認証の設定

認証は、選択したセキュリティ プロトコルに応じてログイン/パスワード ダイアログ、チャレンジ/レスポンス、メッセージング サポート、および暗号化などのユーザの識別方法を提供します。認証は、ユーザに対してネットワークとネットワーク サービスへのアクセスを許可する前に、ユーザの識別を行う方法です。

認証の設定の前提条件

認証の実装は、認証、許可、およびアカウンティング（AAA）認証と非認証方式に分かれています。シスコでは、可能であれば AAA セキュリティ サービスを試用して認証を実装することを推奨します。

認証の設定に関する制約事項

- 設定できる AAA 方式リストの数は 250 です。
- 非標準のオプションを使用して RADIUS サーバを設定し、非標準のオプションを使用せずに別の RADIUS サーバを設定すると、非標準のオプションを使用する RADIUS サーバ ホストでは事前定義されたホストが受け入れられません。**acct-port** キーワードを使用してアカウンティング要求と異なる UDP 宛先ポート、および非標準オプションの有無に関係なく **auth-port** キーワードを使用して認証要求の UDP 宛先ポートに同じ RADIUS サーバホスト IP アドレスを設定した場合、RADIUS サーバは非標準オプションを受け入れません。

認証について

認証の名前付き方式リスト

まず認証方式の名前付きリストを定義して AAA 認証を設定し、その名前付きリストを各種インターフェイスに適用します。この方式リストは、認証のタイプと実行順序を定義したものです。定義されたいずれかの認証方式を実行するには、この方式リストを特定のインターフェイスに適用する必要があります。唯一の例外は、デフォルトの方式リスト（「default」という名前）です。デフォルトの方式リストは、明示的に定義された名前付きの方式リストを持つインターフェイスを除くすべてのインターフェイスに自動的に適用されます。定義済みの方式リストは、デフォルトの方式リストに優先します。

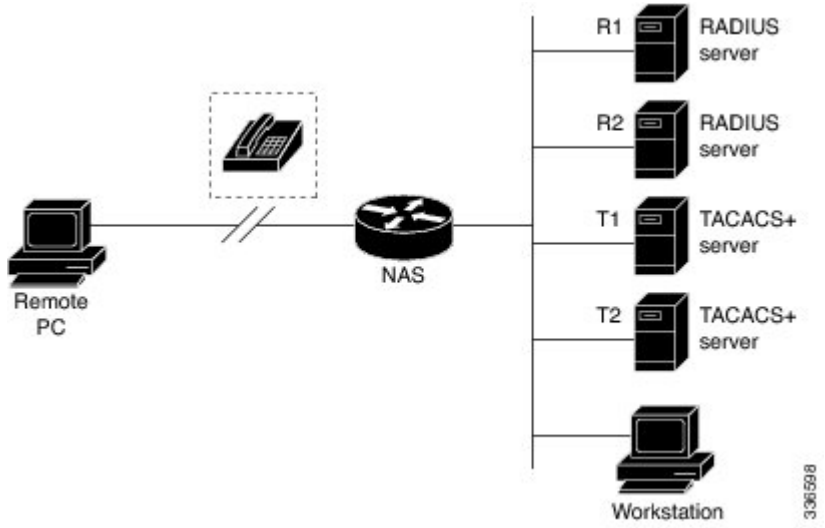
方式リストとは、ユーザ認証のために照会される認証方式を記述したシーケンシャルリストです。方式リストを使用すると、認証に使用するセキュリティプロトコルを1つまたは複数指定できるため、最初の方式が失敗した場合に備えて認証のバックアップ システムを確保できます。シスコ ソフトウェアは、ユーザを認証するため、リストに記載されている最初の方式が使用されます。その方式で応答に失敗した場合、シスコ ソフトウェアは、方式リストに記載されている次の認証方式を選択します。このプロセスは、方式リストのいずれかの認証方式と通信に成功するか、定義されているすべての方式が試行されるまで継続されます。

このソフトウェアでは、前の方式からの応答がない場合にだけ、リストの次の認証方式で認証が試行される、という点に注意してください。このサイクルのいずれかの時点で認証に失敗した場合、つまりセキュリティサーバまたはローカルユーザ名データベースからユーザアクセスを拒否する応答があった場合には、許可プロセスが停止し、それ以上の認証方式は試行されません。

方式リストとサーバ グループ

サーバ グループは、方式リストに使用する既存の RADIUS または TACACS+ サーバ ホストをグループ化する方法の 1 つです。次の図に、4 台のセキュリティ サーバ（R1 と R2 は RADIUS サーバ、T1 と T2 は TACACS+ サーバ）が設置された一般的な AAA ネットワーク設定を示します。R1 と R2 で RADIUS サーバのグループを構成します。T1 と T2 で TACACS+ サーバのグループを構成します。

図 *1 :* 一般的な *AAA* ネットワーク設定



サーバ グループを使用して、設定したサーバ ホストのサブセットを指定し、特定のサービスに使用します。たとえば、サーバ グループを使用すると、R1 および R2 を 1 つのサーバ グループとして定義し、T1 および T2 を別のサーバ グループとして定義できます。また、認証ログインの方式リストに R1 および T1 を指定し、PPP 認証の方式リストに R2 および T2 を指定することもできます。

サーバ グループには、1 台のサーバに対して複数のホスト エントリを含めることができます。エントリごとに固有の識別情報を設定します。固有の識別情報は、IP アドレスと UDP ポート番号の組み合わせで構成されます。これにより、RADIUS ホストとして定義されているさまざまなポートが、固有の AAA サービスを提供できるようになります。つまり、この固有識別情報を使用して、ある IP アドレスに位置する 1 台のサーバ上に複数の UDP ポートが存在する場合、それぞれの UDP ポートに対して RADIUS 要求を送信できます。1 台の RADIUS サーバ上にある異なる 2 つのホスト エントリが 1 つのサービス（認証など）に設定されている場合、設定されている 2 番めのホスト エントリは最初のホスト エントリのフェールオーバー バックアップとして動作します。この例の場合、最初のホスト エントリがアカウンティング サービスの提供に失敗すると、同じデバイスに設定されている 2 番めのホスト エントリを使用してアカウンティング サービスを提供するように、ネットワーク アクセス サーバが試行します（試行される RADIUS ホスト エントリの順番は、設定されている順序に従います）。

サーバグループの設定および着信番号識別サービス（DNIS）番号に基づくサーバグループの設定の詳細については、「RADIUS の設定」または「TACACS+ の設定」を参照してください。

AAA によるログイン認証

イネーブル パスワードによるログイン認証

認証方式としてイネーブルパスワードを指定するには、**enable** キーワード を指定して **aaa authentication login** コマンドを使用します。たとえば、他の方式リストが定義されていない場

合にログイン時のユーザ認証方式としてイネーブルパスワードを指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication login default enable
```

ログイン認証方式としてイネーブル パスワードを使用するには、イネーブル パスワードを定義しておく必要があります。イネーブルパスワードの定義の詳細については、「パスワードおよび権限レベルによるスイッチアクセスの制御」を参照してください。

Kerberos によるログイン認証

Kerberosによる認証は、他のほとんどの認証方式とは異なり、ユーザのパスワードはリモートアクセス サーバに送信されません。ネットワークにログインするリモート ユーザは、ユーザ名の指定を求められます。ユーザのエントリがキー発行局（KDC）に存在する場合は、そのユーザのパスワードを含む暗号化されたチケット認可チケット（TGT）が作成され、デバイスに送信されます。次に、ユーザにパスワードの入力が求められ、デバイスではそのパスワードでTGTの復号が試行されます。復号に成功すると、ユーザは認証され、デバイス上にあるユーザのクレデンシャルキャッシュに TGT が保存されます。

krb5 は KINIT プログラムを使用しますが、デバイスを認証するために、ユーザが KINIT プログラムを実行して TGT を取得する必要はありません。これは、Cisco IOS XE の Kerberos 実装のログイン手順に KINIT が統合されているためです。

ログイン認証方式としてKerberosを指定するには、**krb5** キーワードを指定して **aaa authentication login** コマンドを使用します。たとえば、他の方式リストが定義されていない場合にログイン時のユーザ認証方式として Kerberos を指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication login default krb5
```

ログイン認証方式として Kerberos を使用するには、Kerberos セキュリティ サーバとの通信をイネーブルにしておく必要があります。Kerberosサーバとの通信を確立する方法の詳細については、「Kerberos の設定」の章を参照してください。

ライン パスワードによるログイン認証

ログイン認証方式としてラインパスワードを指定するには、**line** キーワードを指定して **aaa authentication login default** コマンドを使用します。たとえば、他の方式リストが定義されていない場合にログイン時のユーザ認証方式としてラインパスワードを指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication login default line
```

ログイン認証方式としてライン パスワードを使用するには、ライン パスワードを定義しておく必要があります。

ローカル パスワードによるログイン認証

シスコ デバイスが認証にローカルユーザ名データベースを使用するように指定するには **aaa authentication login default** コマンドに **local** キーワードを指定して使用します。たとえば、他

の方式リストが定義されていない場合にログイン時のユーザ認証方式としてローカルユーザ名データベースを指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication login default local
```

group RADIUS によるログイン認証

ログイン認証方式として RADIUS を指定するには、**group radius** を指定して **aaa authentication login default** コマンドを使用します。たとえば、他の方式リストが定義されていない場合にログイン時のユーザ認証方式として RADIUS を指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication login default group radius
```

ログイン認証方式として RADIUS を使用するには、RADIUS セキュリティ サーバとの通信をイネーブルにしておく必要があります。RADIUS サーバとの通信を確立する方法の詳細については、「RADIUS の設定」の章を参照してください。

アクセス要求内の RADIUS 属性 8

aaa authentication login コマンドを使用して RADIUS を指定し、NAS から IP アドレスを要求するようにログインホストを設定すると、グローバル コンフィギュレーション モードで **radius-server attribute 8 include-in-access-req** コマンドを使用して、access-request パケットで属性 8（Framed-IP-Address）を送信できます。このコマンドによって、ユーザ認証の前に、NAS から RADIUS サーバに対してユーザ IP アドレスのヒントを提供できます。

group TACACS によるログイン認証

ログイン認証方式として TACACS+ を指定するには、**group tacacs+** を指定して **aaa authentication login default** コマンドを使用します。たとえば、他の方式リストが定義されていない場合にログイン時のユーザ認証方式として TACACS+ を指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication login default group tacacs+
```

ログイン認証方式として TACACS+ を使用するには、TACACS+ セキュリティ サーバとの通信をイネーブルにしておく必要があります。TACACS+ サーバとの通信を確立する方法の詳細については、「TACACS+ の設定」の章を参照してください。

グループ名によるログイン認証

ログイン認証方式として使用する RADIUS または TACACS+ サーバのサブセットを指定するには、**group** *group-name* 方式を指定して **aaa authentication login default** コマンドを使用します。グループ名とそのグループのメンバを指定して定義するには、**aaa group server** コマンドを使用します。たとえば、**aaa group server** コマンドを使用して、**group loginrad** のメンバを最初に定義します。

```
Device> enable
Device# configure terminal
Device(config)# aaa group server radius loginrad
Device(config-sg-radius)# server 172.16.2.3
Device(config-sg-radius)# server 172.16.2 17
```

```
Device(config-sg-radius)# server 172.16.2.32
Device(config-sg-radius)# end
```

このコマンドにより、172.16.2.3、172.16.2.17、172.16.2.32 の RADIUS サーバがグループ *loginrad* のメンバとして指定されます。

他の方式リストが定義されていない場合にログイン時のユーザ認証方式として **group loginrad** を指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication login default group loginrad
```

ログイン認証方式としてグループ名を使用するには、RADIUS または TACACS+ セキュリティ サーバとの通信をイネーブルにしておく必要があります。RADIUS サーバとの通信を確立する方法の詳細については、「RADIUS の設定」の章を参照してください。TACACS+ サーバとの通信を確立する方法の詳細については、「TACACS+ の設定」の章を参照してください。

AAA による PPP 認証

Kerberos による PPP 認証

PPP を実行するインターフェイスで使用する認証方式として Kerberos を指定するには、**krb5** キーワードを指定して **aaa authentication ppp default Device** コマンドを使用します。たとえば、他の方式リストが定義されていない場合にユーザ認証方式として Kerberos を指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication ppp default krb5
```

PPP 認証方式として Kerberos を使用するには、Kerberos セキュリティ サーバとの通信をイネーブルにしておく必要があります。Kerberos サーバとの通信を確立する方法の詳細については、「Kerberos の設定」の章を参照してください。

（注） Kerberos ログイン認証は、PPP PAP 認証とだけ連携します。

ローカル パスワードによる PPP 認証

シスコ デバイスが認証にローカルユーザ名データベースを使用するように指定するには **aaa authentication ppp default** コマンドに **local** キーワードを指定して使用します。たとえば、他の方式リストが定義されていない場合に、PPP を実行する回線に使用するユーザ認証方式としてローカル ユーザ名データベースを指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication ppp default local
```

group RADIUS による PPP 認証

ログイン認証方式として RADIUS を指定するには、**aaa authentication ppp default group radius** コマンドを使用します。たとえば、他の方式リストが定義されていない場合にログイン時のユーザ認証方式として RADIUS を指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication ppp default group radius
```

PPP 認証方式として RADIUS を使用するには、RADIUS セキュリティ サーバとの通信をイネーブルにしておく必要があります。RADIUS サーバとの通信を確立する方法の詳細については、「RADIUS の設定」の章を参照してください。

アクセス要求内の RADIUS 属性 44

aaa authentication ppp default group radius コマンドを使用して、RADIUS をログイン認証方式として指定すると、グローバル コンフィギュレーション モードで **radius-server attribute 44 include-in-access-req** コマンドを使用して access-request パケットで属性 44（Acct-Session-ID）を送信するようにデバイスを設定できます。このコマンドによって、RADIUS デーモンはコールを開始から終了まで追跡できます。

group TACACS による PPP 認証

ログイン認証方式として TACACS+ を指定するには、**aaa authentication ppp default group tacacs+** コマンドを使用します。たとえば、他の方式リストが定義されていない場合にログイン時のユーザ認証方式として TACACS+ を指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication ppp default group tacacs+
```

PPP 認証方式として TACACS+ を使用するには、TACACS+ セキュリティ サーバとの通信をイネーブルにしておく必要があります。TACACS+ サーバとの通信を確立する方法の詳細については、「TACACS+ の設定」の章を参照してください。

グループ名による PPP 認証

ログイン認証方式として使用する RADIUS または TACACS+ サーバのサブセットを指定するには、**group** *group-name* 方式を指定して **aaa authentication ppp default** コマンドを使用します。グループ名とそのグループのメンバを指定して定義するには、**aaa group server** コマンドを使用します。たとえば、**aaa group server** コマンドを使用して、**group pppradd** のメンバを最初に定義します。

```
Device> enable
Device# configure terminal
Device(config)# aaa group server radius ppprad
Device(config-sg-radius)# server 172.16.2.3
Device(config-sg-radius)# server 172.16.2 17
Device(config-sg-radius)# server 172.16.2.32
Device(config-sg-radius)# end
```

このコマンドにより、172.16.2.3、172.16.2.17、172.16.2.32 の RADIUS サーバがグループ *ppprad* のメンバとして指定されます。

他の方式リストが定義されていない場合にログイン時のユーザ認証方式として **group ppprad** を指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication ppp default group ppprad
```

PPP 認証方式としてグループ名を使用するには、RADIUS または TACACS+ セキュリティ サーバとの通信をイネーブルにしておく必要があります。RADIUS サーバとの通信を確立する方法の詳細については、「RADIUS の設定」の章を参照してください。TACACS+ サーバとの通信を確立する方法の詳細については、「TACACS+ の設定」の章を参照してください。

PPP 要求の AAA スケーラビリティ

ネットワーク アクセス サーバ（NAS）の PPP マネージャによって割り当てられた複数のバックグラウンド プロセスを設定およびモニタして、AAA 認証要求と認可要求に対応できます。AAA スケーラビリティ機能によって、PPP に対する AAA 要求を処理するために使用される複数のプロセスを設定できるようになります。つまり、同時に認証または認可できるユーザ数が増えます。

PPP に対する AAA 要求を処理するために、特定の数のバックグラウンド プロセスを割り当てるには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

```
Device(config)# aaa processes 5000
```

引数 *number* には、PPP に対する AAA 認証要求と認可要求を処理するために確保するバックグラウンド プロセス数を定義します。また、1 ～ 2147483647 の任意の値を設定できます。PPP マネージャが PPP に対する要求を処理する方法のため、この引数には、同時に認証できる新規ユーザの数も定義します。この引数は、いつでも増減できます。

（注） 追加バックグラウンドプロセスの割り当ては、コストが高くなる可能性があります。PPP に対する AAA 要求を処理できるバックグラウンド プロセスの最小数を設定してください。

AAA による ARAP 認証

認可済みゲスト ログインを許可する ARAP 認証

ユーザが EXEC に正常にログイン済みの場合にだけ、ゲストログインを許可するには、**auth-guest** キーワードを指定して **aaa authentication arap default** コマンドを使用します。この方式は ARAP 認証方式リストの先頭に指定する必要がありますが、この方式が成功しなかった場合は引き続き他の方式を試行できます。たとえば、認証のデフォルト方式として、すべての認可済みゲストログイン（つまり、EXEC にログイン済みのユーザによるログイン）を許可し、その方式が失敗した場合にだけ RADIUS を使用するには、次のコマンドを入力します。

```
Device(config)# aaa authentication arap default auth-guest group radius
```

（注） AAA を初期化すると、デフォルトで ARAP によるゲスト ログインはディセーブルになります。ゲストログインを許可するには、**guest** キーワードまたは **auth-guest** キーワードを指定して **aaa authentication arap** {*authentication-list* | **default** コマンドを使用する必要があります。

ゲスト ログインを許可する ARAP 認証

ゲストログインを許可するには、**guest** キーワードを指定して **aaa authentication arap** {**default** | *authentication-list*} コマンドを使用します。この方式は ARAP 認証方式リストの先頭に指定する必要がありますが、この方式が成功しなかった場合は引き続き他の方式を試行できます。たとえば、認証のデフォルト方式としてすべてのゲストログインを許可し、その方式が失敗した場合にだけ RADIUS を使用するには、次のコマンドを入力します。

```
Device(config)# aaa authentication arap default guest group radius
```

ライン パスワードによる ARAP 認証

認証方式としてラインパスワードを指定するには、**line** キーワードを指定して **aaa authentication arap** {**default** | *authentication-list*} コマンドを使用します。たとえば、他の方式リストが定義されていない場合に、ARAPユーザ認証方式としてラインパスワードを指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication arap default line
```

ARAP 認証方式としてライン パスワードを使用するには、ライン パスワードを定義しておく必要があります。

ローカル パスワードによる ARAP 認証

Cisco デバイスが認証にローカルユーザ名データベースを使用するように指定するには **aaa authentication arap** {**default** | *authentication-list*} コマンドに **local** キーワードを指定して使用します。たとえば、他の方式リストが定義されていない場合に、ARAP ユーザ認証方式としてローカル ユーザ名データベースを指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication arap default local
```

group RADIUS による ARAP 認証

ARAP 認証方式として RADIUS を指定するには、**group radius** *method* を指定して **aaa authentication arap** {**default** | *authentication-list*} コマンドを使用します。たとえば、他の方式リストが定義されていない場合にログイン時のユーザ認証方式としてRADIUSを指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication arap default group radius
```

ARAP 認証方式として RADIUS を使用する前に、RADIUS セキュリティ サーバとの通信をイネーブルにしておく必要があります。RADIUS サーバとの通信を確立する方法の詳細については、「RADIUS の設定」の章を参照してください。

group TACACS による ARAP 認証

ARAP 認証方式として TACACS+ を指定するには、**group tacacs+** *method* を指定して **aaa authentication arap** {**default** | *authentication-list*} コマンドを使用します。たとえば、他の方式リストが定義されていない場合にログイン時のユーザ認証方式として TACACS+ を指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication arap default group tacacs+
```

ARAP 認証方式として TACACS+ を使用するには、TACACS+ セキュリティ サーバとの通信をイネーブルにしておく必要があります。TACACS+ サーバとの通信を確立する方法の詳細については、「TACACS+ の設定」の章を参照してください。

グループ名による ARAP 認証

ARAP 認証方式として使用する RADIUS または TACACS+ サーバのサブセットを指定するには、**group** *group-name* 方式を指定して **aaa authentication arap** {**default** | *authentication-list*} コマンドを使用します。グループ名とそのグループのメンバを指定して定義するには、**aaa group server** コマンドを使用します。たとえば、**aaa group server** コマンドを使用して、**group araprad** のメンバを最初に定義します。

```
Device> enable
Device# configure terminal
Device(config)# aaa group server radius araprad
Device(config-sg-radius)# server 172.16.2.3
Device(config-sg-radius)# server 172.16.2 17
Device(config-sg-radius)# server 172.16.2.32
Device(config-sg-radius)# end
```

このコマンドにより、172.16.2.3、172.16.2.17、172.16.2.32 の RADIUS サーバがグループ *araprad* のメンバとして指定されます。

他の方式リストが定義されていない場合にログイン時のユーザ認証方式として **group araprad** を指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication arap default group araprad
```

ARAP 認証方式としてグループ名を使用するには、RADIUS または TACACS+ セキュリティ サーバとの通信をイネーブルにしておく必要があります。RADIUS サーバとの通信を確立する方法の詳細については、「RADIUS の設定」の章を参照してください。TACACS+ サーバとの通信を確立する方法の詳細については、「TACACS+ の設定」の章を参照してください。

AAA による NASI 認証

イネーブル パスワードによる NASI 認証

認証方式としてイネーブルパスワードを指定するには、キーワード **enable** を指定して **aaa authentication nasi** コマンドを使用します。たとえば、他の方式リストが定義されていない場合に、NASI ユーザ認証方式としてイネーブル パスワードを指定するには、次のコマンドを使用します。

```
Devie(config)# aaa authentication nasi default enable
```

認証方式としてイネーブル パスワードを使用するには、イネーブル パスワードを定義しておく必要があります。

group RADIUS による NASI 認証

NASI 認証方式として RADIUS を指定するには、**group radius** 方式を指定して **aaa authentication nasi** コマンドを使用します。たとえば、他の方式リストが定義されていない場合に、NASI ユーザ認証方式として RADIUS を指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication nasi default group radius
```

NASI 認証方式として RADIUS を使用するには、RADIUS セキュリティ サーバとの通信をイネーブルにしておく必要があります。

group TACACS による NASI 認証

NASI 認証方式として TACACS+ を指定するには、**group tacacs+** キーワードを指定して **aaa authentication nasi** コマンドを使用します。たとえば、他の方式リストが定義されていない場合に、NASI ユーザ認証方式として TACACS+ を指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication nasi default group tacacs+
```

認証方式として TACACS+ を使用するには、TACACS+ セキュリティ サーバとの通信をイネーブルにしておく必要があります。

ライン パスワードによる NASI 認証

認証方式としてラインパスワードを指定するには、**line** キーワードを指定して **aaa authentication nasi** コマンドを使用します。たとえば、他の方式リストが定義されていない場合に、NASI ユーザ認証方式としてライン パスワードを指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication nasi default line
```

NASI 認証方式としてライン パスワードを使用するには、ライン パスワードを定義しておく必要があります。

ローカル パスワードによる NASI 認証

シスコ デバイスが認証情報にローカルユーザ名データベースを使用するように指定するには **aaa authentication nasi** コマンドに **local** キーワードを指定して使用します。たとえば、他の方式リストが定義されていない場合に、NASI ユーザ認証方式としてローカル ユーザ名データベースを指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication nasi default local
```

グループ名による NASI 認証

NASI 認証方式として使用する RADIUS または TACACS+ サーバのサブセットを指定するには、**group** *group-name* 方式を指定して **aaa authentication nasi** コマンドを使用します。グループ名とそのグループのメンバを指定して定義するには、**aaa group server** コマンドを使用します。たとえば、**aaa group server** コマンドを使用して、**group nasirad** のメンバを最初に定義します。

```
Device> enable
Device# configure terminal
Device(config)# aaa group server radius nasirad
Device(config-sg-radius)# server 172.16.2.3
Device(config-sg-radius)# server 172.16.2 17
Device(config-sg-radius)# server 172.16.2.32
Device(config-sg-radius)# end
```

このコマンドにより、172.16.2.3、172.16.2.17、172.16.2.32 の RADIUS サーバがグループ *nasirad* のメンバとして指定されます。

他の方式リストが定義されていない場合にログイン時のユーザ認証方式として **group nasirad** を指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication nasi default group nasirad
```

NASI 認証方式としてグループ名を使用するには、RADIUS または TACACS+ セキュリティ サーバとの通信をイネーブルにしておく必要があります。

ログイン入力にかける時間の指定

timeout login response コマンドを使用すると、ログイン入力（ユーザ名やパスワードなど）がタイムアウトするまでの待機時間を指定できます。デフォルトのログイン値は 30 秒です。**timeout login response** コマンドを使用して、1 ～ 300 秒のタイムアウト値を指定できます。30 秒というデフォルトのログインタイムアウト値を変更するには、ライン コンフィギュレーション モードで次のコマンドを使用します。

```
Device(config-line)# timeout login response 30
```

特権レベルでのパスワード保護

ユーザが特権EXECコマンドレベルにアクセスできるかどうかを判断するときに使用する一連の認証方式を作成するには、**aaa authentication enable default** コマンドを使用します。最大 4 つの認証方式を指定できます。追加の認証方式は、その前の方式でエラーが返された場合に限り使用されます。前の方式が失敗した場合は使用されません。すべての方式でエラーが返されても引き続き認証を行うように指定するには、コマンドラインの最後の方式として **none** を指定します。

グローバル コンフィギュレーション モードで次のコマンドを使用します。

```
Device(config)# authentication enable default radius
```

または

```
Device(config)# authentication enable default tacacs
```

パスワード プロンプトに表示するテキストの変更

Cisco IOS XE ソフトウェアからユーザに対してパスワードの入力を求めるときに表示されるデフォルトテキストを変更するには、**aaa authentication password-prompt** コマンドを使用します。このコマンドによって、イネーブル パスワードと、リモート セキュリティ サーバから提供されていないログイン パスワードのパスワード プロンプトが変更されます。このコマンドの **no** 形式を使用すると、パスワードプロンプトが次のデフォルト値に戻ります。

```
Password:
```

aaa authentication password-prompt コマンドでは、リモートのTACACS+ サーバまたはRADIUS サーバから提供されるダイアログは変更されません。

aaa authentication password-prompt コマンドは、RADIUS をログイン方式として使用するときに機能します。RADIUS サーバに到達不能の場合でも、コマンドで定義されたパスワードプロンプトが表示されます。**aaa authentication password-prompt** コマンドは、TACACS+ では機能しません。TACACS+は、NASに対して、ユーザに表示するパスワードプロンプトを提供します。TACACS+ サーバが到達可能な場合、NAS はそのサーバからパスワードプロンプトを受け取り、**aaa authentication password-prompt** コマンドで定義したプロンプトではなく、受け取ったプロンプトを使用します。TACACS+ サーバが到達不能の場合、**aaa authentication password-prompt** コマンドで定義したパスワードプロンプトが使用される可能性があります。

グローバル コンフィギュレーション モードで次のコマンドを使用します。

```
Device(config)# aaa authentication password-prompt “Enter your password now:”
```

PPP セッションの二重認証

PPP セッションを認証できるのは、PAP または CHAP の単一の認証方法を使用した場合だけです。二重認証方式の場合、ネットワーク アクセス権を得るには、リモート ユーザが（CHAP または PAP 認証後に）認証の第 2 段階に合格する必要があります。

この第2段階（「二重」）の認証には、ユーザがパスワードを知っている必要がありますが、ユーザのリモートホストにパスワードは保存されません。そのため、第2段階の認証は、ホストではなくユーザに固有です。その結果、リモート ホストから情報が盗まれた場合でも有効な、追加のセキュリティ レベルが実現します。さらに、ユーザ別にネットワーク特権をカスタマイズできるため、柔軟性も高くなります。

第2段階の認証には、CHAPではサポートされないトークン カードなど、ワンタイム パスワードを使用できます。ワンタイム パスワードを使用している場合、ユーザ パスワードが盗まれても盗用者の役に立ちません。

二重認証の機能

二重認証を使用する場合、2つの認証/認可段階があります。この2つの段階は、リモートユーザがダイヤルインした後、およびPPP セッションが開始された後に発生します。

第1段階では、ユーザがリモートホスト名を使用してログインしてCHAP（またはPAP）がリモートホストを認証し、次にPPPがAAAとネゴシエートしてリモートホストを認可します。このプロセスで、リモート ホストに関連付けられたネットワーク アクセス特権は、そのユーザに関連付けられます。

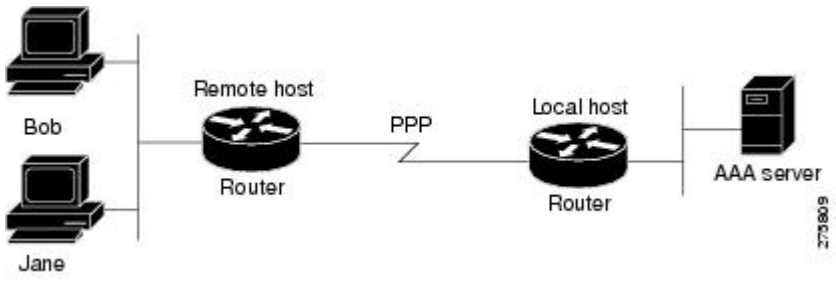
（注） ローカル ホストに対して Telnet 接続だけを許可するように、この第 1 段階ではネットワーク管理者が認可を制限することを推奨します。

第2段階では、リモートユーザが、認証を受けるネットワーク アクセス サーバに対してTelnetを送信する必要があります。リモート ユーザがログインする場合、AAA ログイン認証を使用してユーザを認証する必要があります。次に、AAA を使用して再度許可を受けるために、**access-profile** コマンドを入力する必要があります。この認可が完了すると、ユーザは二重に認証され、ユーザ別のネットワーク特権に従ってネットワークにアクセスできるようになります。

システム管理者は、セキュリティ サーバで適切なパラメータを設定することで、各認証段階の後にリモート ユーザが保持するネットワーク特権を決定します。二重認証を使用するには、**access-profile** コマンドを発行してアクティブ化する必要があります。

注意 複数のホストがネットワーク アクセス サーバに対して PPP 接続を共有する場合、二重認証によって望ましくない状況が発生することがあります（次の図を参照）。まず、ユーザ Bob が PPP セッションを開始し、ネットワーク アクセス サーバで二重認証をアクティブにした場合（次の図を参照）、Bob の PPP セッションが期限切れになるまで、他のすべてのユーザは Bob と同じネットワーク特権を持つことになります。この問題が発生するのは、PPPセッション時に Bob の認可プロファイルがネットワーク アクセス サーバのインターフェイスに適用され、他のユーザからの PPP トラフィックに Bob が確立した PPP セッションが使用されるためです。第 2 に、Bob が PPP セッションを開始して二重認証をアクティブにし、（Bob の PPP セッションが期限切れになる前に）別のユーザ Jane が **access-profile** コマンドを実行する場合（または、Jane がネットワーク アクセス サーバに Telnet を送信し、**autocommand access-profile** が実行された場合）、再度許可が発生し、Jane の許可プロファイルがインターフェイスに適用され、Bob のプロファイルは置換されます。その結果、Bob の PPP トラフィックの不通や中止が発生することや、Bob が本来は持っていないレベルの特権が Bob に付与されることがあります。

図 *2:* 危険性を伴うトポロジ：複数のホストがネットワーク アクセス サーバに対する *PPP* 接続を共有



二重認証後のユーザ プロファイルへのアクセス

二重認証で、リモート ユーザがローカル ホスト名を使用してローカル ホストに対する PPP リンクを確立すると、リモート ホストは CHAP（または PAP）認証されます。CHAP（または PAP）認証後、PPP は AAA とネゴシエートして、リモート ホストに関連付けられたネットワーク アクセス特権をユーザに割り当てます（この段階の特権では、ユーザがローカル ホストに接続するには Telnet 接続を必須にするという制限を付けることを推奨します）。

ユーザが二重認証の第 2 段階を開始する必要があり、ローカル ホストに対して Telnet 接続を確立する場合、ユーザは個人のユーザ名とパスワード（CHAP または PAP のユーザ名とパスワードとは異なります）を入力します。この処理の結果、個人のユーザ名/パスワードに従って AAA 認証が発生します。ただし、ローカル ホストに関連付けられた初期の権限が有効です。ローカルホストに関連付けられた権限は、**access-profile** コマンドを使用して、ユーザプロファイルのユーザ用に定義されている権限で置き換えられるか、結合されます。

二重認証後にユーザ プロファイルにアクセスするには、EXEC コンフィギュレーション モードで次のコマンドを使用します。

```
Device> access-profile merge ignore-sanity-checks
```

autocommand として実行するように **access-profile** コマンドを設定した場合、リモートユーザのログイン後に自動的に実行されます。

CHAP 認証または PAP 認証

ISP のダイヤル ソリューションに使用されている最も一般的なトランスポートプロトコルの 1 つは、ポイントツーポイントプロトコル（PPP）です。従来、リモートユーザはアクセス サーバにダイヤルインして、PPP セッションを開始していました。PPP のネゴシエート後は、リモート ユーザは ISP ネットワークに接続され、そしてインターネットに接続されます。

ISP はアクセス サーバへの接続を顧客に限定したいため、リモート ユーザはアクセス サーバに対して認証を受けてから、PPPセッションを開始する必要があります。通常、リモートユーザは、アクセス サーバからのプロンプトに応じてユーザ名とパスワードを入力して、認証を受けます。これは実行可能なソリューションですが、管理が困難で、リモートユーザにとっても面倒です。

よりよいソリューションは、PPPに組み込まれた認証プロトコルを使用することです。この場合、リモート ユーザはアクセス サーバにダイヤルインし、アクセス サーバと PPP の最小サブセットを開始します。この操作で、ISP のネットワークに対するアクセス権はリモート ユーザに付与されません。単に、アクセス サーバがリモート デバイスと通話できるだけです。

現在、PPP は 2 つの認証プロトコルをサポートします。パスワード認証プロトコル（PAP）およびチャレンジ ハンドシェイク認証プロトコル（CHAP）の 2 つです。いずれも RFC 1334 で規定され、同期インターフェイスと非同期インターフェイスでサポートされます。PAP または CHAP を介する認証は、サーバからのプロンプトを受けてユーザ名とパスワードを入力する方法と同等です。CHAP の場合、接続の間にリモートユーザのパスワードは送信されないため、より安全性が高いと考えられます。

（PAP 認証または CHAP 認証の有無に関係なく）PPP はダイヤルアウト ソリューションでもサポートされます。アクセス サーバがダイヤルアウト機能を使用するのは、アクセス サーバからリモート デバイスに対してコールを開始し、PPP などのトランスポート プロトコルを起動しようとするときです。

（注） CHAP または PAP を使用するには、PPP カプセル化を実行する必要があります。

インターフェイスで CHAP をイネーブルにし、リモート デバイスがそのインターフェイスに接続しようとすると、アクセス サーバからリモート デバイスに CHAP パケットが送信されます。CHAP パケットは、リモート デバイスに応答するように要求または「チャレンジ」します。チャレンジ パケットは、ローカル デバイスの ID、ランダム番号、およびホスト名から構成されます。

リモート デバイスは、チャレンジ パケットを受信すると、ID、リモート デバイスのパスワード、およびランダム番号を連結し、リモートデバイスのパスワードを使用してすべてを暗号化します。リモートデバイスは、その結果を、暗号化プロセスで使用されたパスワードに関連付けられた名前とともにアクセス サーバに返信します。

アクセス サーバがその応答を受信すると、受信した名前を使用して、ユーザ データベースに保存されているパスワードを取得します。取得したパスワードは、暗号化プロセスで使用されたリモート デバイスと同じパスワードです。アクセス サーバは、新しく取得したパスワード

を使用して、連結された情報を暗号化します。その結果が応答パケットで送信された結果と一致する場合、認証は成功です。

CHAP 認証を使用する利点は、リモート デバイスのパスワードがクリア テキストで送信されないことです。結果として、他のデバイスによるパスワード盗用や、ISP のネットワークに対する不正アクセスの取得を回避できます。

CHAP トランザクションが発生するのは、リンクが確立したときだけです。アクセス サーバは、以降のコール中にパスワードを要求しません（ただし、ローカルデバイスは、コール中に他のデバイスからこのような要求があった場合、応答する可能性があります）。

PAP を有効にすると、アクセスサーバに接続しようとするリモートデバイスは、認証要求を送信する必要があります。認証要求に指定されているユーザ名とパスワードが受け入れられた場合、Cisco IOS XE ソフトウェアから認証の確認応答が送信されます。

CHAP または PAP をイネーブルにすると、アクセス サーバは、ダイヤルインするリモート デバイスからの認証を必須にするようになります。イネーブルにしたプロトコルをリモートデバイスがサポートしていない場合、コールはドロップされます。

CHAP または PAP を使用するには、次のタスクを実行する必要があります。

- PPP カプセル化をイネーブルにします。
- インターフェイスで CHAP または PAP をイネーブルにします。
- CHAP の場合、認証が必須の各リモート システムについて、ホスト名の認証および秘密（パスワード）を設定します。

PPP カプセル化の有効化

PPP カプセル化をイネーブルにするには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

```
Device(config-if)# encapsulation ppp
```

このコマンドはインターフェイスで PPP を有効にします。

PAP または CHAP のイネーブル化

PPP カプセル化として設定されているインターフェイスで、CHAP 認証または PAP 認証をイネーブルにするには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

```
Device(config-if)# ppp authentication chap pap
```

サポートされる認証プロトコルと、使用順序を定義します。このコマンドの *protocol1* と protocol2 は、CHAP、MS-CHAP、および PAP のプロトコルを示します。PPP 認証は、まず *protocol1* に指定された最初の認証方式を使用して試行されます。認証に *protocol1* を使用できない場合は、次に設定されているプロトコルを使用して認証のネゴシエーションを行います。

インターフェイスで **ppp authentication chap** を設定する場合、そのインターフェイスで PPP 接続を開始するすべての受信コールは、CHAP を使用して認証される必要があります。同様に、**ppp authentication pap** を設定する場合、PPP 接続を開始するすべての受信コールは、PAP を使

用して認証される必要があります。**ppp authentication chap pap** を設定する場合、アクセスサーバは、CHAP を使用して PPP セッションを開始するすべての受信コールを認証しようとします。リモート デバイスが CHAP をサポートしない場合、アクセス サーバは PAP を使用してコールを認証しようとします。リモート デバイスが CHAP も PAP もサポートしない場合、認証は失敗し、コールはドロップされます。**ppp authentication pap chap** を設定する場合、アクセスサーバは、PAP を使用して PPP セッションを開始するすべての受信コールを認証しようとします。リモート デバイスが PAP をサポートしない場合、アクセス サーバは CHAP を使用してコールを認証しようとします。リモートデバイスがいずれのプロトコルもサポートしない場合、認証は失敗し、コールはドロップされます。**callin** キーワードを指定して **ppp authentication** コマンドを設定すると、アクセスサーバは、リモートデバイスがコールを開始した場合にだけ、リモートデバイスの認証を行います。

認証方式リストと **one-time** キーワードを使用できるのは、AAA を有効にした場合だけです。TACACS または拡張 TACACS を有効にしている場合は、使用できません。**ppp authentication** コマンドを使用して認証方式リストの名前を指定すると、PPP は、指定した方式リストに定義されている方式を使用して、接続を認証しようとします。AAA をイネーブルにし、名前で定義されている方式リストがない場合、PPP は、デフォルトに定義されている方式を使用して接続を認証しようとします。**one-time** キーワードを指定して **ppp authentication** コマンドを使用すると、認証中にワンタイムパスワードをサポートできます。

if-needed キーワードを使用できるのは、TACACS または拡張 TACACS を使用している場合だけです。**if-needed** キーワードを指定して **ppp authentication** コマンドを使用することは、現在のコール期間中にリモートデバイスがまだ認証されていない場合にだけ、PPP が PAP または CHAP を介してリモートデバイスを認証することを示します。リモートデバイスが、標準のログイン手順で認証を受け、EXEC プロンプトから PPP を開始した場合、**ppp authentication chap if-needed** がインターフェイスで設定されていれば、PPP は CHAP を介して認証しません。

⚠

注意 **aaa authentication ppp** コマンドを使用して設定されていない *list-name* を使用する場合、その回線での PPP は無効になります。

着信認証と発信認証

PPP は双方向の認証をサポートしています。通常、リモート デバイスがアクセス サーバにダイヤルインするときは、それが許可されているアクセスであることをリモートデバイスが証明するように、アクセス サーバから要求されます。これは着信認証と呼ばれます。同時に、リモート デバイスは、身元を証明するようにアクセス サーバに要求することもできます。これは発信認証と呼ばれます。また、アクセス サーバは、リモート デバイスに対してコールを開始するときにも、発信認証を実行します。

発信 PAP 認証のイネーブル化

発信 PAP 認証をイネーブルにするには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

```
Device(config-if)# ppp pap sent-username username1 password password1
```

アクセスサーバからリモートデバイスに対してコールを開始する場合は常に、またはアウトバウンド認証のためにリモートデバイスの要求に応答する必要がある場合は、**ppp pap sent-username** コマンドで指定されたユーザ名とパスワードを使用して自身を認証します。

PAP 認証要求の拒否

ピアからの PAP 認証要求を拒否するには（つまり、すべてのコールで PAP 認証をディセーブルにするには）、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

```
Device(config-if)# ppp pap refuse
```

refuse キーワードが使用されない場合、デバイスはピアから受信した PAP 認証チャレンジを拒否しません。

共通 CHAP パスワードの作成

リモート CHAP 認証の場合、不明なピアからのチャレンジに応じて使用する共通の CHAP シークレットパスワードを作成するようにデバイスを設定できます。たとえば、新しい（つまり、不明な）デバイスが追加されたデバイス（別のベンダーの、または古いバージョンの Cisco IOS XE ソフトウェアを実行しているデバイス）のロータリーを呼び出します。**ppp chap password** コマンドを使用すると、任意のダイヤラインターフェイスまたは非同期グループインターフェイスで、複数のユーザ名およびパスワード コンフィギュレーション コマンドをこのコマンドの単一のコピーで置換できます。

デバイスのコレクションに発信するデバイスが、共通の CHAP シークレットパスワードを設定できるようにするには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

```
Device(config-if)# ppp chap password secret
```

CHAP 認証要求の拒否

ピアからの CHAP 認証要求を拒否するには（つまり、すべてのコールで CHAP 認証をディセーブルにするには）、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

```
Device(config-if)# ppp chap refuse calling
```

calling キーワードが使用されると、デバイスは、ピアから受信した CHAP 認証チャレンジへの応答を拒否します。ただし、デバイスが送信する CHAP チャレンジに対しては、ピアが応答することを必須とします。

（**ppp pap sent-username** コマンドを使用して）アウトバウンド PAP が有効になっている場合、拒否パケットの認証方式として、PAP が使用されます。

ピアが認証されるまで CHAP 認証を遅延する

CHAP 認証を要求するピアがデバイスから認証を受けるまで、デバイスがこのピアを認証しないように指定するには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

```
Device(config-if)# ppp chap wait secret
```

このコマンド（デフォルト）により、CHAP認証を要求するピアがデバイスから認証を受けるまで、デバイスがこのピアを認証しないように指定します。**no ppp chap wait**コマンドにより、デバイスが認証チャレンジに対して即時に応答するように指定されます。

MS-CHAP の使用

マイクロソフト チャレンジ ハンドシェイク認証プロトコル（MS-CHAP）は、Microsoft バージョンの CHAP であり、RFC 1994 の拡張です。標準バージョンの CHAP と同様に、MS-CHAP は PPP 認証に使用されます。この場合、Microsoft Windows NT または Microsoft Windows 95 を使用する PC と、ネットワーク アクセス サーバとして動作する Cisco デバイスまたはアクセスサーバとの間に認証が発生します。

MS-CHAP と標準の CHAP の違いは次のとおりです。

- MS-CHAP をイネーブルにするには、LCP オプション 3 の Authentication Protocol で、CHAP Algorithm 0x80 をネゴシエートします。
- MS-CHAP 応答パケットは、Microsoft Windows NT 3.5 および 3.51、Microsoft Windows 95、および Microsoft LAN Manager 2.*x* と互換性を持つように設計されたフォーマットです。このフォーマットを使用する場合、オーセンティケータは、クリアパスワードまたは可逆的に暗号化されたパスワードを保存する必要はありません。
- MS-CHAP には、オーセンティケータが制御する認証リトライ メカニズムがあります。
- MS-CHAP には、オーセンティケータが制御するチャレンジパスワード メカニズムがあります。
- MS-CHAP には、Failure パケット メッセージ フィールドで返される「reason-for failure」コード セットが定義されています。

実装したセキュリティ プロトコルに応じて、AAA セキュリティ サービスの有無にかかわらず、MS-CHAP による PPP 認証を使用できます。AAA をイネーブルにしている場合、MS-CHAP を使用する PPP 認証は、TACACS+ および RADIUS の両方と併用できます。次の表に、RADIUS が MS-CHAP をサポートできるベンダー固有 RADIUS 属性（IETF Attribute 26）を示します。

表 *1: MS-CHAP* 用のベンダー固有 *RADIUS* 属性

ベンダー ID 番号	ベンダータイプ 番号	ベンダー固有属性	説明
311	11	MSCHAP-Challenge	ネットワーク アクセス サーバが MS-CHAP ユーザに送信するチャレンジが含まれます。これは、Access-Request パケットと Access-Challenge パケットの両方で使用できます。

ベンダー ID 番号	ベンダータイプ 番号	ベンダー固有属性	説明
211	11	MSCHAP-Response	PPP MS-CHAP ユーザがチャレンジに対する応答で提供するレスポンス値が含まれます。Access-Request パケットでしか使用されません。この属性は、PPP CHAP ID と同じです

ドメイン ストリッピング

AAA ブロードキャスト アカウンティング機能を有効にすると、アカウンティング情報を複数の AAA サーバに同時に送信できます。つまり、アカウンティング情報を 1 つまた複数の AAA サーバに同時にブロードキャストすることが可能です。この機能を使用すると、プライベートおよびパブリック AAA サーバにアカウント情報を送信できます。この機能では、音声アプリケーションによる課金情報も提供されます。

ドメイン ストリッピング機能を使用すると、ドメイン ストリッピングをサーバ グループ レベルで設定できます。

サーバ単位のグループ コンフィギュレーションはグローバル コンフィギュレーションを上書きします。ドメイン ストリッピングが、グローバルではイネーブルではないがサーバ グループでイネーブルになっている場合、そのサーバ グループに対してのみイネーブルになります。また、Virtual Routing and Forwarding（VRF）固有のドメイン ストリッピングがグローバルで設定されていて、別の VRF のドメイン ストリッピングがサーバ グループで設定されている場合、ドメイン ストリッピングは両方の VRF でイネーブルになります。VRF の設定は、サーバ グループ コンフィギュレーション モードから取得されます。サーバグループ コンフィギュレーションがグローバル コンフィギュレーション モードでディセーブルになっているが、サーバ グループ コンフィギュレーション モードで使用可能である場合、サーバグループ コンフィギュレーション モードでのすべての設定が適用可能です。

ドメイン ストリッピングおよびブロードキャスト アカウンティングを設定した後で、設定ごとに別個のアカウンティング レコードを作成できます。

domain-stripping コマンドと **directed-request** コマンドの両方が有効になっている場合、ドメインストリッピングが優先され、ダイレクトリクエスト機能は動作しません。

認証の設定方法

AAA を使用したログイン認証の設定

AAA セキュリティ サービスにより、さまざまなログイン認証方式を容易に実行できるようになります。**aaa authentication login** コマンドを使用すると、サポートされているログイン認証方式のいずれを使用するかに関係なく、AAA 認証が有効になります。 **aaa authentication login**

コマンドを使用すると、ログイン時に試行する認証方式リストを 1 つまたは複数作成できます。これらのリストは、**login authentication** ライン コンフィギュレーション コマンドによって適用されます。

AAA を使用してログイン認証を設定するには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

手順

	コマンドまたはアクション	目的
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	**aaa new-model** 例： `Device(config)# aaa new-model`	AAA をイネーブルにします。
ステップ 4	**aaa authentication login** {**default** | *list-name*} *method1*[*method2...*] 例： `Device(config)# aaa authentication login default local`	ローカルな認証リストを作成します。
ステップ 5	**line** [**aux** | **console** | **tty** | **vty**] **line-number** [**ending-line-number**] 例： `Device(config)# line vty 1`	認証リストを適用する回線について、ライン コンフィギュレーション モードを開始します。
ステップ 6	**login authentication** {**default** | *list-name*} 例： `Device(config-line)# login authentication default`	1 つの回線または複数回線に認証リストを適用します。

	コマンドまたはアクション	目的
ステップ 7	**end** 例： `Device(config-line)# end`	回線コンフィギュレーション モードを終了します。続いて、特権 EXEC モードに戻ります。

次のタスク

list-name は、作成するリストを指定するときに使用される名前で、文字列を使用できます。method 引数は、認証アルゴリズムが試行する実際の方式を指します。追加の認証方式は、その前の方式でエラーが返された場合に限り使用されます。前の方式が失敗した場合は使用されません。すべての方式でエラーが返されても引き続き認証を行うように指定するには、コマンドラインの最後の方式として **none** を指定します。

たとえば、（この例では）TACACS+ サーバでエラーが返されても引き続き認証を行うように指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication login default group tacacs+ none
```

（注） **none** キーワードを指定すると、すべてのユーザがログイン認証に成功するため、認証のバックアップ方式としてだけ使用してください。

login authentication コマンドにリストが指定されていない場合に使用するデフォルトのリストを作成するには、**default** キーワードの後ろにデフォルト状況で使用される方式を指定します。デフォルト認証方式リストは、自動的にすべてのインターフェイスに適用されます。

たとえば、ログイン時のユーザ認証のデフォルト方式として RADIUS を指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication login default group radius
```

AAA を使用した PPP 認証の設定

AAA セキュリティ サービスにより、PPP を実行するシリアル インターフェイスに使用できるさまざまな認証方式の実行が容易になります。**aaa authentication ppp** コマンドを使用すると、サポートされている PPP 認証方式のいずれを使用するかに関係なく、AAA 認証が有効になります。

PPP を使用してシリアル回線に AAA 認証方式を設定するには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

手順

	コマンドまたはアクション	目的
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	**aaa new-model** 例： `Device(config)# aaa new-model`	AAA をイネーブルにします。
ステップ 4	**aaa authentication ppp** {**default** | *list-name*} *method1*[*method2...*] 例： `Device(config)# aaa authentication ppp-auth default local`	ローカルな認証リストを作成します。
ステップ 5	**interface** *interface-type interface-number* 例： `Device(config)# interface gigabitethernet 0/1/0`	認証リストを適用するインターフェイスについて、インターフェイス コンフィギュレーション モードを開始します。
ステップ 6	**ppp authentication** {*protocol1* [*protocol2...*]} [**if-needed**] {**default** | *list-name*} [**callin**] [**one-time**][**optional**] 例： `Device(config)# ppp authentication ms-chap ppp-auth`	1 つの回線または複数回線に認証リストを適用します。このコマンドの *protocol1* と *protocol2* は、CHAP、MS-CHAP、および PAP のプロトコルを示します。PPP 認証は、まず *protocol1* に指定された最初の認証方式を使用して試行されます。認証に *protocol1* を使用できない場合は、次に設定されているプロトコルを使用して認証のネゴシエーションを行います。

	コマンドまたはアクション	目的
ステップ 7	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

次のタスク

aaa authentication ppp コマンドを使用して、PPP を介して認証を試行するときに使用する認証方式のリストを 1 つまたは複数作成します。これらのリストは、**ppp authentication** ライン コンフィギュレーション コマンドによって適用されます。

名前付きリストが **ppp authentication** コマンドに指定されていない場合に使用するデフォルトのリストを作成するには、**default** キーワードの後ろにデフォルト状況で使用される方式を指定します。

たとえば、ユーザ認証のデフォルト方式としてローカル ユーザ名データベースを指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication ppp default local
```

list-name は、作成するリストを指定するときに使用される名前で、任意の文字列を使用できます。method 引数は、認証アルゴリズムが試行する実際の方式を指します。追加の認証方式は、その前の方式でエラーが返された場合に限り使用されます。前の方式が失敗した場合は使用されません。すべての方式でエラーが返されても引き続き認証を行うように指定するには、コマンドラインの最後の方式として **none** を指定します。

たとえば、（この例では）TACACS+ サーバでエラーが返されても引き続き認証を行うように指定するには、次のコマンドを入力します。

```
Device(config)# aaa authentication ppp default group tacacs+ none
```

（注） **none** を指定するとすべてのユーザが認証に成功してログインできるようになるため、認証のバックアップ方式として使用する必要があります。

AAA を使用した ARAP 認証の設定

aaa authentication arap コマンドを使用して、AppleTalk Remote Access Protocol（ARAP）ユーザがデバイスにログインを試行するときに使用する認証方式のリストを 1 つまたは複数作成できます。これらのリストは、**arap authentication** ライン コンフィギュレーション コマンドで使用されます。

グローバル コンフィギュレーション モードで次のコマンドを使用します。

手順

	コマンドまたはアクション	目的
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	**aaa new-model** 例： `Device(config)# aaa new-model`	AAA をイネーブルにします。
ステップ 4	**aaa authentication arap** 例： `Device(config)# aaa authentication arap` 例： ARAPユーザに対する認証をイネーブルにします。	
ステップ 5	**line** *number* 例： `Device(config)# line 1`	（任意）ライン コンフィギュレーション モードに変更します。
ステップ 6	Device(config-line)# **autoselect arap** 例： `Device(config-line)# auto-select arap`	（任意）ARAPの自動選択をイネーブルにします。
ステップ 7	**autoselect during-login** 例： `Device(config-line)# autoselect during-login`	（任意）ユーザ ログイン時にARAPセッションを自動的に開始します。

	コマンドまたはアクション	目的
ステップ 8	**arap authentication** *list-name* 例： `Device(config-line)# arap authentication arap-authen`	（任意：**default** が **aaa authentication arap** コマンドに使用されている場合は不要）回線上の ARAP に対する TACACS+ 認証を有効にします。
ステップ 9	**end** 例： `Device(config-line)# end`	ライン コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。

次のタスク

list-name は、作成するリストを指定するときに使用される名前で、任意の文字列を使用できます。*method* 引数は、認証アルゴリズムが試行する方式の実際のリストを指します。試行は入力されている順序で行われます。

名前付きリストが **arap authentication** コマンドに指定されていない場合に使用するデフォルトのリストを作成するには、**default** キーワードの後ろにデフォルト状況で使用される方式を指定します。

追加の認証方式は、その前の方式でエラーが返された場合に限り使用されます。前の方式が失敗した場合は使用されません。すべての方式でエラーが返されても引き続き認証を行うように指定するには、コマンドラインの最後の方式として **none** を指定します。

（注） **none** を指定するとすべてのユーザのログインが認証されるようになるため、認証のバックアップ方式として使用する必要があります。

たとえば、ARAP とともに使用するデフォルトの AAA 認証方式リストを作成するには、次のコマンドを使用します。

```
Device(config)# aaa authentication arap default if-needed none
```

ARAP に同じ認証方式リストを作成し、リストに *MIS-access* と名前を付けるには、次のコマンドを入力します。

```
Device(config)# aaa authentication arap MIS-access if-needed none
```

AAA を使用した NASI 認証の設定

aaa authentication nasi コマンドを使用して、NetWare Asynchronous Services Interface（NASI）ユーザがデバイスにログインを試行するときに使用する認証方式のリストを 1 つまたは複数作

成できます。これらのリストは、**nasi authentication line** コンフィギュレーション コマンドで使用されます。

AAA を使用して NASI 認証を設定するには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

手順

	コマンドまたはアクション	目的
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	**aaa new-model** 例： `Device(config)# aaa new-model`	AAA をグローバルに有効にします。
ステップ 4	**aaa authentication nasi** 例： `Device(config)# aaa authentication nasi`	NASI ユーザに対する認証をイネーブルにします。
ステップ 5	**line** *number* 例： `Device(config)# line 4`	（任意：**aaa authentication nasi** コマンドで **default** が使用されている場合は不要。）ライン コンフィギュレーション モードを開始します。
ステップ 6	**nasi authentication** *list-name* 例： `Device(config-line)# nasi authentication nasi-authen`	（任意：**aaa authentication nasi** コマンドで **default** が使用されている場合は不要。）回線でNASIの認証を有効にします。

	コマンドまたはアクション	目的
ステップ 7	**end** 例： `Device(config-line)# end`	ライン コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。

次のタスク

list-name は、作成するリストを指定するときに使用される名前で、任意の文字列を使用できます。*method* 引数は、認証アルゴリズムが試行する方式の実際のリストを指します。試行は入力されている順序で行われます。

aaa authentication nasi コマンドに名前付きリストが指定されていない場合に使用するデフォルトのリストを作成するには、**default** キーワードの後ろにデフォルト状況で使用される方式を指定します。

追加の認証方式は、その前の方式でエラーが返された場合に限り使用されます。前の方式が失敗した場合は使用されません。すべての方式でエラーが返されても引き続き認証を行うように指定するには、コマンドラインの最後の方式として **none** を指定します。

（注） **none** を指定するとすべてのユーザのログインが認証されるようになるため、認証のバックアップ方式として使用する必要があります。

ユーザ名が空のアクセス要求が RADIUS サーバに送信されないようにする

次の設定手順では、ユーザ名が空のアクセス要求が RADIUS サーバに送信されないようにする方法について説明します。この機能により、RADIUS サーバとの不要なやりとりを回避でき、RADIUS ログの量を少なくすることができます。

手順

	コマンドまたはアクション	目的
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。

	コマンドまたはアクション	目的
ステップ 3	**aaa new-model** 例： `Device(config)# configure terminal`	AAA をグローバルに有効にします。
ステップ 4	**aaa authentication suppress null-username** 例： `Device(config)# aaa authentication suppress null-username`	ユーザ名が空のアクセス要求が RADIUS サーバに送信されないようにします。
ステップ 5	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

AAA 認証のメッセージ バナーの設定

AAA は、設定可能でパーソナライズされたログインおよび failed-login バナーの使用をサポートします。ユーザが AAA を使用して認証を受けるシステムにログインする場合、および何らかの理由で認証が失敗した場合に表示されるメッセージ バナーを設定できます。

ログイン バナーの設定

ユーザがログインするときに表示されるメッセージを設定する（デフォルトのログイン メッセージを置き換える）には、次のタスクを実行します。

始める前に

ログインバナーを作成するには、デリミタを設定する必要があります。設定することで、続くテキスト文字列をバナーとして表示する必要があることがシステムに通知されます。次に、テキスト文字列自体を設定する必要があります。デリミタは、バナーの末尾を示すために、テキスト ストリングの末尾で繰り返されます。デリミタには、拡張 ASCII 文字セットの任意の 1 文字を使用できます。ただし、デリミタとして定義した文字は、バナー用のテキスト文字列には使用できません。

手順

	コマンドまたはアクション	目的
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）。

	コマンドまたはアクション	目的
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	**aaa new-model** 例： `Device(config)# aaa new-model`	AAA をイネーブルにします。
ステップ 4	**aaa authentication banner** *delimiter string delimiter* 例： `Device(config)# aaa authentication banner *Unauthorized use is prohibited.*`	パーソナライズされたログイン バナーを作成します。
ステップ 5	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

Failed-Login バナーの設定

ユーザ ログインが失敗したときに表示されるメッセージを設定する（デフォルトの failed-login メッセージを置き換える）には、次のタスクを実行します。

始める前に

failed-loginバナーを作成するには、デリミタを設定する必要があります。設定することで、続くテキスト文字列をバナーとして表示する必要があることがシステムに通知されます。次に、テキスト文字列自体を設定する必要があります。デリミタは、failed-loginバナーの末尾を示すために、テキスト ストリングの末尾で繰り返されます。デリミタには、拡張 ASCII 文字セットの任意の1文字を使用できます。ただし、デリミタとして定義した文字は、バナーを構成するテキスト ストリングには使用できません。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例：	特権 EXEC モードを有効にします。

	コマンドまたはアクション	目的
	Device> enable	• 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	**aaa new-model** 例： Device(config)# **aaa new-model**	AAA をイネーブルにします。
ステップ 4	**aaa authentication fail-message** *delimiter string delimiter* 例： Device(config)# **aaa authentication fail-message *Failed login. Try again.***	ユーザ ログインが失敗したときに表示されるメッセージを作成します。
ステップ 5	**end** 例： Device(config)# **end**	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

AAA パケット オブ ディスコネクトの設定

特定のセッション属性が指定された場合、パケット オブ ディスコネクト（POD）によってネットワーク アクセス サーバ（NAS）の接続が終了されます。UNIX ワークステーション上にある POD クライアントでは、AAA から取得したセッション情報を使用して、ネットワーク アクセス サーバで実行されている POD サーバに接続解除パケットを送信します。NAS では、1 つまたは複数の一致するキー属性を含む任意の着信ユーザ セッションを終了します。必要なフィールドがない場合、または完全一致が見つからない場合、要求は拒否されます。

POD を設定するには、グローバル コンフィギュレーション モードで次のタスクを実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例：	特権 EXEC モードを有効にします。

	コマンドまたはアクション	目的
	`Device> enable`	• 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	**aaa accounting network default start-stop radius** 例： `Device(config)# aaa accounting network default start-stop radius`	AAA アカウンティング レコードをイネーブルにします。
ステップ 4	**aaa accounting delay-start** 例： `Device(config)# aaa accounting delay-start`	（任意）POD パケットで使用できるように、Framed-IP-Address が割り当てられるまで、開始アカウンティング レコードの生成を遅延します。
ステップ 5	**aaa pod server server-key** *string* 例： `Device(config)# aaa pod server server-key xyz123`	POD の受信イネーブルにします。
ステップ 6	**radius server** *name* non-standard 例： `Device(config)# radius server radser`	RADIUS サーバを設定し、RADIUS サーバ コンフィギュレーション モードを開始します。
ステップ 7	**address** {**ipv4** | **ipv6**} *hostname* 例： `Device(config-radius-server)# address ipv4 radius-host`	RADIUS ホストを設定します。
ステップ 8	**end** 例： `Device(config-radius-server)# end`	RADIUS サーバ コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

二重認証の設定

二重認証を設定するには、次の手順を実行します。

1. **aaa-new model** グローバル コンフィギュレーション コマンドを使用して、AAA を有効にします。
2. **aaa authentication** コマンドを使用して、ログインおよびPPP認証方式リストを使用するようにネットワークアクセスサーバを設定します。次に、これらの方式リストを適切な回線やインターフェイスに適用します。
3. **aaa authorization** コマンドを使用して、ログイン時の AAA ネットワーク許可を設定します。
4. セキュリティ プロトコル パラメータ（たとえば、RADIUS または TACACS+）を設定します。
5. セキュリティ サーバで、ユーザがローカル ホストに接続できるアクセス コントロール リストの AV ペアを使用するには、Telnet 接続を確立する必要があります。
6. （任意）autocommand として **access-profile** コマンドを設定します。autocommand を設定すると、リモートユーザは、個人のユーザプロファイルに関連付けられた許可済み権限にアクセスするために、手動で **access-profile** コマンドを入力する必要はなくなります。

（注） **access-profile** コマンドが autocommand として設定されている場合でも、二重認証を完了するには、ユーザがローカルホストに Telnet を送信し、ログインする必要があります。

ユーザ固有の許可ステートメントを作成する場合、次の規則に従います（これらの規則は、**access-profile** コマンドのデフォルトの動作に関連します）。

- セキュリティ サーバでアクセス コントロール リストの AV ペアを設定する場合、有効な AV ペアを使用します。
- リモートユーザがインターフェイスの既存の許可（第 2 段階の認証/許可の前に存在する許可）を使用し、異なるアクセスコントロールリスト（ACL）を持つようにするには、ユーザ固有の許可定義で ACL AV ペアだけを指定します。この方法は、デフォルトの認可プロファイルを設定してリモート ホストに適用し、ACL はユーザ別に適用する場合などに有効です。
- これらのユーザ固有の許可ステートメントを後でインターフェイスに適用すると、ユーザの許可に使用する **access-profile** コマンドの実行形式によって、既存のインターフェイス設定に追加することや、既存のインターフェイス設定を置き換えることができます。許可ステートメントを設定する前に、**access-profile** コマンドの機能について理解する必要があります。
- ISDN または Multilink PPP を使用する予定がある場合、ローカル ホストで仮想テンプレートも設定する必要があります。

自動二重認証のイネーブル化

自動二重認証を実装することで、ユーザにとって二重認証プロセスが容易になります。自動二重認証は、二重認証が持つセキュリティ上の利点をすべて備えていますが、リモートユーザにとってよりシンプルでユーザフレンドリなインターフェイスです。二重認証の場合、ユーザ認証の第 2 レベルは、ユーザがネットワーク アクセス サーバまたはルータに Telnet に送信し、ユーザ名とパスワードを入力したときに完了します。自動二重認証の場合、ユーザがネットワーク アクセス サーバに Telnet を送信する必要はありません。その代わり、ユーザ名とパスワードまたは Personal Identification Number（PIN）の入力を求めるダイアログボックスが表示されます。自動二重認証機能を使用するには、対応するクライアント アプリケーションがリモート ユーザ ホストで実行されている必要があります。

（注） 自動二重認証は、既存の二重認証機能と同様に、Multilink PPP ISDN 接続専用です。自動二重認証は、X.25 や SLIP など他のプロトコルとは併用できません。

自動二重認証は、既存の二重認証機能の強化です。自動二重認証を設定するには、まず次の手順を実行して二重認証を設定する必要があります。

1. **aaa-new model** グローバル コンフィギュレーション コマンドを使用して、AAA を有効にします。
2. **aaa authentication** コマンドを使用して、ログインおよびPPP認証方式リストを使用するようにネットワークアクセスサーバを設定します。次に、これらの方式リストを適切な回線やインターフェイスに適用します。
3. **aaa authorization** コマンドを使用して、ログイン時の AAA ネットワーク許可を設定します。
4. セキュリティ プロトコル パラメータ（たとえば、RADIUS または TACACS+）を設定します。
5. セキュリティ サーバで、ユーザがローカル ホストに接続できるアクセス コントロール リストの AV ペアを使用するには、Telnet 接続を確立する必要があります。
6. autocommand として **access-profile** コマンドを設定します。autocommand を設定すると、リモートユーザは、個人のユーザプロファイルに関連付けられた許可済み権限にアクセスするために、手動で **access-profile** コマンドを入力する必要はなくなります。

（注） **access-profile** コマンドが autocommand として設定されている場合でも、二重認証を完了するには、ユーザがローカルホストに Telnet を送信し、ログインする必要があります。

ユーザ固有の許可ステートメントを作成する場合、次の規則に従います（これらの規則は、**access-profile** コマンドのデフォルトの動作に関連します）。

- セキュリティ サーバでアクセス コントロール リストの AV ペアを設定する場合、有効な AV ペアを使用します。
- リモート ユーザがインターフェイスの既存の認可（第 2 段階の認証/認可の前に存在する認可）を使用し、異なるアクセス コントロール リスト（ACL）を持つようにするには、ユーザ固有の認可定義で ACL AV ペアだけを指定します。この方法は、デフォルトの認可プロファイルを設定してリモート ホストに適用し、ACL はユーザ別に適用する場合などに有効です。
- これらのユーザ固有の許可ステートメントを後でインターフェイスに適用すると、ユーザの許可に使用する **access-profile** コマンドの実行方法によって、既存のインターフェイス設定に追加することや、既存のインターフェイス設定を置き換えることができます。許可ステートメントを設定する前に、**access-profile** コマンドの機能について理解する必要があります。
- ISDN または Multilink PPP を使用する予定がある場合、ローカル ホストで仮想テンプレートも設定する必要があります。

自動二重認証の設定

自動二重認証を設定するには、次のタスクを実行します。

手順

	コマンドまたはアクション	目的
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	**ip trigger-authentication** [**timeout** *seconds*] [**port** *number*] **例**： `Device(config)# ip trigger-authentication timeout 120`	二重認証の自動化をイネーブルにします。
ステップ 4	**interface** *type number* **例**： `Device(config)# interface`	インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	gigabitethernet 1/0/17	
ステップ 5	**ip trigger-authentication** 例： `Device(config-if)# ip rigger-authentication`	自動二重認証をインターフェイスに適用します。
ステップ 6	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

自動二重認証のトラブルシューティング

自動二重認証の問題を解決するには、特権 EXEC モードで次のコマンドを使用します。

手順

	コマンドまたはアクション	目的
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	**show ip trigger-authentication** 例： `Device# show ip trigger-authentication`	自動二重認証が試行され、成功または失敗したリモート ホストのリストが表示されます。
ステップ 3	**clear ip trigger-authentication** 例： `Device# clear ip trigger-authentication`	自動二重認証が試行されたリモート ホストのリストをクリアします（これは、**show ip trigger-authentication** コマンドで表示されるテーブルをクリアします）。
ステップ 4	**debug ip trigger-authentication** 例： `Device# debug ip trigger-authentication`	自動二重認証に関する debug の出力が表示されます。

サーバ グループ レベルでのドメイン ストリッピングの設定

手順

	コマンドまたはアクション	目的
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	**aaa group server radius** *server-name* 例： `Device(config)# aaa group server radius rad1`	RADIUS サーバを追加し、サーバ グループ RADIUS コンフィギュレーション モードを開始します。 • *server-name* 引数には、RADIUS サーバ グループ名を指定します。
ステップ 4	**domain-stripping** [**strip-suffix** *word*] [**right-to-left**] [**prefix-delimiter** *word*] [**delimiter** *word*] 例： `Device(config-sg-radius)# domain-stripping delimiter username@example.com`	サーバ グループ レベルでドメイン ストリッピングを設定します。
ステップ 5	**end** 例： `Device(config-sg-radius)# end`	サーバ グループ RADIUS コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

非 AAA 認証方式の設定

ライン パスワード保護の設定

このタスクは、パスワードを入力し、パスワードチェック処理を確立することで、端末回線にアクセス コントロールを提供するために使用します。

（注） ライン パスワード保護を設定し、TACACS または拡張 TACACS を設定する場合、TACACS のユーザ名とパスワードの方が、ライン パスワードよりも優先されます。まだセキュリティ ポリシーを実装していない場合、AAA を使用することを推奨します。

手順

	コマンドまたはアクション	目的
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	**line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*] 例： `Device(config)# line console 0`	ライン コンフィギュレーション モードを開始します。
ステップ 4	**password** *password* 例： `Device(config-line)# secret word`	回線上の端末または他のデバイスにパスワードを割り当てます。パスワード チェッカでは大文字と小文字が区別され、スペースを使用できます。たとえば、パスワード「Secret」とパスワード「secret」は異なるパスワードです。また、「two words」は有効なパスワードです。
ステップ 5	**login** 例： `Device(config-line)# login`	ログイン時のパスワード チェックをイネーブルにします。 このコマンドの **no** 形式を使用してパスワードチェックを無効にすると、ライン パスワード検証を無効にできます。

	コマンドまたはアクション	目的
		（注） **login** コマンドによって変更されるのはユーザ名および特権レベルだけであり、シェルは実行されません。したがって、autocommand は実行されません。この状況で autocommand を実行するには、Telnet セッションをデバイスに復帰（ループバック）させる必要があります。この方法で autocommand を実装する場合は、デバイスがセキュアな Telnet セッションを使用するように設定されていることを確認してください。
ステップ 6	**end** 例： `Device(config-line)# end`	回線コンフィギュレーション モードを終了します。続いて、特権 EXEC モードに戻ります。

ユーザ名認証の確立

ユーザ名ベースの認証システムを作成できます。これは、次のような場合に役立ちます。

- TACACS をサポートしないネットワークに、TACACS のようなユーザ名と暗号化されたパスワード認証システムを提供する場合
- 特殊なケース（たとえば、アクセスリストの確認、パスワードの確認なし、ログイン時の autocommand の実行、「エスケープなし」の状況など）に備えたログインを提供する場合

ユーザ名認証を確立するには、次のタスクを実行します。

手順

	コマンドまたはアクション	目的
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：	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。

	コマンドまたはアクション	目的
	`Device# configure terminal`	

ステップ 3	次のいずれかを実行します。
	• **username** *name* [**nopassword** | **password** *password* | **password** *encryption-type encrypted password*]
	• **username** *name* [**access-class** *number*]
	例：
	`Device(config)# username superuser password superpassword password 7 encrypted-password`
	`Device(config)# username user1 access-class access-user`
		暗号化されたパスワードを使用してユーザ名認証を確立します。
		または
		（任意）アクセス リストによるユーザ名認証を確立します。

ステップ 4	**username** *name* [**privilege** *level*] 例： `Device(config)# username user1 privilege 5`	（任意）ユーザの特権レベルを設定します。
ステップ 5	**username** *name* [**autocommand** *command*] 例： `Device(config)# username user1 autocommand show users`	（任意）自動実行されるコマンドを指定します。
ステップ 6	**username** *name* [**noescape**] [**nohangup**] 例： `Device(config)# username user1 noescapee`	（任意）「エスケープなし」のログイン環境を設定します。
ステップ 7	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

次のタスク

キーワード **noescape** を指定すると、ユーザは接続先のホストでエスケープ文字を使用できなくなります。**nohangup** 機能を使用すると、autocommand の使用後に接続が解除されません。

⚠

注意 **service password-encryption** コマンドを有効にしない限り、設定のパスワードはクリアテキストで表示されます。

MS-CHAP を使用した PPP 認証の定義

MS-CHAP を使用して PPP 認証を定義するには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

手順

	コマンドまたはアクション	目的
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	**encapsulation ppp** 例： `Device(config)# encapsulation ppp`	PPP カプセル化をイネーブルにします。
ステップ 4	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/2`	インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 5	**ppp authentication ms-chap** [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**] 例： `Device(config-if)# ppp authentication ms-chap default callin`	MS-CHAP を使用して PPP 認証を定義します。
ステップ 6	**end** 例： `Device(config-if)# end`	インターフェイスコンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

次のタスク

あるインターフェイスで **ppp authentication ms-chap** を設定する場合、PPP 接続を開始するそのインターフェイスに着信するすべてのコールは、MS-CHAP を使用して認証する必要があります。**callin** キーワードを指定して **ppp authentication** コマンドを設定すると、アクセスサーバは、リモートデバイスがコールを開始した場合にだけ、リモートデバイスの認証を行います。

認証方式リストと **one-time** キーワードを使用できるのは、AAA を有効にした場合だけです。TACACS または拡張 TACACS を有効にしている場合は、使用できません。**ppp authentication** コマンドを使用して認証方式リストの名前を指定すると、PPPは、指定した方式リストに定義されている方式を使用して、接続を認証しようとします。AAA をイネーブルにし、名前で定義されている方式リストがない場合、PPPは、デフォルトに定義されている方式を使用して接続を認証しようとします。**one-time** キーワードを指定して **ppp authentication** コマンドを使用すると、認証中にワンタイムパスワードをサポートできます。

if-needed キーワードを使用できるのは、TACACS または拡張 TACACS を使用している場合だけです。**if-needed** キーワードを指定して **ppp authentication** コマンドを使用することは、現在のコール期間中にリモートデバイスがまだ認証されていない場合にだけ、PPP が MS-CHAP を介してリモートデバイスを認証することを示します。リモートデバイスが、標準のログイン手順で認証を受け、EXEC プロンプトから PPP を開始した場合、**ppp authentication chap if-needed** が設定されていれば、PPP は MS-CHAP を介して認証しません。

（注） MS-CHAP を使用する PPP 認証と、ユーザ名認証を併用する場合、ローカル ユーザ名/パスワード データベースに MS-CHAP シークレットを含める必要があります。

認証の設定例

例：方式リストの設定

たとえば、システム管理者が、すべてのインターフェイスに同じ認証方式を使用して PPP 接続を認証する、というセキュリティ ソリューションを決定したとします。RADIUS グループでは、まず認証情報のために R1 に接続し、応答がない場合、R2 に接続します。R2 が応答しない場合、TACACS+ グループの T1 に接続し、T1 が応答しない場合、T2 に接続します。すべての指定したサーバが応答しなかった場合、認証はアクセス サーバ自体のローカル ユーザ名データベースで行われます。このソリューションを実装するには、システム管理者が次のコマンドを入力してデフォルトの方式リストを作成します。

```
Device> enable
Device# configure terminal
Device(config)# aaa authentication ppp default group radius group tacacs+ local
Device(config)# exit
```

この例では、「default」が方式リストの名前です。この方式リストにプロトコルを含める場合、名前の後に、照会される順で指定します。デフォルトのリストは、すべてのインターフェイスに自動的に適用されます。

リモート ユーザがネットワークにダイヤルインしようとすると、ネットワーク アクセス サーバは、まず R1 に認証情報を照会します。ユーザが R1 から認証されると、R1 からネットワーク アクセス サーバに対して PASS 応答が発行され、ユーザはネットワークにアクセスできるようになります。R1 から FAIL 応答が返されると、ユーザはアクセスを拒否され、セッションは終了します。R1 が応答しない場合、ネットワーク アクセス サーバでは ERROR として処理され、認証情報について R2 に照会されます。このパターンは、ユーザが認証または拒否されるか、セッションが終了するまで、残りの指定した方式について続行されます。

FAIL 応答は ERROR とまったく異なる点に注意してください。FAIL とは、適用可能な認証データベースに含まれる、認証の成功に必要な基準をユーザが満たしていないことを示します。認証は FAIL 応答で終了します。ERROR とは、認証の照会に対してサーバが応答しなかったことを示します。そのため、認証は試行されません。ERROR が検出された場合にだけ、認証方式リストに定義されている次の認証方式が AAA によって選択されます。

たとえば、システム管理者が、1 つのインターフェイス、または一部のインターフェイスにだけ方式リストを適用するとします。この場合、システム管理者は名前付き方式リストを作成し、その名前付きリストを対象のインターフェイスに適用します。次に、システム管理者が、インターフェイス 3 にだけ適用する認証方式を実装する場合の例を示します。

```
Device> enable
Device# configure terminal
Device(config)# Device(config)#
Device(config)# aaa authentication ppp server-group1 group radius group tacacs+ local
none
Device(config)# interface gigabitethernet 1/0/3
Device(config-if)#  ppp authentication chap server-group1
Device(config-if)# end
```

この例では、「apple」が方式リストの名前です。また、この方式リストに含まれるプロトコルは、名前の後に、実行する順で指定されています。方式リストを作成すると、該当するインターフェイスに適用されます。AAA および PPP 認証コマンド両方の方式リスト名（apple）は一致する必要があります。

次の例では、システム管理者がサーバ グループを使用し、PPP 認証の場合は R2 および T2 だけが有効であることを指定します。この場合、管理者は、メンバがそれぞれ R2（172.16.2.7）と T2（172.16.2.77）であるサーバ グループを定義する必要があります。この例では、RADIUS サーバグループ「rad2only」は**aaa group server** コマンドを使用して次のように定義されます。

```
Device> enable
Device# configure terminal
Device(config)# aaa group server radius rad2only
Device(config-sg-radius)# server 172.16.2.7
Device(config-sg-radius)# end
```

TACACS+ サーバグループ「tac2only」は、**aaa group server** コマンドを使用して次のように定義されます。

```
Device> enable
Device# configure terminal
Device(config)# aaa group server tacacs+ tac2only
```

```
Device(config-sg-tacacs)# server 172.16.2.77
Device(config-sg-tacacs)# end
```

次に、管理者はサーバ グループを使用して PPP 認証を適用します。この例では、PPP 認証用のデフォルト方式リストは **group rad2only** 、**group tac2only**、**local** の順序に従います。

```
Device> enable
Device# configure terminal
Device(config)# aaa authentication ppp default group rad2only group tac2only local
Device(config)# exit
```

VTY 回線で方式リストを設定する場合、対応する方式リストを AAA に追加する必要があります。次の例は、VTY 回線の下に方式リストを設定する方法を示しています。

```
Device> enable
Device# configure terminal
Device(config)# line vty 0 4
Device(config-line)# authorization commands 15 auth1
Device(config-line)# exit
```

次の例は、AAA で方式リストを設定する方法を示しています。

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authorization commands 15 auth1 group tacacs+
Device(config)# exit
```

VTY 回線で方式リストを設定しない場合、デフォルトの方式リストを AAA に追加する必要があります。次の例は、方式リストを使用しない VTY 設定を示しています。

```
Device> enable
Device# configure terminal
Device(config)# line vty 0 4
Device(config-line)# end
```

次の例は、デフォルトの方式リストを設定する方法を示しています。

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authorization commands 15 default group tacacs+
Device(config)# exit
```

例：RADIUS 認証

ここでは、RADIUS を使用する 2 つの設定例を紹介します。

次に、RADIUS を使用して認証および認可を行うようにルータを設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# aaa authentication login radius-login group radius local
Device(config)# aaa authentication ppp radius-ppp if-needed group radius
Device(config)# aaa authorization exec default group radius if-authenticated
Device(config)# aaa authorization network default group radius
```

```
Device(config)# line 3
Device(config-line)# login authentication radius-login
Device(config-line)# exit
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# ppp authentication radius-ppp
Device(config-if)# end
```

この RADIUS 認証および認可設定のサンプル行は、次のように定義されます。

- aaa authentication login radius-login group radius local コマンドを実行すると、ルータは、ログイン プロンプトで認証に RADIUS を使用するように設定されます。RADIUS がエラーを返すと、ユーザはローカル データベースを使用して認証されます。
- aaa authentication ppp radius-ppp if-needed group radius コマンドを実行すると、ユーザがまだログインしていない場合、Cisco IOS XE ソフトウェアは CHAP または PAP による PPP 認証を使用するように設定されます。EXEC 施設がユーザを認証すると、PPP 認証は実行されません。
- aaa authorization exec default group radius if-authenticated コマンドを実行すると、autocommand や特権レベルなど、EXEC 認可時に使用される情報について、RADIUS データベースに照会されます。ただし、ユーザの認証が成功した場合にだけ、権限が付与されます。
- aaa authorization network default group radius コマンドを実行すると、ネットワーク認可、アドレス割り当て、および他のアクセス リストについて RADIUS に照会されます。
- **login authentication radius-login** コマンドを使用すると、ライン 3 について radius-login 方式リストが有効になります。
- **ppp authentication radius-ppp** コマンドを使用すると、シリアルインターフェイス 0 について radius-ppp 方式リストが有効になります。

次に、ユーザ名とパスワードの入力を求め、その内容を確認し、ユーザの EXEC レベルを認可し、特権レベル 2 の認可方式として指定するように、ルータを設定する例を示します。この例では、ユーザ名プロンプトにローカルユーザ名を入力すると、そのユーザ名が認証に使用されます。

ローカルデータベースを使用してユーザが認証されると、RADIUS 認証からのデータは保存されないため、RADIUS を使用する EXEC 認可は失敗します。また、この方式リストではローカル データベースを使用して autocommand を検索します。autocommand がない場合、ユーザは EXEC ユーザになります。次に、ユーザが特権レベル 2 に設定されているコマンドを発行しようとすると、TACACS+ を使用してコマンドの認可が試行されます。

```
Device> enable
Device# configure terminal
Device(config)# aaa authentication login default group radius local
Device(config)# aaa authorization exec default group radius local
Device(config)# aaa authorization command 2 default group tacacs+ if-authenticated
Device(config)# radius server radserver
Device(config-sg-radius)# address ipv4 10.2.3.1
Device(config-sg-radius)# exit
Device(config)# radius-server attribute 44 include-in-access-req
Device(config)# radius-server attribute 8 include-in-access-req
Device(config)# end
```

この RADIUS 認証および認可設定のサンプル行は、次のように定義されます。

- aaa authentication login default group radius local コマンドにより、RADIUS（RADIUS が応答しない場合はルータのローカル ユーザ データベース）がユーザ名およびパスワードを確認するように指定します。
- aaa authorization exec default group radius local コマンドにより、RADIUS を使用してユーザが認証される場合、ユーザの EXEC レベルの設定に RADIUS 認証情報を使用するように指定します。RADIUS 情報が使用されない場合、このコマンドにより、EXEC 認可にローカル ユーザ データベースが使用されるように指定します。
- aaa authorization command 2 default group tacacs+ if-authenticated コマンドにより、すでにユーザの認証が成功している場合、特権レベル 2 に設定されているコマンドに TACACS+ 認可を指定します。
- radius-server host 172.16.71.146 auth-port 1645 acct-port 1646 コマンドにより、RADIUS サーバ ホストの IP アドレス、認証要求の UDP 宛先ポート、およびアカウンティング要求の UDP 宛先ポートを指定します。
- radius-server attribute 44 include-in-access-req コマンドにより、access-request パケットで RADIUS 属性 44（Acct-Session-ID）を送信します。
- radius-server attribute 8 include-in-access-req コマンドにより、access-request パケットで RADIUS 属性 8（Framed-IP-Address）を送信します。

例：TACACS 認証

次に、PPP 認証に使用するセキュリティ プロトコルとして TACACS+ を設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication ppp test group tacacs+ local
Device(config)# interface gigabitethernet 1/1/2
Device(config-if)# ppp authentication chap pap test
Device(config-if)# exit
Device(config)# tacacs-server host 192.0.2.3
Device(config)# tacacs-server key key1
Device(config)# end
```

この TACACS+ 認証設定のサンプル行は、次のように定義されます。

- **aaa new-model** コマンドは、AAA セキュリティ サービスをイネーブルにします。
- **aaa authentication** コマンドにより、PPP を実行するシリアル インターフェイスに使用する方式リスト「test」を定義します。キーワード **group tacacs+** は、TACACS+ を介して認証を実行することを示します。認証中に TACACS+ から何らかのエラーが返される場合、キーワード **local** は、ネットワーク アクセス サーバ上のローカル データベースを使用して認証が試行されることを示します。
- **interface** コマンドにより、回線を選択します。

- **ppp authentication** コマンドにより、この回線に test 方式リストを適用します。
- **tacacs-server host** コマンドにより、TACACS+ デーモンが 192.0.2.3 という IP アドレスを持っていると指定します。
- **tacacs-server key** コマンドにより、共有暗号キーが「key1」になるように定義します。

次に、PPP に AAA 認証を設定する例を示します。

```
Device(config)# aaa authentication ppp default if-needed group tacacs+ local
```

この例のキーワード **default** は、デフォルトですべてのインターフェイスに PPP 認証が適用されることを示します。**if-needed** キーワードは、ユーザが ASCII ログイン手順を介してすでに認証済みの場合、PPP は不要なので、スキップできることを示します。認証が必要な場合、**group tacacs+** キーワードは、TACACS+ を介して認証が実行されることを示します。認証中に TACACS+ から何らかのエラーが返される場合、キーワード **local** は、ネットワーク アクセス サーバ上のローカル データベースを使用して認証が試行されることを示します。

次に、PAP に同じ認証アルゴリズムを作成し、「default」ではなく「MIS-access」の方式リストを呼び出す例を示します。

```
Device> enable
Device# configure terminal
Device(config)# aaa authentication ppp MIS-access if-needed group tacacs+ local
Device(config)# interface gigabitethernet 1/1/2
Device(config)# ppp authentication pap MIS-access
Device(config)# end
```

この例では、リストはどのインターフェイスにも適用されないため（自動的にすべてのインターフェイスに適用されるデフォルトリストとは異なります）、管理者は **interface** コマンドを使用して、この認証スキームを適用するインターフェイスを選択する必要があります。次に、管理者は **ppp authentication** コマンドを使用して、選択したインターフェイスにこの方式リストを適用する必要があります。

例：Kerberos 認証

ログイン認証方式として Kerberos を指定するには、次のコマンドを使用します。

```
Device> enable
Device# configure terminal
Device(config)# aaa authentication login default krb5
Device(config)# end
```

PPP に Kerberos 認証を指定するには、次のコマンドを使用します。

```
Device> enable
Device# configure terminal
Device(config)# aaa authentication ppp default krb5
Device(config)# end
```

例：AAA スケーラビリティ

次に、セキュリティ プロトコルとしてRADIUSによるAAAを使用する一般的なセキュリティ設定例を示します。この例では、ネットワーク アクセス サーバは、16バックアップ プロセスを割り当てて PPP に対する AAA 要求を処理するように設定されています。

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# radius server radserver
Device(config-sg-radius)# address ipv4 radius-host
Device(config-sg-radius)# key myRaDiUSpassWoRd
Device(config-sg-radius)# exit
Device(config)# radius-server configure-nas
Device(config)# username root password ALongPassword
Device(config)# aaa authentication ppp dialins group radius local
Device(config)# aaa authentication login admins local
Device(config)# aaa authorization network default group radius local
Device(config)# aaa accounting network default start-stop group radius
Device(config)# aaa processes 16
Device(config)# line 1 16
Device(config-line)# autoselect ppp
Device(config-line)# autoselect during-login
Device(config-line)# login authentication admins
Device(config-line)# modem dialin
Device(config-line)# exit
Device(config)# interface gigabitethernet 1/2/0
Device(config-if)# group-range 1 16
Device(config-if)# encapsulation ppp
Device(config-if)# ppp authentication pap dialins
Device(config-if)# end
```

この RADIUS AAA 設定のサンプル行は、次のように定義されます。

- **aaa new-model** コマンドは、AAA ネットワーク セキュリティ サービスをイネーブルにします。
- **address ipv4** {*hostname* | *host-address*} コマンドはRADIUS サーバホストの名前を定義します。
- **key** コマンドは、ネットワーク アクセス サーバと RADIUS サーバ ホストの間の共有秘密テキスト文字列を定義します。
- **radius-server configure-nas** コマンドは、デバイスが最初に起動したときに、シスコ ルータまたはアクセスサーバがスタティックルートと IP プール定義について RADIUS サーバに照会するように定義します。
- **username** コマンドはユーザ名とパスワードを定義します。これらの情報は、PPPパスワード認証プロトコル（PAP）の発信元身元確認に使用されます。
- **aaa authentication ppp dialins group radius local** コマンドで、まず RADIUS 認証を指定する認証方式リスト「dialins」を定義します。次に、（RADIUS サーバが応答しない場合）PPP を使用するシリアル回線でローカル認証が使用されます。
- **aaa authentication login admins local** コマンドは、ログイン認証に別の方式リスト「admins」を定義します。

- **aaa authorization network default group radius local** コマンドは、アドレスと他のネットワーク パラメータを RADIUS ユーザに割り当てるために使用されます。
- **aaa accounting network default start-stop group radius** コマンドは、PPP の使用状況を追跡します。
- **aaa processes** コマンドにより、PPP に対する AAA 要求を処理するために 16 個のバックグラウンドプロセスを割り当てます。
- **line** コマンドはコンフィギュレーション モードをグローバル コンフィギュレーションからライン コンフィギュレーションに切り替え、設定対象の回線を指定します。
- **autoselect ppp** コマンドは、選択した回線上で PPP セッションを自動的に開始できるようにします。
- **autoselect during-login** コマンドを使用すると、Return キーを押さずにユーザ名およびパスワードのプロンプトが表示されます。ユーザがログインすると、autoselect 機能（この場合は PPP）が開始します。
- **login authentication admins** コマンドは、ログイン認証に「admins」方式リストを適用します。
- **modem dialin** コマンドは、選択した回線に接続されているモデムを設定し、着信コールだけを受け入れるようにします。
- **interface group-async** コマンドは、非同期インターフェイス グループを選択して定義します。
- **group-range** コマンドは、インターフェイス グループ内のメンバ非同期インターフェイスを定義します。
- **encapsulation ppp** コマンドは、指定のインターフェイスに使用されるカプセル化方式として PPP を設定します。
- **ppp authentication pap dialins** コマンドは「dialins」方式リストを指定したインターフェイスに適用します。

例：AAA 認証のログイン バナーおよび Failed-Login バナーの設定

次に、ユーザがシステムにログインするときに表示されるログイン バナー（この場合、「Unauthorized Access Prohibited」というフレーズ）を設定する例を示します。アスタリスク（*）はデリミタとして使用されます。RADIUS はデフォルト ログイン認証方式として指定されます。

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication banner *Unauthorized Access Prohibited*
Device(config)# aaa authentication login default group radius
Device(config)# end
```

この設定によって、次のログイン バナーが表示されます。

```
Unauthorized Access Prohibited
Username:
```

次の例では、ユーザがシステムにログインしようとして失敗すると表示される Failed-Login バナー（この場合、「Failed login. Try again」というフレーズ）を設定する方法を示します。アスタリスク（*）はデリミタとして使用されます。RADIUS はデフォルト ログイン認証方式として指定されます。

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication banner *Unauthorized Access Prohibited*
Device(config)# aaa authentication fail-message *Failed login. Try again.*
Device(config)# aaa authentication login default group radius
Device(config)# end
```

この設定によって、次のログイン バナーおよび Failed-Login バナーが表示されます。

```
Unauthorized Access Prohibited
Username:
Password:
Failed login. Try again.
```

例：AAA パケット オブ ディスコネクト サーバ キー

次に、パケット オブ ディスコネクト（POD）を設定する例を示します。その結果、特定のセッション属性が指定されると、ネットワーク アクセス サーバ（NAS）の接続が終了します。

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication ppp default radius
Device(config)# aaa accounting network default start-stop radius
Device(config)# aaa accounting delay-start
Device(config)# aaa pod server server-key xyz123
Device(config)# radius server non-standard
Device(config-sg-radius)# address ipv4 10.2.1.1
Device(config-sg-radius)# key rad123
Device(config-sg-radius)# end
```

例：二重認証

ここでは、二重認証に使用できる設定例を示します。実際のネットワークおよびセキュリティ要件によっては、この例とは大幅に異なる可能性があります。

（注） 設定例には、特定の IP アドレスと他の特定の情報が含まれます。この情報は説明のための例であり、実際の設定には異なる IP アドレス、異なるユーザ名とパスワード、異なる認可ステートメントを使用します。

例：二重認証による AAA のローカルホストの設定

次の 2 つの例では、PPP とログイン認証、およびネットワークと EXEC 認可に AAA を使用するようにローカル ホストを設定する方法を示します。例はそれぞれRADIUS の例と TACACS+ の例です。

いずれの例でも、先頭の 3 行で AAA を設定し、特定のサーバを AAA サーバとして設定しています。続く 2 行で PPP およびログイン認証に AAA を設定し、最後の 2 行でネットワークおよびEXEC 認可を設定します。最後の行が必要なのは、**access-profile** コマンドを autocommand として実行する場合だけです。

次に、RADIUS AAA サーバを使用するデバイス設定の例を示します。

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# radius server radserver
Device(config-sg-radisu)# address ipv4 secureserver
Device(config-sg-radius)# key myradiuskey
Device(config-sg-radius)# exit
Device(config)# aaa authentication ppp default group radius
Device(config)# aaa authentication login default group radius
Device(config)# aaa authorization network default group radius
Device(config)# aaa authorization exec default group radius
Device(config)# end
```

次に、TACACS+ サーバを使用するデバイス設定の例を示します。

```
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# tacacs-server host security
Device(config)# tacacs-server key mytacacskey
Device(config)# aaa authentication ppp default group tacacs+
Device(config)# aaa authentication login default group tacacs+
Device(config)# aaa authorization network default group tacacs+
Device(config)# aaa authorization exec default group tacacs+
Device(config)# end
```

例：第 1 段階の PPP 認証と許可に関する AAA サーバの設定

次に、AAA サーバでの設定例を示します。また、RADIUS 用の AAA 設定例の一部を示します。

TACACS+ サーバも同様に設定できます（「TACACS による設定完了の例」を参照してください）。

この例では、二重認証の第 1 段階で CHAP によって認証される「hostx」というリモート ホストに関する認証/認可を定義します。ACL AV ペアは、リモート ホストによる Telnet 接続をローカル ホストに制限しています。ローカル ホストの IP アドレスは 10.0.0.2 です。

次に、RADIUS 用の AAA サーバの設定例を一部示します。

```
hostx   Password = "welcome"
        User-Service-Type = Framed-User,
```

```
        Framed-Protocol = PPP,
        cisco-avpair = "lcp:interface-config=ip unnumbered fastethernet 0",
        cisco-avpair = "ip:inacl#3=permit tcp any 172.21.114.0 0.0.0.255 eq telnet",
        cisco-avpair = "ip:inacl#4=deny icmp any any",
        cisco-avpair = "ip:route#5=10.0.0.0 255.0.0.0",
        cisco-avpair = "ip:route#6=10.10.0.0 255.0.0.0",
        cisco-avpair = "ipx:inacl#3=deny any"
```

例：第 2 段階の Per-User 認証と許可に関する AAA サーバの設定

ここでは、RADIUS サーバでの AAA 設定例の一部を示します。これらの設定では、ユーザ名が「user1」のユーザの認証と許可を定義します。このユーザは、二重認証の第2段階でユーザ認証されます。

TACACS+ サーバも同様に設定できます

3 つの例は、**access-profile** コマンドの 3 つの各形式で使用できる RADIUS AAA 設定の例を示します。

最初の例は、**access-profile** コマンドのデフォルトの形式（キーワードなし）で機能する AAA 設定例の一部を示します。1 つの ACL AV ペアのみが定義されます。また、この例では autocommand として **access-profile** コマンドも設定します。

```
user1     Password = "welcome"
          User-Service-Type = Shell-User,
          cisco-avpair = "shell:autocmd=access-profile"
          User-Service-Type = Framed-User,
          Framed-Protocol = PPP,
          cisco-avpair = "ip:inacl#3=permit tcp any host 10.0.0.2 eq telnet",
          cisco-avpair = "ip:inacl#4=deny icmp any any"
```

2 番目の例は、**access-profile** コマンドの **access-profile merge** 形式で機能する AAA 設定例の一部を示します。また、この例では autocommand として **access-profile merge** コマンドも設定します。

```
user1     Password = "welcome"
          User-Service-Type = Shell-User,
          cisco-avpair = "shell:autocmd=access-profile merge"
          User-Service-Type = Framed-User,
          Framed-Protocol = PPP,
          cisco-avpair = "ip:inacl#3=permit tcp any any"
          cisco-avpair = "ip:route=10.0.0.0 255.255.0.0",
          cisco-avpair = "ip:route=10.1.0.0 255.255.0.0",
          cisco-avpair = "ip:route=10.2.0.0 255.255.0.0"
```

3 番目の例は、**access-profile** コマンドの **access-profile replace** 形式で機能する AAA 設定例の一部を示します。また、この例では autocommand として **access-profile replace** コマンドも設定します。

```
user1     Password = "welcome"
          User-Service-Type = Shell-User,
          cisco-avpair = "shell:autocmd=access-profile replace"
          User-Service-Type = Framed-User,
          Framed-Protocol = PPP,
          cisco-avpair = "ip:inacl#3=permit tcp any any",
          cisco-avpair = "ip:inacl#4=permit icmp any any",
          cisco-avpair = "ip:route=10.10.0.0 255.255.0.0",
```

```
cisco-avpair = "ip:route=10.11.0.0 255.255.0.0",
cisco-avpair = "ip:route=10.12.0.0 255.255.0.0"
```

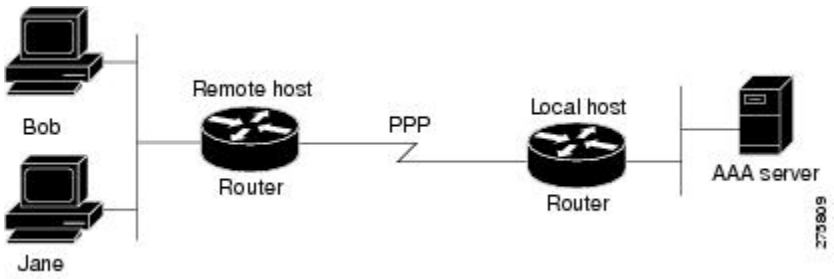
例：TACACS による設定完了

この例では、リモートホスト（二重認証の第1段階で使用）および特定のユーザ（二重認証の第 2 段階で使用）の両方向けの、TACACS+ 認可プロファイルの設定を示します。

この設定例は、リモートホスト「hostx」および 3 ユーザ（ユーザ名が「user_default」、「user_merge」、および「user_replace」）の TACACS+ サーバ上にある認証/許可プロファイルを示します。これら 3 つのユーザ名の設定は、**access-profile** コマンドの 3 種類のフォームに対応する異なる設定を示しています。また、3 つのユーザ設定は、**access-profile** コマンドの各形式について autocommand の設定方法も示しています。

次の図に、トポロジを示します。図の後に、TACACS+ 設定ファイルの例を示します。

図 *3:* 二重認証のトポロジ例



この設定例は、リモートホスト「hostx」および 3 ユーザ（ユーザ名が「user_default」、「user_merge」、および「user_replace」）の TACACS+ サーバ上にある認証/許可プロファイルを示します。

```
key = "mytacacskey"
default authorization = permit
#-----------------------------Remote Host (BRI)-------------------------
#
# This allows the remote host to be authenticated by the local host
# during fist-stage authentication, and provides the remote host
# authorization profile.
#
#-----------------------------------------------------------------------
user = hostx
{
    login = cleartext "welcome"
    chap = cleartext "welcome"
    service = ppp protocol = lcp {
                interface-config="ip unnumbered fastethernet 0"
    }
    service = ppp protocol = ip {
            # It is important to have the hash sign and some string after
            # it. This indicates to the NAS that you have a per-user
            # config.
            inacl#3="permit tcp any 172.21.114.0 0.0.0.255 eq telnet"
            inacl#4="deny icmp any any"
            route#5="10.0.0.0 255.0.0.0"
            route#6="10.10.0.0 255.0.0.0"
    }
```

```
    service = ppp protocol = ipx {
            # see previous comment about the hash sign and string, in protocol = ip
            inacl#3="deny any"
    }
}
#------------------- "access-profile" default user "only acls" ------------------
#
# Without arguments, access-profile removes any access-lists it can find
# in the old configuration (both per-user and per-interface), and makes sure
# that the new profile contains ONLY access-list definitions.
#
#-------------------------------------------------------------------------------
user = user_default
{
        login = cleartext "welcome"
        chap = cleartext "welcome"
        service = exec
        {
                # This is the autocommand that executes when user_default logs in.
                autocmd = "access-profile"
        }
        service = ppp protocol = ip {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!
                inacl#3="permit tcp any host 10.0.0.2 eq telnet"
                inacl#4="deny icmp any any"
        }
        service = ppp protocol = ipx {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!
        }
}
#---------------------- "access-profile merge" user ---------------------------
#
# With the 'merge' option, first all old access-lists are removed (as before),
# but then (almost) all AV pairs are uploaded and installed. This will allow
# for uploading any custom static routes, sap-filters, and so on, that the user
# may need in his or her profile. This needs to be used with care, as it leaves
# open the possibility of conflicting configurations.
#
#-------------------------------------------------------------------------------
user = user_merge
{
        login = cleartext "welcome"
        chap = cleartext "welcome"
        service = exec
        {
                # This is the autocommand that executes when user_merge logs in.
                autocmd = "access-profile merge"
        }
        service = ppp protocol = ip
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!
                inacl#3="permit tcp any any"
```

```
                route#2=”10.0.0.0 255.255.0.0"
                route#3=”10.1.0.0 255.255.0.0"
                route#4=”10.2.0.0 255.255.0.0"
        }
        service = ppp protocol = ipx
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!
        }
}
#--------------------- “access-profile replace” user ---------------------------
#
# With the 'replace' option, ALL old configuration is removed and ALL new
# configuration is installed.
#
# One caveat: access-profile checks the new configuration for address-pool and
# address AV pairs. As addresses cannot be renegotiated at this point, the
# command will fail (and complain) when it encounters such an AV pair.
# Such AV pairs are considered to be “invalid” for this context.
#-------------------------------------------------------------------------------
user = user_replace
{
        login = cleartex
t
“
welcome
”
        chap = cleartext “welcome”
        service = exec
        {
                # This is the autocommand that executes when user_replace logs in.
                autocmd = “access-profile replace”
        }
        service = ppp protocol = ip
        {
                # Put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IP
                # access-lists (not even the ones installed prior to
                # this)!
                inacl#3=”permit tcp any any”
                inacl#4=”permit icmp any any”
                route#2=”10.10.0.0 255.255.0.0"
                route#3=”10.11.0.0 255.255.0.0"
                route#4=”10.12.0.0 255.255.0.0"
        }
        service = ppp protocol = ipx
        {
                # put whatever access-lists, static routes, whatever
                # here.
                # If you leave this blank, the user will have NO IPX
                # access-lists (not even the ones installed prior to
                # this)!
        }
}
```

例：自動二重認証

次に、自動二重認証が設定された設定ファイル全体の例を示します。自動二重認証に適用されるコンフィギュレーション コマンドは、2つのアスタリスク（**）を使用した記述よりも優先されます。

```
Current configuration:
!
version 16.10
no service password-encryption
!
hostname myrouter
!
!
! **The following AAA commands are used to configure double authentication:
!
! **The following command enables AAA:
aaa new-model
! **The following command enables user authentication via the RADIUS AAA server:
!
aaa authentication login default none
aaa authentication ppp default group radius
! **The following command causes the remote user's authorization profile to be
! downloaded from the AAA server to the router when required:
!
aaa authorization network default group radius
!
enable password mypassword
!
ip host blue 172.21.127.226
ip host green 172.21.127.218
ip host red 172.21.127.114
ip domain-name example.com
ip name-server 172.16.2.75
!
!
interface GigabitEthernet0/0/0
 ip address 172.21.127.186 255.255.255.248
 no ip route-cache
 no ip mroute-cache
 no keepalive
 ntp disable
 no cdp enable
!
interface Virtual-Template1
 ip unnumbered loopback0
 no ip route-cache
 no ip mroute-cache
!
! **The following command specifies that device authentication occurs via PPP CHAP:
 ppp authentication chap
!
router eigrp 109
 network 172.21.0.0
 no auto-summary
!
ip default-gateway 172.21.127.185
no ip classless
ip route 172.21.127.114 255.255.255.255 172.21.127.113
! **Virtual profiles are required for double authentication to work:
virtual-profile virtual-template 1
```

```
dialer-list 1 protocol ip permit
no cdp run
! **The following command defines where the TACACS+ AAA server is:
tacacs-server host 172.16.57.35 port 1049
tacacs-server timeout 90
! **The following command defines the key to use with TACACS+ traffic (required):
tacacs-server key mytacacskey
snmp-server community public RO
!
line con 0
 exec-timeout 0 0
 login authentication console
line aux 0
 transport input all
line vty 0 4
 exec-timeout 0 0
 password lab
!
end
```

認証設定の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Fuji 16.9.2	AAA Authentication	認証は、選択したセキュリティ プロトコルに応じてログイン/パスワード ダイアログ、チャレンジ/レスポンス、メッセージング サポート、および暗号化などのユーザの識別方法を提供します。認証は、ユーザに対してネットワークとネットワーク サービスへのアクセスを許可する前に、ユーザの識別を行う方法です。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn からアクセスします。