトレースレベルを設定するには、特権EXECモードまたは診断モードで **set platform software trace** コマンドを入力します。

次の例では、スロット0のESPプロセッサのForwarding ManagerでACLモジュールに関するトレースレベルを `info` に設定します。

```
set platform software trace forwarding-manager F0 acl info
```

トレースバッファの内容の表示

トレースバッファ内またはファイル内のトレースメッセージを表示するには、特権EXECモードまたは診断モードで **show logging process** コマンドを入力します。次の例では、**show logging process command** コマンドを使用して、Route Processor スロット0でのHost Managerプロセスのトレースメッセージを表示します。

```
Router# show logging process host-manager R0
08/23 12:09:14.408 [uipeer]: (info): Looking for a ui_req msg
08/23 12:09:14.408 [uipeer]: (info): Start of request handling for con 0x100a61c8
08/23 12:09:14.399 [uipeer]: (info): Accepted connection for 14 as 0x100a61c8
08/23 12:09:14.399 [uipeer]: (info): Received new connection 0x100a61c8 on descriptor
14
08/23 12:09:14.398 [uipeer]: (info): Accepting command connection on listen fd 7
08/23 11:53:57.440 [uipeer]: (info): Going to send a status update to the shell manager
in slot 0
08/23 11:53:47.417 [uipeer]: (info): Going to send a status update to the shell manager
in slot 0
```

例：パケットトレースの使用

次に、パケットトレースを使用してCisco ASR 1006ルータのNAT設定でパケットドロップのトラブルシューティングを行うシナリオの例を示します。この例には、パケットトレース機能によって提供される詳細レベルを効果的に利用して問題に関する情報を収集し、問題を切り分けて、解決策を見つける方法が示されています。

このシナリオでは、問題があることはわかりますが、どこからトラブルシューティングを開始すればよいかはわかりません。したがって、多数の着信パケットのパケットトレースのサマリーにアクセスすることを検討する必要があります。

```
Router# debug platform condition ingress
Router# debug platform packet-trace packet 2048 summary-only
Router# debug platform condition start
Router# debug platform condition stop
Router# show platform packet-trace summary
Pkt  Input          Output          State  Reason
0    Gi0/0/0         Gi0/0/0        DROP   402 (NoStatsUpdate)
1    internal0/0/rp:0 internal0/0/rp:0 PUNT   21  (RP<->QFP keepalive)
2    internal0/0/recycle:0 Gi0/0/0        FWD
```

この出力には、ギガビットイーサネットインターフェイス 0/0/0 の NAT 設定が原因でパケットがドロップされていることが示されています。これによって、問題は特定のインターフェイスで発生していることがわかります。この情報を使用して、トレースするパケットを制限し、データキャプチャのパケット数を減らし、検査レベルを上げることができます。

```
Router# debug platform packet-trace packet 256
Router# debug platform packet-trace punt
Router# debug platform condition interface Gi0/0/0
Router# debug platform condition start
Router# debug platform condition stop
Router# show platform packet-trace summary
Router# show platform packet-trace 15
Packet: 15          CBUG ID: 238
Summary
  Input      : GigabitEthernet0/0/0
  Output     : internal0/0/rp:1
  State      : PUNT 55 (For-us control)
  Timestamp
    Start    : 1166288346725 ns (06/06/2016 09:09:42.202734 UTC)
    Stop     : 1166288383210 ns (06/06/2016 09:09:42.202770 UTC)
Path Trace
  Feature: IPV4
    Input      : GigabitEthernet0/0/0
    Output     : <unknown>
    Source     : 10.64.68.3
    Destination : 224.0.0.102
    Protocol   : 17 (UDP)
    SrcPort    : 1985
    DstPort    : 1985
IOSd Path Flow: Packet: 15    CBUG ID: 238
  Feature: INFRA
    Pkt Direction: IN
    Packet Rcvd From CPP
  Feature: IP
    Pkt Direction: IN
    Source      : 10.64.68.122
    Destination : 10.64.68.255
  Feature: IP
    Pkt Direction: IN
    Packet Enqueued in IP layer
    Source      : 10.64.68.122
    Destination : 10.64.68.255
    Interface   : GigabitEthernet0/0/0
  Feature: UDP
    Pkt Direction: IN
    src          : 10.64.68.122(1053)
```

```
dst      : 10.64.68.255(1947)
length  : 48
```

Router#**show platform packet-trace packet 10**

Packet: 10 CBUG ID: 10

Summary

```
Input      : GigabitEthernet0/0/0
Output     : internal0/0/rp:0
State      : PUNT 55 (For-us control)
Timestamp
  Start    : 274777907351 ns (01/10/2020 10:56:47.918494 UTC)
  Stop     : 274777922664 ns (01/10/2020 10:56:47.918509 UTC)
```

Path Trace

```
Feature: IPV4(Input)
Input      : GigabitEthernet0/0/0
Output     : <unknown>
Source     : 10.78.106.2
Destination : 224.0.0.102
Protocol   : 17 (UDP)
  SrcPort  : 1985
  DstPort  : 1985
```

IOSd Path Flow: Packet: 10 CBUG ID: 10

```
Feature: INFRA
Pkt Direction: IN
```

Packet Rcvd From DATAPLANE

```
Feature: IP
Pkt Direction: IN
Packet Enqueued in IP layer
Source      : 10.78.106.2
Destination : 224.0.0.102
Interface   : GigabitEthernet0/0/0
```

```
Feature: UDP
Pkt Direction: IN DROP
Pkt : DROPPED
UDP: Discarding silently
src      : 881 10.78.106.2(1985)
dst      : 224.0.0.102(1985)
length   : 60
```

Router#**show platform packet-trace packet 12**

Packet: 12 CBUG ID: 767

Summary

```
Input      : GigabitEthernet3
Output     : internal0/0/rp:0
State      : PUNT 11 (For-us data)
Timestamp
  Start    : 16120990774814 ns (01/20/2020 12:38:02.816435 UTC)
  Stop     : 16120990801840 ns (01/20/2020 12:38:02.816462 UTC)
```

Path Trace

```
Feature: IPV4(Input)
Input      : GigabitEthernet3
Output     : <unknown>
Source     : 12.1.1.1
Destination : 12.1.1.2
Protocol   : 6 (TCP)
  SrcPort  : 46593
  DstPort  : 23
```

IOSd Path Flow: Packet: 12 CBUG ID: 767

```
Feature: INFRA
Pkt Direction: IN
Packet Rcvd From DATAPLANE
```

```

Feature: IP
  Pkt Direction: IN
  Packet Enqueued in IP layer
  Source      : 12.1.1.1
  Destination : 12.1.1.2
  Interface   : GigabitEthernet3

Feature: IP
  Pkt Direction: IN
  FORWARDEDTo transport layer
  Source      : 12.1.1.1
  Destination : 12.1.1.2
  Interface   : GigabitEthernet3

Feature: TCP
  Pkt Direction: IN
  tcp0: I NoTCB 12.1.1.1:46593 12.1.1.2:23 seq 1925377975 OPTS 4 SYN WIN 4128

Router# show platform packet-trace summary
Pkt  Input                Output                State Reason
0    INJ.2                  Gi1                   FWD
1    Gi1                    internal0/0/rp:0     PUNT 11 (For-us data)
2    INJ.2                  Gi1                   FWD
3    Gi1                    internal0/0/rp:0     PUNT 11 (For-us data)
4    INJ.2                  Gi1                   FWD
5    INJ.2                  Gi1                   FWD
6    Gi1                    internal0/0/rp:0     PUNT 11 (For-us data)
7    Gi1                    internal0/0/rp:0     PUNT 11 (For-us data)
8    Gi1                    internal0/0/rp:0     PUNT 11 (For-us data)
9    Gi1                    internal0/0/rp:0     PUNT 11 (For-us data)
10   INJ.2                  Gi1                   FWD
11   INJ.2                  Gi1                   FWD
12   INJ.2                  Gi1                   FWD
13   Gi1                    internal0/0/rp:0     PUNT 11 (For-us data)
14   Gi1                    internal0/0/rp:0     PUNT 11 (For-us data)
15   Gi1                    internal0/0/rp:0     PUNT 11 (For-us data)
16   INJ.2                  Gi1                   FWD
    
```

次に、パケットトレースデータの統計を表示する例を示します。

```

Router#show platform packet-trace statistics
Packets Summary
  Matched 3
  Traced 3
Packets Received
  Ingress 0
  Inject 0
Packets Processed
  Forward 0
  Punt 3
  Count   Code Cause
  3       56  RP injected for-us control
  Drop 0
  Consume 0

          PKT_DIR_IN
          Dropped      Consumed      Forwarded
INFRA      0              0              0
TCP        0              0              0
UDP        0              0              0
IP         0              0              0
IPV6      0              0              0
ARP        0              0              0

          PKT_DIR_OUT
    
```

	Dropped	Consumed	Forwarded
INFRA	0	0	0
TCP	0	0	0
UDP	0	0	0
IP	0	0	0
IPV6	0	0	0
ARP	0	0	0

次に、コントロールプレーンからフォワーディングプロセッサに挿入およびパントされるパケットを表示する例を示します。

```

Router#debug platform condition ipv4 10.118.74.53/32 both
Router#Router#debug platform condition start
Router#debug platform packet-trace packet 200
Packet count rounded up from 200 to 256

Router#show platform packet-tracer packet 0
show plat pack pa 0
Packet: 0          CBUG ID: 674
Summary
  Input       : GigabitEthernet1
  Output      : internal0/0/rp:0
  State       : PUNT 11 (For-us data)
  Timestamp
    Start     : 17756544435656 ns (06/29/2020 18:19:17.326313 UTC)
    Stop      : 17756544469451 ns (06/29/2020 18:19:17.326346 UTC)
  Path Trace
    Feature: IPv4(Input)
      Input       : GigabitEthernet1
      Output      : <unknown>
      Source      : 10.118.74.53
      Destination : 198.51.100.38
      Protocol    : 17 (UDP)
      SrcPort     : 2640
      DstPort     : 500

IOSd Path Flow: Packet: 0    CBUG ID: 674
  Feature: INFRA
  Pkt Direction: IN
  Packet Rcvd From DATAPLANE

  Feature: IP
  Pkt Direction: IN
  Packet Enqueued in IP layer
  Source       : 10.118.74.53
  Destination  : 198.51.100.38
  Interface    : GigabitEthernet1

  Feature: IP
  Pkt Direction: IN
  FORWARDED To transport layer
  Source       : 10.118.74.53
  Destination  : 198.51.100.38
  Interface    : GigabitEthernet1

  Feature: UDP
  Pkt Direction: IN
  DROPPED
  UDP: Checksum error: dropping
  Source       : 10.118.74.53(2640)
  Destination  : 198.51.100.38(500)

Router#show platform packet-tracer packet 2
Packet: 2          CBUG ID: 2

```

```
IOSd Path Flow:
  Feature: TCP
  Pkt Direction: OUTtcp0: O SYNRCVD 198.51.100.38:22 198.51.100.55:52774 seq 3052140910
  OPTS 4 ACK 2346709419 SYN WIN 4128

  Feature: TCP
  Pkt Direction: OUT
  FORWARDED
  TCP: Connection is in SYNRCVD state
  ACK      : 2346709419
  SEQ      : 3052140910
  Source   : 198.51.100.38(22)
  Destination : 198.51.100.55(52774)

  Feature: IP
  Pkt Direction: OUTRoute out the generated packet.srcaddr: 198.51.100.38, dstaddr:
  198.51.100.55

  Feature: IP
  Pkt Direction: OUTInject and forward successful srcaddr: 198.51.100.38, dstaddr:
  198.51.100.55

  Feature: TCP
  Pkt Direction: OUTtcp0: O SYNRCVD 198.51.100.38:22 198.51.100.55:52774 seq 3052140910
  OPTS 4 ACK 2346709419 SYN WIN 4128
Summary
  Input      : INJ.2
  Output     : GigabitEthernet1
  State      : FWD
  Timestamp
    Start    : 490928006866 ns (06/29/2020 13:31:30.807879 UTC)
    Stop     : 490928038567 ns (06/29/2020 13:31:30.807911 UTC)
Path Trace
  Feature: IPV4(Input)
  Input      : internal0/0/rp:0
  Output     : <unknown>
  Source     : 172.18.124.38
  Destination : 172.18.124.55
  Protocol   : 6 (TCP)
  SrcPort    : 22
  DstPort    : 52774
  Feature: IPSec
  Result     : IPSEC_RESULT_DENY
  Action     : SEND_CLEAR
  SA Handle  : 0
  Peer Addr  : 55.124.18.172
  Local Addr : 38.124.18.172

Router#
```


翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。