TCP 認証オプションの設定

本書では、Cisco NX-OS デバイスに TCP 認証オプションを設定する手順について説明します。

TCP 認証オプションについて

RFC 5925 で定義されている TCP 認証オプション（TCP-AO）を使用すると、より強力なメッセージ認証コード（MAC）を使用して、長期間の TCP 接続をリプレイから保護できます。

TCP-AO は、RFC 2385 で定義されている TCP MD5 の代替案です。TCP MD5 とは異なり、TCP-AO はコリジョン攻撃に対する耐性があり、アルゴリズム的俊敏性とキー管理のサポートを提供します。

TCP-AO には次のような顕著な特徴があります。

- TCP-AO は、長時間の TCP 接続のセキュリティを強化するために、より強力なメッセージ認証コード（MAC）の使用をサポートしています。
- TCP-AO は、長期的な TCP 接続のリプレイから保護し、より明示的なキー管理を提供することで、エンドポイント間のキー変更を調整します。

TCP-AO 機能により TCP MD5 は廃止されます。Cisco NX-OS デバイスは、レガシー BGP ピアの TCP-MD5 オプションを引き続きサポートします。ただし、一方の端がデバイスに TCP MD5 オプションが構成され、もう一方のピアリングに TCP-AO オプションが構成されている構成はサポートされていません。

TCP-AO キー チェーン

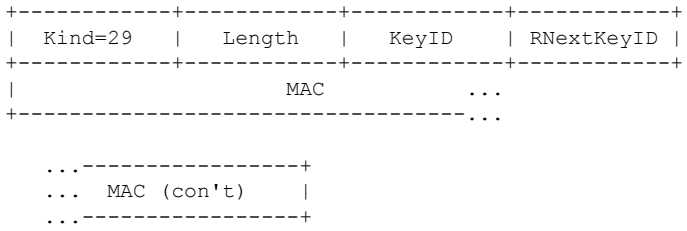
TCP-AO は、トラフィック キー、およびキーと MAC アルゴリズムを使用して生成されたメッセージ認証コード（MAC）に基づいています。トラフィック キーは、TCP-AO キー チェーンで設定できるマスター キーから導出されます。TCP-AO キー チェーンを作成し、チェーン内のキーを設定するには、グローバル コンフィギュレーション モードで **key chain** *key-chain-name* **tcp** コマンドを使用します。TCP 接続を介して通信する両方のピアで TCP-AO キー チェーンを設定する必要があります。

TCP-AO キー チェーンのキーには、次の設定可能なプロパティがあります。

設定可能なプロパティ	説明 （**Description**）
send-id	発信セグメントの TCP-AO オプションのキー識別子。 ルータで構成された送信識別子は、ピアで構成された受信識別子と一致する必要があります。
recv-id	認証時に着信セグメントの TCP-AO キー識別子と比較されるキー識別子。 ルータで構成された受信識別子は、ピアで構成された送信識別子と一致する必要があります。
cryptographic-algorithm	発信セグメントの MAC を作成するために使用される MAC アルゴリズム。アルゴリズムは次のいずれかになります。 • AES-128-CMAC 認証アルゴリズム • HMAC-SHA-1 認証アルゴリズム • HMAC-SHA-256 認証アルゴリズム
include-tcp-options	このフラグは、MAC の計算に TCP-AO 以外の TCP オプションを使用するかどうかを示します。 このフラグを有効にすると、すべてのオプションの内容とゼロで埋められた認証オプションが MAC の計算に使用されます。 フラグを無効にすると、TCP-AO 以外のすべてのオプションが MAC 計算から除外されます。 このフラグはデフォルトでは無効になっています。 （注） このフラグの設定は、アプリケーション設定を使用可能にすると、アプリケーション設定によって上書きされます。

設定可能なプロパティ	説明 （**Description**）
send-lifetime	この設定は、キーが有効であり、TCP セグメントの TCP-AO ベースの認証に使用できる時間を決定します。キーのライフタイムが経過し、キーが期限切れになると、ライフタイムが最も若い次のキーが選択されます。
key-string	キー文字列は、両方のピアで設定された事前共有マスターキーであり、トラフィック キーを導出するために使用されます。

TCP-AO 形式



TLV 形式のフィールドは、次のとおりです。

- Kind：TCP-AO を示す 29 という値。
- Length：TCP-AO シーケンスの長さを示します。
- KeyID：トラフィック キーの生成に使用されるマスター キー タプル（MKT）の送信識別子。
- RNextKeyID：受信したセグメントの認証に使用できる MKT の受信識別子。
- MAC：TCP セグメント データとプレフィックス付き疑似ヘッダーに対して計算された MAC。

マスター キー タプル

トラフィック キーは、個々の TCP セグメントのメッセージ認証コードを計算するために使用されるキー情報です。

マスター キー タプル（MKT）を使用すると、一意のトラフィック キーを導出し、それらのトラフィック キーの生成に必要なキーマテリアルを含めることができます。MKTは、トラフィック キーが設定されるパラメータを示します。パラメータには、TCP オプションが認証されているかどうか、そしてトラフィック キーの導出およびMAC 計算に使用されるアルゴリズムの指示子が含まれます。

各 MKT には、次の 2 つの識別子があります：

- **SendID**：**SendID** 識別子は、発信セグメントの TCP AO オプションの KeyID 識別子として挿入されます。

- **RecvID**： **RecvID は** 、着信セグメントの TCP AO キー ID と照合されます。

TCP-AO キー ロールオーバー

TCP-AO キーは、send-lifetime を使用して設定された定義済みの期間有効です。send-lifetime が設定されていない場合、キーは非アクティブと見なされます。キー ロールオーバーは、キーの送信ライフタイムに基づいて開始されます。

TCP-AO は、TCP-AO オプション フィールドの RNextKeyID および KeyID フィールドを使用して、新しい MKT の使用を調整します。ヒットレス キー ロールオーバーの場合、キーチェーン設定の新しいキーと古いキーには、少なくとも 15 分間のオーバーラップが必要です。これは、TCP-AO が新しい MKT の使用を調整するのに十分な時間を確保するために必要です。

キー ロールオーバーが開始されると、ピア ルータの 1 つ（たとえばルータ A）が、ロールオーバーが必要であることを示します。ロールオーバーが必要であることを示すために、ルータ A は使用する新しい MKT の受信識別子（recv-id）に RNextKeyID を設定します。TCP セグメントを受信すると、ピア ルータ（たとえばルータ B）は、データベースで送信識別子（send-id）を検索して、TCP-AO ペイロードの RNextKeyID によって示される MKT を見つけます。キーが使用可能で有効な場合、ルータ B は現在のキーを新しい MKT に設定します。ルータ B がロールオーバーした後、ルータ A も現在のキーを新しいプライマリ キー タプルに設定します。

送信ライフタイムと送信ライフタイムの有効期限が重複してキー ロールオーバーが開始されます。

現在のキーの有効期限が切れる前にアクティブ化できる新しいキーを設定しないと、キーがタイムアウトして期限切れになる可能性があります。このような期限が切れると、ピア ルータが期限切れのキーで認証されたセグメントを拒否し、再送信が発生することがあります。再送信タイムアウト（RTO）が原因で接続が失敗する可能性があります。新しい有効なキーが構成済みで使用可能な場合、接続を再確立することができます。

注意事項と制約事項

- キーチェーン内の各キーの send-id と recv-id は一意である必要があります。send-id と recv-id は 0 ～ 255 の範囲から選択する必要があるため、TCP-AO キーチェーンに含められるのは最大 256 個のキーです。
- アプリケーション接続に関連付けられるキーチェーンは 1 つだけです。ロールオーバーは、常にこのキーチェーンのキー内で実行されます。
- 使用中のキーが期限切れになった場合は、有効なライフタイムを持つ新しいキーがそれぞれの側で設定されます。キーがロールオーバーするまで、セグメントの損失が予想されます。
- TCP-AO キーチェーン キーをアクティブと見なすには、send-id、recv-id、key-string、send-lifetime、および cryptographic-algorithm のすべての設定を行う必要があります。

- キー チェーン ソフトウェア プロセスでは、送信ライフタイム構成に基づいて最新のキーが使用されます。または、同じキー キーチェーンの 2 つの異なるキーに同じ send-lifetime が設定されている場合は、最後に設定されたキーを選択します。同じ送信ライフタイムを持つ 2 つのキーを設定することは、ベスト プラクティスではなく、推奨されません。
- ユーザーは、重複する 2 つのキー間の重複時間を 15 分以上に設定する必要があります。
- key-string、send-id、recv-id、cryptographic-algorithm、send-lifetime など使用中のキーの設定を変更すると、TCP 接続フラップが発生します。
- キーチェーンの設定タイプは、クライアントプロトコル内でリンクされているタイプと一致している必要があります。これらのタイプの不一致がある状態で試行されると、ユーザーに通知するための syslog メッセージが生成されます。たとえば、keychain_abc という名前のキーチェーンが Macsec キーチェーンとして設定されていても、BGP で TCP キーチェーンとして関連付けられている場合はサポートされません。同様に、キーチェーンが最初にクライアントに関連付けられ（前方参照と呼ばれるプロセス）、別のキーチェーンタイプとして設定される場合もサポートされません。

TCP キーチェーンおよびキーの設定

始める前に

- キー文字列、送信ライフタイム、暗号化アルゴリズム、およびキーの ID が両方のピアで一致することを確認します。
- ルータの送信 ID がピアルータの受信 ID と一致していることを確認します。個別のキースペースを使用する必要がある場合を除き、両方のパラメータに同じ ID を使用することをお勧めします。
- キーの送信 ID と受信 ID を同じキーチェーン内の別のキーに再利用することはできません。
- AES パスワード暗号化機能が有効になっており、プライマリ キーが構成されている場合、キーストリングは暗号化され、タイプ 6 形式で保存されます。それ以外の場合、パスワードはタイプ 7 暗号化形式で保存されます。
- 詳細については、「プライマリ キーの設定および AES パスワード暗号化機能のイネーブル化」を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `switch# configure terminal`	グローバル構成モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**key chain** *name* **tcp** 例： `switch(config)# key chain bgp-keys tcp`	指定したキーチェーンのキーチェーン 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	**key** *key-ID* 例： `switch(config-tcpkeychain)# key 13`	指定したキーのキー コンフィギュレーション モードを開始します。*key-ID* 引数は、0 ～ 65535 の整数で指定する必要があります。
ステップ 4	**send-id** *send-ID* 例： `switch(config-tcpkeychain-tcpkey)# send-id 2`	キーの送信 ID を指定します。send-ID は、0 ～ 255 の範囲内で、キーチェーンごとに一意の値である必要があります。
ステップ 5	**recv-id** *recv-ID* 例： `switch(config-tcpkeychain-tcpkey)# recv-id 2`	キーの受信 ID を指定します。recv-ID は、0 ～ 255 の範囲内で、キーチェーンごとに一意の値である必要があります。
ステップ 6	**key-string** *[encryption-type] text-string* 例： `switch(config-tcpkeychain-tcpkey)# key-string 0 AS3cureStr1ng`	そのキーのテキスト ストリングを設定します。text-string 引数は英数字で指定します。特殊文字も使用できます。大文字と小文字は区別されます。 Encryption-type 引数に、次のいずれかの値を指定します。 • 0：入力した text-string 引数は、暗号化されていないテキスト文字列です。これがデフォルトです。 • 6：Cisco NX-OS リリース 10.3(3)F 以降、Cisco Nexus 9000 シリーズ プラットフォーム スイッチでシスコ独自の（タイプ 6 暗号化）方式がサポートされています。 • 7：入力した text-string 引数は、暗号化されています。シスコ固有の暗号方式で暗号化されます。このオプションは、別の Cisco NX-OS デバイス上で実行した **show key chain** コマンドの暗号化出力に基づいて、テキスト文字列を入力する場合に役立ちます。

<table>
<tr><th></th><th>コマンドまたはアクション</th><th>目的</th></tr>
<tr><td></td><td></td><td>key-string コマンドには、text-string での次の特殊文字の使用に関する制限があります。
<table>
<tr><th>特殊文字</th><th>説明（Description）</th><th>注</th></tr>
<tr><td>|</td><td>縦棒またはパイプ</td><td>キー文字列の先頭ではサポートされていません</td></tr>
<tr><td>></td><td>右辺と比較して大きい</td><td>キー文字列の先頭ではサポートされていません</td></tr>
<tr><td>\</td><td>バックスラッシュ</td><td>キー文字列の先頭または末尾ではサポートされていません</td></tr>
<tr><td>（</td><td>左丸かっこ</td><td>キー文字列の先頭ではサポートされていません</td></tr>
<tr><td>'</td><td>アポストロフィ</td><td>キー文字列の先頭ではサポートされていません</td></tr>
<tr><td>"</td><td>引用符</td><td>キー文字列の先頭ではサポートされていません</td></tr>
<tr><td>?</td><td>疑問符</td><td>サポート。ただし、疑問符（?）を入力する前に Ctrl+V を押します。</td></tr>
</table>
コマンドでの特殊文字の使用方法の詳細については、「コマンドラインインターフェイスについて」セクションを参照してください。</td></tr>
<tr><td>ステップ 7</td><td>[no] cryptographic-algorithm {HMAC-SHA-1 | HMAC-SHA-256 | AES-128-CMAC }
例：</td><td>TCP セグメントの MAC の計算に使用するアルゴリズムを指定します。1 つのキー に設定できる暗号化アルゴリズムは 1 つだけです。</td></tr>
</table>

	コマンドまたはアクション	目的
	`switch(config-tcpkeychain-tcpkey)# cryptographic-algorithm HMAC-SHA-1`	
ステップ 8	**send-lifetime [local]** *start-time* **duration** [duration-value | infinite | end-time] 例： `switch(config-tcpkeychain-tcpkey)# send-lifetime local 01:01:01 Jan 01 2023 01:01:01 Jan 10 2023`	キーの送信ライフタイムを設定します。デフォルトでは、デバイスは start-time および end-time 引数を UTC として扱います。**local** キーワードを指定すると、デバイスはこれらの時間を現地時間として扱います。 start-time 引数は、キーがアクティブになる日時です。 送信ライフタイムの終了時は次のいずれかのオプションで指定できます。 • **duration** duration-value：ライフタイムの長さ（秒）。最大値は 2147483646 秒（約 68 年）です。 • **infinite**：キーの送信ライフタイムは期限切れになりません。 • end-time：end-time 引数はキーがアクティブでなくなる日時です。
ステップ 9	（任意） **include-tcp-options** 例： `switch(config-tcpkeychain-tcpkey)# include-tcp-options`	パケットの「MAC」ダイジェストを計算中に TCP ヘッダー（TCP AO オプション以外）の一部の「TCP オプション」全体を含める必要があるかどうかを指定するためのオプションの構成です。

TCP キーチェーンの確認

コマンド	目的
show key chain *[name] [detail]*	デバイスに設定されているキーチェーンを表示します。

```
switch# show key chain
Key-Chain bgp_keys tcp
  Key 2 -- text 7 "070e234f"
    send-id 2
    recv-id 2
    cryptographic-algorithm AES_128_CMAC
    send lifetime UTC (08:17:00 May 29 2023)-(08:21:00 May 29 2023)
    include-tcp-options
  Key 3 -- text 7 "070c2058"
    send-id 3
    recv-id 4
```

```
    cryptographic-algorithm HMAC-SHA-1
    send lifetime UTC (08:20:00 May 29 2023)-(always valid) [active]
    include-tcp-options
  Key 12 -- text ""
    send lifetime UTC (08:20:00 May 29 2023)-(always valid)
```

（注） [active] は、キーが有効でアクティブであることを示します。それ以外の場合、キーは非アクティブです。上記の例では、キー 3 のみがアクティブで使用可能です。

show key chain detail コマンドは、アクティブなキーと非アクティブなキーを明示的に表示します。タイプ 6 暗号化の場合、show key chain detail コマンドを実行すると、タイプ 6 キー文字列が復号化可能かどうかも表示されます。また、クライアントがパケットを認証するために現在使用している最も新しいアクティブな送信キーも表示されます。

```
switch# show key chain detail
Key-Chain bgp_keys tcp
  Key 1 -- text 6 "JDYk9k4kmaciqaH6Eu2+9C0tmCRl9k7JAMYs/fXGbW1lmHP88PAA=="
    Type6 Decryptable: yes
    send-id 1
    recv-id 1
    cryptographic-algorithm HMAC-SHA-1
    send lifetime local (18:15:42 May 15 2023)-(always valid) [active]
    include-tcp-options
    accept-ao-mismatch
  Key 2 -- text 6 "JDYkB+Fs8u3ujRDpFSu4tH6H7iTS45JJA6sKeGsBD0L3HjGDeg9AA=="
    Type6 Decryptable: yes
    send-id 2
    recv-id 2
    cryptographic-algorithm AES_128_CMAC
    send lifetime local (17:10:47 May 15 2023)-(18:15:42 May 15 2023) [inactive]

  youngest active send key: 1
```

TCP キーチェーンの構成例

bgp_keys という名前の TCP キーチェーンを設定する例を示します。各キー テキスト ストリングは暗号化されています。キーのライフタイム設定は重複しています。

```
key chain bgp_keys tcp
  key 1
    send-id 1
    recv-id 1
    key-string 7 070e234f
    send-lifetime 01:00:00 Oct 10 2023 01:00:00 Oct 11 2023
    cryptographic-algorithm AES-128-CMAC
  key 2
    send-id 2
    recv-id 2
    key-string 7 075e731f
    send-lifetime 00:45:00 Oct 11 2023 01:00:00 Oct 12 2023
    cryptographic-algorithm HMAC-SHA-256
    include-tcp-options
```

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。