



Guard モジュールの設定

この章では、Cisco Anomaly Guard Module (Guard モジュール) のサービスの設定方法について説明します。

この章には、Guard モジュールの関連製品である Cisco Detector (Detector) についての記述があります。Detector は、DDoS 攻撃 (分散型サービス拒絶攻撃) を検出するデバイスです。Detector は、ゾーントラフィックのコピーを分析します。Detector は、ゾーンが攻撃を受けていると判断したときに、Guard モジュールの攻撃軽減サービスをアクティブにできます。また、Detector は Guard モジュールとゾーン設定を同期させることができます。Detector の詳細については、『*Cisco Traffic Anomaly Detector Module Configuration Guide*』および『*Cisco Traffic Anomaly Detector Configuration Guide*』を参照してください。

この章は、次の項で構成されています。

- [Guard モジュールのサービスのアクティブ化](#)
- [AAA を使用したアクセス コントロールの設定](#)
- [Detector との通信の確立](#)
- [SSH 鍵の管理](#)
- [SFTP 接続および SCP 接続用の鍵の設定](#)
- [ホスト名の変更](#)
- [SNMP トラップのイネーブル化](#)
- [SNMP コミュニティ スtring の設定](#)
- [ログイン バナーの設定](#)
- [Web-Based Manager ログの設定](#)
- [セッション タイムアウトの設定](#)

Guard モジュールのサービスのアクティブ化

Guard モジュールには、いくつかのサービス オプションがあります。そのサービス オプションをアクティブにするには、サービスをイネーブルにしてから、サービスへのアクセスを許可される IP アドレスを定義します。この項では、常にアクティブであるセキュア シェル サービス以外のサービスをアクティブにする方法について説明します。

Guard モジュールのサービスは次のとおりです。

- ノード間通信サービス: Guard モジュールは、Detector との通信チャネルを確立するときこのサービスを使用します。詳細については、P.4-18 の「Detector との通信の確立」を参照してください。
- Simple Network Management Protocol (SNMP; 簡易ネットワーク管理プロトコル) サーバ サービス: SNMP を使用して Guard モジュールにアクセスすることにより、次の MIB で定義された情報を取得できます。
 - Riverhead の専用 MIB。
 - MIB2 (RFC1213-MIB): EGP グループおよび伝送 MIB グループ以外のすべての MIB グループ。
 - UCDAVIS (UCD-SNMP-MIB): memory、latable、systemStats、version、および snmperrs の MIB グループのみ。

MIB 定義の詳細については、このソフトウェア バージョンでリリースされた MIB ファイルを参照してください。



(注) Riverhead MIB には、64 ビットのカウンタが含まれています。MIB を読み取るには、SNMP バージョン 2 をサポートするブラウザを使用する必要があります。

- SNMP トラップ サービス: snmp-trap サービスをアクティブにすると、Guard モジュールは SNMP トラップを生成します。詳細については、P.4-27 の「SNMP トラップのイネーブル化」を参照してください。
- Secure Shell (SSH; セキュア シェル) サービス: SSH サービスは常にアクティブです。詳細については、P.3-17 の「SSH を使用した Guard モジュールへのアクセス」および P.4-23 の「SSH 鍵の管理」を参照してください。
- Web-Based Manager (WBM) サービス: Web ブラウザを使用して Guard モジュールを監視および制御できます。詳細については、P.3-15 の「Cisco Web-Based Manager による Guard モジュールの管理」を参照してください。
- MultiDevice Manager (MDM) サービス: Web ブラウザを使用して、MDM サーバから Guard モジュール、およびその他の Guard や Detector デバイスを監視および制御できます。詳細については、P.3-16 の「Cisco DDoS MultiDevice Manager による Guard モジュールの管理」を参照してください。

デフォルトでは、SSH 以外、Guard モジュールのすべてのサービスはディセーブルになっています。

Guard モジュールのサービスをアクティブにするには、次の手順を実行します。

ステップ 1 設定モードで次のコマンドを入力して、Guard モジュールのサービスをイネーブルにします。

```
service {internode-comm | mdm | snmp-server | snmp-trap | wbm}
```

表 4-1 に、service コマンドのキーワードを示します。

表 4-1 service コマンドのキーワード

サービス	説明
internode-comm	ノード間通信サービスを指定します。
mdm	MDM サービスを指定します。
snmp-server	SNMP サーバ サービスを指定します。
snmp-trap	SNMP トラップ サービスを指定します。
wbm	WBM サービスを指定します。

ステップ 2 次のいずれかのコマンドを入力して、Guard モジュールのサービスへのアクセスを許可します。

- MDM サービスの場合は、設定モードで次のコマンドを入力することにより、MDM から Guard モジュールのサービスへのアクセスを許可します。

```
mdm server ip-addr
```


ip-addr 引数には、MDM サーバの IP アドレスを指定します。IP アドレスをドット区切り 10 進表記で入力します。

- 他のすべてのサービスの場合は、設定モードで次のコマンドを入力して、Guard モジュールのサービスへのアクセスを許可し、接続をイネーブルにします。

```
permit {internode-comm {ip-address-general [ip-mask] | *} | snmp-server
{ip-address-general [ip-mask] | *} | ssh {ip-address-general [ip-mask] | *}
[if-service] | wbm {ip-address-general [ip-mask] | *} [if-service]}
```

表 4-2 に、**permit** コマンドの引数とキーワードを示します。

表 4-2 permit コマンドの引数とキーワード

パラメータ	説明
internode-comm	ノード間通信サービスを指定します。
<i>ip-address-general</i>	アクセスを許可する IP アドレス。IP アドレスをドット区切り 10 進表記で入力します (たとえば 192.168.10.2)。
<i>ip-mask</i>	(オプション) IP サブネットマスク。サブネットマスクをドット区切り 10 進表記で入力します (たとえば 255.255.255.0)。デフォルトのサブネットマスクは、255.255.255.255 です。
*	すべてのデバイスの IP アドレスからのサービスへのアクセスを許可するワイルドカード文字。  注意 セキュリティ上の理由から、すべての IP アドレスからのサービスへのアクセスを許可することはお勧めしません。
snmp-server	SNMP サーバ サービスを指定します。
ssh	SSH サービスを指定します。
<i>if-service</i>	(オプションで 1 Gbps 動作のみ) ユーザアクセスを管理インターフェイスだけに制限する管理ポート指定子。デフォルトはすべてのインターフェイスです。eth1 を入力します。 デフォルト設定に戻すには、このコマンドの no バージョンを使用します。
wbm	WBM サービスを指定します。

次の例は、WBM サービスをイネーブルにする方法を示しています。

```
user@GUARD-conf# service wbm
user@GUARD-conf# permit wbm 192.168.10.35
```

AAA を使用したアクセス コントロールの設定

認証、認可、アカウントिंग（AAA）とは、Guard モジュールおよび Guard モジュールのサービスへのユーザ アクセスを制御する方式のことです。AAA には次の機能があります。

- 認証：ユーザに対しシステムおよびシステム サービスへのアクセスを許可する前に、そのユーザを識別します。
- 認可：ユーザがシステムへのアクセス権を取得した後で、実行が許可される内容を決定します。このプロセスは、ユーザ認証後に発生します。
- アカウントिंग：ユーザが実行中または実行済みの内容を記録します。アカウントングにより、ユーザがアクセスしているサービスを追跡することができます。

Guard モジュールには、次のシステム ユーザ アカウントが事前設定されています。

- **admin : admin** ユーザ アカウントには、管理者アクセス権が設定されています。そのため、Guard モジュールの CLI とその機能にアクセスできます。Guard モジュールの CLI に初めて接続するときは、このアカウントのパスワードを設定する必要があります。新しいユーザ アカウントを設定するには **admin** ユーザ アカウントを使用します。
- **riverhead : riverhead** ユーザ アカウントには、ダイナミック (dynamic) のアクセス権が設定されています。Guard モジュールはこのユーザ アカウントを使用して、Detector との最初の通信チャネルを確立します。Guard モジュール CLI に初めて接続すると、このアカウントに対するパスワードを設定するように要求されます。

システム ユーザ アカウントは削除できません。

Guard モジュールのユーザ コミュニティをドメインに分割し、パスワードを割り当てて管理アクセスの安全性を高めることができます。初期設定が完了した後は、ユーザのアクションを監視できるように新しいアカウントを作成し、システム ユーザ アカウントは使用しないことをお勧めします。

この項では、次のトピックについて取り上げます。

- [認証の設定](#)
- [認可の設定](#)
- [アカウントングの設定](#)
- [TACACS+ サーバアトリビュートの設定](#)

認証の設定

ユーザが Guard モジュールにログインしようとしたとき、または (**enable** コマンドを使用して) 上位の特権レベルを要求したときに、Guard モジュールで使用する認証方式を設定することができます。Guard モジュールは、次の認証オプションを提供します。

- ローカル認証：ローカルに設定されたログイン パスワードおよびイネーブル パスワードを使用する認証。この認証方式がデフォルトです。詳細については、[P.4-6](#) の「[ローカル認証の設定](#)」を参照してください。
- Terminal Access Controller Access Control System Plus (TACACS+) 認証：1 つ以上の TACACS+ サーバを使用するリモート ユーザ認証。



(注) TACACS+ サーバ上で認証するユーザを定義するときは、そのユーザに対する認可も定義する必要があります。これを定義しないと、ユーザは **show** コマンドにしかアクセスできなくなります (P.4-9 の「ローカル認可の設定」を参照)。

Guard モジュールは、ユーザ認証方式の一方または両方を使用するように設定できます。TACACS+ 認証を使用する場合は、複数の TACACS+ サーバを定義できます。複数の認証方式を定義すると、通信エラーによって最初の認証方式が失敗した場合にバックアップが提供されます。

Guard モジュールは、ユーザが Guard モジュールで定義した方式を、そのときの順序に従って使用することにより、ユーザを認証します。定義済みの認証方式のリストを表示するには、**show running config** コマンドを入力します。Guard モジュールは、リストの最初の方式を使用してユーザの認証を試みます。最初の認証方式が応答しない場合、続いて Guard モジュールはリストにある次の認証方式を選択し、以後応答する方式が見つかるまでこの動作を続けます。

tacacs-server first-hit コマンドを使用して、Guard モジュールが最初の TACACS+ サーバからの応答を受信する際に実行するアクションを設定できます。**first-hit** オプション (デフォルト設定) がディセーブルの場合に最初のサーバが認証を拒否すると、Guard モジュールは続けて他の TACACS+ サーバをスキャンして、認証を受け入れるサーバを検出します。定義されている TACACS+ サーバがいずれも認証を受け入れなかった場合、または Guard モジュールがどのサーバとも通信できなかった場合は、ユーザ認証は失敗します。**first-hit** オプションをイネーブルにすると、Guard モジュールは応答する最初の TACACS+ サーバの認証応答 (拒否または承認) を最終決定として受け入れます。デフォルトでは、**first-hit** オプションはディセーブルになっています。**tacacs-server first-hit** コマンドの詳細については、P.4-16 の「TACACS+ 検索方式の設定」を参照してください。



(注) Guard モジュールは、Guard モジュールが TACACS+ サーバと通信できない場合にユーザ認証でそのローカルデータベースをフォールバックとして使用するよう設定できます (「認証方式の設定」の項を参照)。

この項では、次のトピックについて取り上げます。

- [認証方式の設定](#)
- [ローカル認証の設定](#)

認証方式の設定

Guard モジュールで使用する認証方式を設定するには、次の手順を実行します。

ステップ 1 TACACS+ 認証が必要な場合は、TACACS+ サーバ接続を設定します。詳細については、P.4-14 の「TACACS+ サーバアトリビュートの設定」を参照してください。

ステップ 2 設定モードで次のコマンドを入力し、認証方式を定義します。

```
aaa authentication {enable | login} {local | tacacs+} [tacacs+ | local]
```

表 4-3 に、**aaa authentication** コマンドのキーワードを示します。

表 4-3 aaa authentication コマンドのキーワード

パラメータ	説明
enable	ユーザが上位の特権レベルを入力する場合に、Guard モジュールでの認証を許可します。
login	ユーザのログイン時に Guard モジュールでの認証を許可します。
local	ユーザの認証で Guard モジュールが使用するローカル データベースを指定します。
tacacs+	TACACS+ サーバがユーザを認証できるようにします。
tacacs+ local	(オプション) 設定された認可方式が失敗した場合の代替の認可方式を指定します。

認証方式を変更するには、このコマンドを再入力します。

次の例は、上位の特権レベルに入る際に認証を行うように設定する方法を示しています。最初の認証方式は TACACS+ に設定され、2 番目の認証方式はローカル ユーザ データベースに設定されています。

```
user@GUARD-conf# aaa authentication enable tacacs+ local
```

ローカル認証の設定

Guard モジュールには、管理者特権を持つ事前設定されたユーザ名（ユーザ定義と呼ばれる）があり、このユーザ名を使用して新しいユーザを作成できます。ユーザ定義を使用すると、Guard モジュールのユーザ コミュニティをドメインに分割し、安全な管理アクセスのためにパスワードを割り当てることができます。

TACACS+ サーバを使用した CLI ユーザの認証をイネーブルにするには、[P.4-4](#) の「[認証の設定](#)」を参照してください。

この項では、次のトピックについて取り上げます。

- [ユーザの追加](#)
- [自分のパスワードの変更](#)
- [他のユーザのパスワードの変更](#)
- [ローカル ユーザ データベースからのユーザの削除](#)

ユーザの追加

Guard モジュールのローカル データベースにユーザを追加するには、設定モードで次のコマンドを使用します。

```
username username {admin | config | dynamic | show} [password]
```

表 4-4 に、**username** コマンドの引数とキーワードを示します。

表 4-4 username コマンドの引数とキーワード

パラメータ	説明
<i>username</i>	新しいユーザの名前。英字で始まる 1 ～ 63 文字の英数字の文字列を大文字と小文字を区別して入力します。この文字列にはスペースを含めることはできませんが、アンダースコアを含めることはできます。
admin	すべての操作にアクセスできます。
config	ユーザの定義、削除、および修正に関連する操作を除いて、すべての操作にアクセスできます。
dynamic	監視と診断、保護、およびラーニングに関する操作にアクセスできます。 dynamic 特権を持つユーザは、フレックスコンテンツ フィルタおよび動的フィルタを設定することもできます。
show	監視操作と診断操作にアクセスできます。
<i>password</i>	(オプション) 新しいユーザのパスワード。6 ～ 24 文字の英数字の文字列を、スペースを使用せずに大文字と小文字を区別して入力します。パスワードを入力しない場合、入力するよう要求されます。

次の例は、新しいユーザを設定し、パスワードを設定する方法を示しています。

```
user@GUARD-conf# username Robbin config 123456
```

ユーザはパスワードをクリア テキストで入力しますが、Guard モジュールの設定ファイルでは、パスワードが暗号化された形式で表示されます。次の例は、Guard モジュールの設定ファイル (running-config) を表示します。

```
username Richard config encrypted 840xdMk3
```

上の例の **encrypted** キーワードは、パスワードが暗号化されていることを示しています。

Guard モジュール上に設定されているユーザのリストを表示するには、**show running-config** コマンドまたは **show guard** コマンドを使用します。

現在 CLI にログインしているユーザのリストを表示するには、**show users** コマンドを使用します。

自分のパスワードの変更

ユーザは、自分自身のパスワードを変更することができます。管理者は、自分自身のパスワードと、他のユーザのパスワードを変更できます (P.4-8 の「他のユーザのパスワードの変更」を参照)。

自分自身のパスワードを変更するには、次の手順を実行します。

ステップ 1 グローバル モードで次のコマンドを入力します。

```
password
```

ステップ 2 現在のパスワードを入力します。新しいパスワードの入力を求めるプロンプトが表示されます。

ステップ 3 新しいパスワードを入力します。パスワードは、スペースを含まない 6 ～ 24 文字の英数字の文字列である必要があります。パスワードでは大文字と小文字が区別されます。確認のため新しいパスワードを再入力するよう求めるプロンプトが表示されます。

次の例は、自分のパスワードを変更する方法を示しています。

```
user@GUARD# password
Old Password: <old-password>
New Password: <new-password>
Retype New Password: <new-password>
```

他のユーザのパスワードの変更

他のユーザのパスワードを変更するには、管理ユーザ特権を持っている必要があります。

他のユーザのパスワードを変更するには、次の手順を実行します。

ステップ 1 グローバル モードで次のコマンドを入力します。

```
password username-password
```

username-password 引数は、変更対象のパスワードの持ち主のユーザです。

ステップ 2 新しいパスワードを入力します。

パスワードは、スペースを含まない 6 ~ 24 文字の英数字の文字列である必要があります。パスワードでは大文字と小文字が区別されます。確認のため新しいパスワードを再入力するよう求めるプロンプトが表示されます。

次の例では、管理者がユーザ Jose のパスワードを変更する方法を示しています。

```
user@GUARD# password Jose
New Password: <new-password>
Retype New Password: <new-password>
```

ローカル ユーザ データベースからのユーザの削除

ローカル ユーザ データベースからユーザを削除すると、そのローカル ユーザ データベースのみを使用して認証を行っている場合、関連付けられているユーザが Guard モジュールにアクセスできなくなります。

Guard モジュールのローカル ユーザ データベースからユーザを削除するには、**no username username** コマンドを使用します。

次の例は、ローカル ユーザ データベースからユーザを削除する方法を示しています。

```
user@GUARD-conf# no username Robbin
```

認可の設定

システム管理者は、ユーザが使用できるサービスを制限することができます。認可をイネーブルにすると、Guard モジュールはユーザ プロファイルを確認します。ユーザ プロファイルは、ローカル ユーザ データベース内または TACACS+ セキュリティ サーバ上にあります。ユーザは、そのユーザのプロファイル内の情報で許可されている場合にのみ、要求したサービスへのアクセスを許可されます。

ユーザがコマンドを実行しようとするときに Guard モジュールで使用する認可方式を設定することができます。Guard モジュールでは、次の認可オプションが提供されています。

- TACACS+ 認可：1 つ以上の TACACS+ サーバを使用するリモート ユーザ認可方式。
次の 2 種類の TACACS+ 認可がサポートされています。
 - EXEC 認可：ユーザが Guard モジュールにログインして認証されたときに、そのユーザの特権レベルを決定します。
 - コマンド認可：ユーザがコマンドを入力すると、そのコマンドの許可を取得するために、TACACS+ サーバを調べます。TACACS+ 認可では、コマンドごとにアクセス権を指定できます。

**注意**

認可は **copy running-config** コマンドに制限することをお勧めします。その理由は、**copy running-config** コマンドを使用すると、ユーザは、設定ファイル内のすべてのコマンドを認可しているかどうかに関係なくすべての設定コマンドを実行できるからです。

- ローカル認可：コマンドグループのアクセス コントロールにローカルで設定されたユーザ プロファイルを使用する認可方式。認可は、指定された特権レベルのすべてのコマンドに対して定義されます。この認可方式がデフォルトです。

Guard モジュールは、TACACS+ サーバへの通信に失敗した場合、ローカル認可を使用できます。

ユーザ認証方式を定義する順次認証リストを設定できます。順次認証リストには認証に使用する方式を 1 つ以上指定でき、最初の認証方式への通信が失敗した場合は、バックアップが提供されます。

Guard モジュールは、最初にリストされた方式を使用してユーザを認可します。その方式が応答しない場合、Guard モジュールは 2 番目の認可方式を選択します。認可方式が両方とも成功しなかった場合にのみ、認可は失敗します。

この項では、次のトピックについて取り上げます。

- [ローカル認可の設定](#)
- [認可方式の設定](#)
- [TACACS+ サーバの設定例](#)
- [ゾーン名のタブ補完のディセーブル化](#)

ローカル認可の設定

Guard モジュールの操作にアクセスできるかどうかは、ユーザの特権レベルによって決まります。システム管理者は、ユーザが使用できる操作を制限することができます。Guard モジュールは、ユーザのプロファイルをチェックして、ユーザのアクセス権を確認します。認可されると、ユーザは、そのユーザのプロファイル内の情報で許可されている場合にのみ、要求した操作へのアクセス権を付与されます。ユーザの特権レベルについては、[表 3-1](#) を参照してください。

この項では、次のトピックについて取り上げます。

- [パスワードを使用した特権レベルの割り当て](#)
- [ユーザ特権レベル間の移動](#)

パスワードを使用した特権レベルの割り当て

管理者は、ユーザの特権レベルへのアクセスを制限するパスワードを設定できます。特権レベルおよびパスワードを指定したら、この特権レベルにアクセスする必要のあるユーザにそのパスワードを付与することができます。ユーザは、特権レベルのパスワードを知らないと、そのパスワードで保護されたレベルに移動することはできません。

ローカルパスワードを設定して特権レベルへのアクセスを制御するには、設定モードで次のコマンドを使用します。

```
enable password [level level] [password]
```

表 4-5 に、**enable password** コマンドの引数とキーワードを示します。

表 4-5 enable password コマンドの引数とキーワード

パラメータ	説明
level level	(オプション) 次のユーザ特権レベルのいずれかを指定します。 <ul style="list-style-type: none"> admin : すべての操作にアクセスできる。 config : ユーザの定義、削除、および修正に関連する操作を除いて、すべての操作にアクセスできる。 dynamic : 監視と診断、保護、およびラーニングに関する操作にアクセスできる。dynamic 特権を持つユーザは、フレックスコンテンツ フィルタおよび動的フィルタを設定することもできます。 show : 監視操作と診断操作にアクセスできる。 デフォルトのレベルは admin です。
password	(オプション) 特権レベルのパスワード。パスワードは、スペースを含まない、6～24 文字の英数字の文字列である必要があります。パスワードでは大文字と小文字が区別されます。パスワードを入力しない場合、入力するよう要求されます。

次の例は、ユーザの特権レベル **admin** にパスワードを割り当てる方法を示しています。

```
user@GUARD-conf# enable password level admin abc123
```

ユーザ特権レベル間の移動

認可されたユーザ（特権レベルパスワードを知っているユーザ）は、ユーザ特権レベル間を移動できます。

ユーザ特権レベル間を移動するには、次の手順を実行します。

ステップ 1 グローバル モードで次のコマンドを入力します。

```
enable [level]
```

表 4-6 に、ユーザ特権レベルを指定するオプションの *level* 引数の値を示します。

表 4-6 enable コマンドの引数

パラメータ	説明
admin	すべての操作にアクセスできます。これがデフォルトです。
config	ユーザの定義、削除、および修正に関連する操作を除いて、すべての操作にアクセスできます。
dynamic	監視と診断、保護、およびラーニングに関する操作にアクセスできます。dynamic 特権を持つユーザは、フレックスコンテンツ フィルタおよび動的フィルタを設定することもできます。
show	監視操作と診断操作にアクセスできます。

ステップ 2 特権レベルのパスワードを入力します。

次の例は、ユーザの特権レベル **admin** に切り替える方法を示しています。

```
user@GUARD> enable admin
Enter enable admin Password: <password>
```

show 特権レベル (表 4-5 の説明を参照) に戻る場合は、**disable** コマンドを使用します。

認可方式の設定

認可方式を設定するには、次の手順を実行します。

ステップ 1 TACACS+ 認可が必要な場合は、TACACS+ サーバ接続を設定します。詳細については、P.4-14 の「TACACS+ サーバアトリビュートの設定」を参照してください。


ステップ 2 設定モードで次のいずれかのコマンドを入力して、認可方式を定義します。

- **aaa authorization exec tacacs+**
- **aaa authorization commands level tacacs+**

認可方式のシーケンシャルなリストを設定できます。各方式について、**aaa authorization** コマンドを入力します。認証方式を削除するには、このコマンドの **no** 形式を使用します。

表 4-7 に、**aaa authorization** コマンドの引数とキーワードを示します。

表 4-7 aaa authorization コマンドの引数とキーワード

パラメータ	説明
exec	ユーザが EXEC シェルの実行を許可されているかどうかを判別するために認可が実行されます。Guard モジュールは、TACACS+ サーバに確認して、認証されたユーザの特権レベルを判別します。  注意 ユーザ認可を設定する前に、TACACS+ サーバでユーザの認証を設定する必要があります。設定をしていないと、Guard モジュールにアクセスできないことがあります (P.4-8 の「認可の設定」を参照)。
commands	指定された特権レベルのすべてのコマンドに対して認可が実行されます。複数の特権レベルの認可を設定するには、認可が必要な特権レベルごとにこのコマンドを使用します。
level	次の特権レベルの 1 つを認可します。 <ul style="list-style-type: none"> • admin : すべての操作にアクセスできる。 • config : ユーザの定義、削除、および修正に関連する操作を除いて、すべての操作にアクセスできる。 • dynamic : 監視と診断、保護、およびラーニングに関する操作にアクセスできる。dynamic 特権を持つユーザは、フレックスコンテンツ フィルタおよび動的フィルタを設定することもできます。
tacacs+	TACACS+ サーバでユーザのアクセス権を確認します。

Guard モジュールのパフォーマンスに影響する可能性があるため、**show** 特権レベルコマンドに対する認可は設定しないことをお勧めします。



(注)

コンソールセッションから入力したコマンドには、TACACS+ 認可は実行されません。

次の例は、**config** 特権レベルを必要とするコマンドの認可を設定する方法を示しています。最初の認可方式は TACACS+ に設定され、2 番目の認可方式はローカル ユーザ データベースに設定されています。

```
user@GUARD-conf# aaa authorization commands config tacacs+ local
```



注意

設定コマンドモードにアクセスできるようにするには、**dynamic** ユーザ特権レベルに対するアクセス権を付与するか、**configure** コマンドへのアクセス権を指定する必要があります。

TACACS+ サーバの設定例

TACACS+ サーバのデータベースで、各コマンドの認可を指定することができます。

次の例は、ユーザ Zoe に対して、TACACS+ サーバ上で認可を設定する方法を示しています。

```
user=Zoe {
  cmd = protect {
    permit .*
  }
  cmd = "no protect" {
    permit .*
  }
  cmd = learning {
    deny policy*
  }
  cmd = "no learning" {
    deny .*
  }
  cmd = dynamic-filter {
    permit .*
  }
  cmd = "no dynamic-filter" {
    permit .*
  }
  cmd = flex-filter {
    deny .*
  }
  cmd = "no flex-filter" {
    deny .*
  }
}
```

ゾーン名のタブ補完のディセーブル化

ゾーン名を入力するときのタブ補完機能をディセーブルにして、ゾーン設定へのアクセスを認可されたユーザーのみに制限できます。この設定は、ゾーン名を指定するすべてのコマンドに適用されます。

グローバル モードまたは設定モードで **zone** コマンド、**no zone** コマンド、**show zone** コマンド、および **deactivate** コマンドなどのコマンドを入力しても、Guard モジュールはゾーン名の表示や補完を行わなくなります。ゾーン名を完全に入力する、ゾーン操作モードを変更する、またはゾーン統計情報を表示する必要があります。

ゾーン名のタブ補完をディセーブルにすると、Guard モジュールは **tab-complete zone-list** コマンドを TACACS+ サーバに送信します。認可されたユーザーに対してゾーン名のタブ補完をイネーブルにするには、**tab-complete zone-list** コマンドに対する認可を TACACS+ サーバ上で設定します。

次の例は、すべての **zone** コマンドでゾーン名のタブ補完をディセーブルにする方法を示しています。

```
user@GUARD-conf# aaa authorization commands zone-completion tacacs+
```

ゾーン名のタブ補完をイネーブルにするには、このコマンドの **no** 形式を使用します。

アカウントिंगの設定

アカウントिंग管理により、ユーザーがアクセスしているサービスを追跡し、TACACS+ サーバにアカウントング情報を保存することができます。課金、レポート、またはセキュリティ目的で、要求されたサービスのアカウントングをイネーブルにできます。デフォルトでは、Guard モジュールのアカウントング管理はディセーブルに設定されています。

アカウントングを設定するには、次の手順を実行します。

ステップ 1 TACACS+ サーバ接続を設定します。詳細については、[P.4-14](#) の「[TACACS+ サーバアトリビュートの設定](#)」を参照してください。

ステップ 2 設定モードで次のコマンドを入力して、アカウントングを設定します。

```
aaa accounting commands {show | dynamic | config | admin} stop-only {local | tacacs+}
```

[表 4-8](#) に、**aaa accounting commands** コマンドのキーワードを示します。

表 4-8 aaa accounting commands コマンドのキーワード

パラメータ	説明
show dynamic config admin	指定された特権レベルのアカウントングを定義します (ユーザーの特権レベルについては、 表 3-1 を参照)。
stop-only	コマンドの実行が終了したときにアクションを記録します。
local	アカウントング情報を保存しません。
tacacs+	アカウントング情報の記録に TACACS+ サーバのデータベースを使用します。

複数の特権レベルにアカウントングを設定するには、アカウントングが必要な特権レベルごとに **aaa accounting** コマンドを入力します。

アカウント管理は、`config` ユーザ特権レベルに対してのみイネーブルにすることをお勧めします。アカウントデータの追跡および保存を行うと、Guard モジュールのパフォーマンスに影響を及ぼす可能性があります。

特権レベルのアカウント管理を削除するには、このコマンドの `no` 形式を使用します。

次の例は、TACACS+ サーバ上で `config` 特権レベルを必要とするコマンドのアカウント管理を設定する方法を示しています。

```
user@GUARD-conf# aaa accounting commands config stop-only tacacs+
```

TACACS+ サーバアトリビュートの設定

TACACS+ サーバで認証、認可、またはアカウント管理をイネーブルにするには、TACACS+ サーバアトリビュートを設定する必要があります。



注意

TACACS+ 認証方式を適用する前に、TACACS+ サーバアトリビュートを設定しておく必要があります。設定していない場合は、Guard モジュールにアクセスできないことがあります。

TACACS+ サーバのアトリビュートを設定するには、次の手順を実行します。

-
- ステップ 1** `tacacs-server host ip-address port port_number` コマンドを入力して、TACACS+ サーバの IP アドレスを設定します。詳細については、[P.4-15 の「TACACS+ サーバの IP アドレスの設定」](#)を参照してください。
- ステップ 2** `tacacs-server key tacacs-key` コマンドを入力して、Guard モジュールが TACACS+ サーバへのアクセスに使用する暗号鍵を設定します。詳細については、[P.4-15 の「TACACS+ サーバの暗号鍵の設定」](#)を参照してください。
- ステップ 3** (オプション) `tacacs-server first-hit` コマンドを入力して、Guard モジュールが認証に使用する検索方式を設定します。詳細については、[P.4-16 の「TACACS+ 検索方式の設定」](#)を参照してください。
- ステップ 4** (オプション) `tacacs-server timeout timeout` コマンドを入力して、TACACS+ サーバ接続のタイムアウトを設定します。詳細については、[P.4-16 の「TACACS+ サーバの接続タイムアウトの設定」](#)を参照してください。
- ステップ 5** `show tacacs statistics` コマンドを入力して、TACACS+ サーバ接続の統計情報を表示します。詳細については、[P.4-17 の「TACACS+ サーバの統計情報の表示」](#)を参照してください。
-

Guard モジュールのユーザ特権レベルは、TACACS+ の特権番号に次のように対応しています。

- `admin` = 15
- `config` = 10
- `dynamic` = 5
- `show` = 0

この項では、次のトピックについて取り上げます。

- [TACACS+ サーバの IP アドレスの設定](#)
- [TACACS+ サーバの暗号鍵の設定](#)
- [TACACS+ 検索方式の設定](#)
- [TACACS+ サーバの接続タイムアウトの設定](#)
- [TACACS+ サーバの統計情報の表示](#)

TACACS+ サーバの IP アドレスの設定

Guard モジュールが TACACS+ サーバの一連のリストを認証、認可、およびアカウントिंगに使用するように設定できます。Guard モジュールは、TACACS+ サーバリストを使用してユーザを認証、認可、またはアカウントिंग イベントを送信します。そのサーバが応答しない場合、Guard モジュールは 2 番目のサーバを選択します。リスト上のすべてのサーバが応答しなかった場合にのみ、認証または認可は失敗します。

または、Guard モジュールがリストの最初の TACACS+ サーバのみを使用してユーザを認証するように設定することもできます（詳細については、[P.4-16](#) の「[TACACS+ 検索方式の設定](#)」を参照）。

リストには、各 TACACS+ サーバの IP アドレスを定義する必要があります。最大 9 つの TACACS+ サーバを定義できます。

リストに TACACS+ サーバを追加し、IP アドレスを割り当てるには、設定モードで次のコマンドを使用します。

```
tacacs-server host ip-address [port port_number]
```

[表 4-9](#) に、**tacacs-server host** コマンドの引数とキーワードを示します。

表 4-9 tacacs-server host コマンドの引数とキーワード

パラメータ	説明
<i>ip-address</i>	TACACS+ サーバの IP アドレス。IP アドレスをドット区切り 10 進表記で入力します（たとえば 192.168.100.1）。
port port_number	(オプション) 使用するポート番号を指定します。ポート番号を指定しない場合、Guard モジュールはポート 49（デフォルト）を使用します。

TACACS+ サーバは、入力した順序でリストに追加されます。リストには、最大 9 つのサーバを追加できます。

次の例は、TACACS+ サーバリストにサーバを追加する方法を示しています。

```
user@GUARD-conf# tacacs-server host 192.168.33.45 port 60
```

TACACS+ サーバの暗号鍵の設定

TACACS+ サーバにアクセスするための暗号鍵を設定することもできます。暗号鍵は、TACACS+ サーバ上の暗号鍵と一致する必要があります。暗号鍵にスペースを含めることはできません。

サーバの暗号アクセス鍵を設定するには、設定モードで次のコマンドを使用します。

```
tacacs-server key tacacs-key
```

引数 *tacacs-key* は、最大 100 文字の英数字の文字列です。



(注) 定義できる暗号鍵は 1 つだけです。複数の TACACS+ サーバを使用する場合、Guard モジュールは同じ鍵を使用して、すべての TACACS+ サーバとの通信を暗号化します。

次のコマンドを使用して暗号化機能をディセーブルにします。

```
no tacacs-server key
```

次の例は、TACACS+ サーバの暗号鍵を MyKey に設定する方法を示しています。

```
user@GUARD-conf# tacacs-server key MyKey
```

TACACS+ 検索方式の設定

設定モードで **tacacs-server first-hit** コマンドを使用することにより、認証拒否を最終決定と見なし、他の TACACS+ サーバでのそれ以上の検索を停止するように、Guard モジュールを設定することができます。Guard モジュールは、サーバリストの最初の TACACS+ サーバだけを使用してユーザ認証を実行し、応答します。最初の TACACS+ サーバが応答しない場合、Guard モジュールはリストにある次のサーバを選択します。Guard モジュールは、ユーザ認証に対して最初に受け取る承認または拒否を最終決定と見なし、他の TACACS+ サーバを使用してそのユーザを認証する試みを停止します。

ユーザ認証を受け入れるサーバを検出するため、定義された TACACS+ サーバの順次検索を続行するように Guard モジュールを設定するには、設定モードで **no tacacs-server first-hit** コマンドを使用します。この方式が、**first-hit** 動作のデフォルト設定です。定義された TACACS+ サーバすべてがユーザ認証を拒否する場合、または Guard モジュールがどのサーバとも通信できない場合は、ユーザ認証は失敗します。

次の例は、Guard モジュールがリスト内の最初の TACACS+ サーバのみを使用してユーザ認証をするように、TACACS+ 検索方式を設定する方法を示しています。

```
user@GUARD-conf# tacacs-server first-hit
```

TACACS+ サーバの接続タイムアウトの設定

Guard モジュールが TACACS+ サーバからの応答を待つ時間を設定できます。タイムアウトが終了すると、Guard モジュールは次の TACACS+ サーバ（そのようなサーバが設定されている場合）との接続を確立しようとするか、ローカルの AAA にフォールバックします（フォールバックが設定されている場合）。フォールバックの方式が設定されていない場合、認証と認可は失敗します。



(注) すべての TACACS+ サーバとの通信に同じサーバタイムアウトが使用されます。

TACACS+ サーバの接続タイムアウトを設定するには、設定モードで次のコマンドを使用します。

```
tacacs-server timeout timeout
```

timeout 引数には、Guard モジュールが TACACS+ サーバの応答を待つ時間を秒単位で指定します。デフォルトのタイムアウトは 0 です。

次の例は、TACACS+ サーバの接続タイムアウトを 600 秒に設定する方法を示しています。

```
user@GUARD-conf# tacacs-server timeout 600
```

**ヒント**

ネットワークに問題がある場合や、TACACS+ サーバの応答が遅いためタイムアウトが繰り返し発生する場合は、タイムアウトの値を大きくすることができます。

TACACS+ サーバの統計情報の表示

設定モードで **show tacacs statistics** コマンドを使用して、定義した TACACS+ サーバの統計情報を表示できます。

TACACS+ の統計情報をクリアするには、設定モードで **clear tacacs statistics** コマンドを使用します。

表 4-10 に、**show tacacs statistics** コマンド出力のフィールドを示します。

表 4-10 show tacacs statistics コマンド出力のフィールドの説明

フィールド	説明
PASS	Guard モジュールが TACACS+ サーバに正常にアクセスし、アクセス権を付与された回数。
FAIL	Guard モジュールが TACACS+ サーバに正常にアクセスし、アクセスを拒否された回数。
ERROR	Guard モジュールが TACACS+ サーバにアクセスできなかった回数。

Detector との通信の確立

Detector と、Detector のリモート Guard リストに定義した Guard モジュールの間に安全な通信チャネルを確立できます。安全な通信チャネルにより、Detector は次のタスクを実行できます。

- Guard モジュールのアクティブ化 : Detector は、ゾーン トラフィックの異常を検出すると、通信チャネルを使用して、ゾーン保護を行う Guard モジュールをアクティブにし、ゾーン保護中に Guard モジュールをポーリングします。
- ゾーン設定の同期化 : Detector は、通信チャネルを使用して、ゾーン設定情報を Guard モジュールと交換します。

ユーザは、Detector と Guard モジュールの両方で通信チャネル パラメータを設定した後、Detector から Guard モジュールとの接続を開始します。この接続により、Detector は、安全な通信チャネルの確立に必要な鍵と証明書を Guard モジュールと交換します。その後 Detector は接続を閉じ、Guard モジュールをアクティブにする、ゾーン設定を同期させる、または Guard モジュールにポーリングする必要がある場合に、通信チャネルを確立します。

Detector および Guard モジュールは、次の 2 つのタイプの通信チャネルをサポートしています。

- セキュア シェル (SSH) バージョン 2 : Detector が Guard をアクティブにできるようにします。
- Secure Sockets Layer (SSL) : Detector が Guard モジュールのアクティブ化、Guard モジュールのポーリング、およびゾーン設定の同期化を行うことができるようにします。

Detector 上のゾーン リモート Guard リストおよびデフォルト リモート Guard リストを使用して、Detector がゾーン保護と同期化のために通信する Guard (モジュールまたはアプライアンス) を指定します。リモート Guard リストに Guard を指定する場合は、Detector がその Guard と確立する通信チャネルのタイプ (SSH または SSL) を選択します。SSH または SSL 通信チャネルを確立するには、両方のデバイスに SSH サービスが必要です。デフォルトでは、両方のデバイスで SSH サービスが常にイネーブルになっています。SSL 通信チャネルを確立する場合、Detector は Guard との初期接続に限り SSH 通信チャネルを使用します。この初期接続時に、Detector と Guard は鍵と証明書を交換します。



(注)

Detector と Guard モジュールの間に通信チャネルを確立する前に、Guard モジュールを Detector のリモート Guard リストに追加しておく必要があります。詳細については、『Cisco Traffic Anomaly Detector Module Configuration Guide』および『Cisco Traffic Anomaly Detector Configuration Guide』を参照してください。

この項では、次のトピックについて取り上げます。

- [SSL 通信チャネル パラメータの設定](#)
- [SSH 通信チャネル パラメータの設定](#)

SSL 通信チャネル パラメータの設定

次のように Detector が Guard モジュールと相互に通信する必要がある場合は、Detector と Guard モジュールの間に SSL 通信チャネルを設定します。

- Detector がトラフィック異常を検出したときに Guard モジュールをアクティブにする。
- Guard モジュールとゾーン設定を同期化する。
- Guard モジュールをポーリングして、ゾーンに対する攻撃が終了したことを確認する。Detector で検出およびラーニング プロセスをイネーブルにした場合、Detector は、ゾーンに対する攻撃を検出すると、ラーニング プロセス (しきい値調整) を停止します。Detector は、攻撃を軽減するためにアクティブにした Guard モジュールをポーリングし、攻撃がいつ終了したかを確認します。その時点で、Detector は自動的にラーニング プロセスを再開します。

- Guard モジュールとの通信を監視し、リモート処理（Guard モジュールによるゾーン保護のアクティブ化など）が失敗した場合に通知する。

SSL 通信チャネルとは、認証とデータの暗号化を組み合わせるにより安全な接続を提供するもので、デジタル証明書、秘密鍵と公開鍵の交換ペア、および Diffie-Hellman 鍵合意パラメータによって高度なセキュリティを実現します。SSL は、指定された受信者のみがデータを解読できるようにデータを暗号化します。

Guard モジュールおよび Detector はそれぞれ、デジタル証明書を使用して、通信チャネル経由で通信を試みるデバイスを認証します。SSL 証明書にある Guard モジュールと Detector の ID は、デバイスの IP アドレスに関連付けられます。安全な接続を確保するために、Detector は秘密鍵と公開鍵のペアを生成し、公開鍵をリモート Guard リストに定義されている Guard モジュールに配布します。

Guard モジュールと Detector の両方で SSL 通信パラメータを設定した後、2 つのデバイス間で通信チャネルを確立する必要があります。これは、Detector から行います。Guard モジュールへの初期接続中、Detector は Guard モジュール上のユーザ riverhead との SSH 通信チャネルを確立し、その後これらのデバイスが通信チャネルのセキュリティ保護に必要な鍵と証明書を交換します。初期接続後、Detector は、Guard モジュールのアクティブ化、Guard モジュールのポーリング、またはゾーン設定の同期化が必要な場合に SSL 通信チャネルを確立します。

SSL 通信チャネルの片側でいずれか一方のデバイスを交換した場合、またはいずれか一方のデバイスの IP アドレスを変更した場合は、2 つのデバイスが相互の認証に成功するように、両方のデバイスで SSL 証明書を再生成する必要があります。

この項では、次のトピックについて取り上げます。

- SSL 通信チャネルのイネーブル化
- SSL 証明書の再生成

SSL 通信チャネルのイネーブル化

SSL 通信チャネルをイネーブルにするには、次の接続タイプを許可するように Guard モジュールおよび Detector を設定する必要があります。

- SSH : Detector は、SSH 通信チャネルを使用して Guard モジュールとの初期接続を確立し、鍵と証明書を交換します。
- SSL : 初期接続後、Detector は SSL 通信チャネルを使用して、Guard モジュールとのすべての接続を確立します。



注意

Guard モジュールが TACACS+ 認証を使用してユーザを認証している場合、Detector が Guard モジュールとの初期接続中に SSH 通信チャネルを確立できるようにするには、TACACS+ サーバに riverhead ユーザを定義する必要があります。

SSL 通信チャネルをイネーブルにするには、Guard モジュールおよび Detector の両方で次の手順を実行します。

ステップ 1 設定モードで `permit ssh ip-address-general [ip-mask]` を入力して、コンパニオン デバイスの IP アドレスによる SSH サービスへのアクセスを許可します。

ip-address-general 引数および *ip-mask* 引数には、コンパニオン デバイスの IP アドレスを指定します。

ステップ 2 設定モードで `service internode-comm` コマンドを入力して、通信チャネル サービスをイネーブルにします。

- ステップ 3** 設定モードで **permit internode-comm ip-address-general [ip-mask]** コマンドを入力して、コンパニオンデバイスの IP アドレスによる通信チャネル サービスへのアクセスを許可します。

ip-address-general 引数および *ip-mask* 引数には、コンパニオンデバイスの IP アドレスを指定します。

SSL 通信チャネルをイネーブルにするように Guard モジュールおよび Detector を設定した後、Guard と Detector の間で通信チャネルを確立できます。通信チャネルの確立については、『*Cisco Traffic Anomaly Detector Module Configuration Guide*』または『*Cisco Traffic Anomaly Detector Configuration Guide*』を参照してください。

SSL 証明書の再生成

SSL 証明書で Guard モジュールと Detector を識別する鍵は、IP アドレスに関連付けられます。次のタスクを実行する場合は、通信チャネルの両側で Guard モジュールと Detector の新しい SSL 証明書を再生成する必要があります。

- いずれか一方のデバイスの IP アドレスを変更する。
- いずれか一方のデバイスを交換する。

新しい SSL 証明書を再生成するプロセスには、両方のデバイスから現在の証明書を削除する作業が含まれます。現在使用している SSL 証明書を表示するには、**show internode-comm certs** コマンドを使用します。

現在使用している SSL 証明書を再生成するには、次の手順を実行します。

- ステップ 1** 設定モードで次のコマンドを入力して、Guard モジュールの SSL 証明書を Detector から削除します。

```
cert remove cert-host-ip
```

cert-host-ip 引数には、Guard モジュールの IP アドレスを指定します。すべての Guard モジュールの SSL 証明書を削除するには、アスタリスク (*) を入力します。

- ステップ 2** 設定モードで次のコマンドを入力して、Detector の SSL 証明書を Guard モジュールから削除します。

```
cert remove cert-host-ip
```

cert-host-ip 引数には、Detector の IP アドレスを指定します。Guard モジュールとの通信チャネルを確立しているすべての Detector の SSL 証明書を削除するには、アスタリスク (*) を入力します。

次の例は、SSL 証明書を削除する方法を示しています。

```
user@GUARD-conf# cert remove 10.56.36.4
```

- ステップ 3** Guard モジュールを交換する場合は、Detector から SSH ホスト鍵も削除する必要があります。設定モードで次のコマンドを使用して、Guard モジュールの SSH ホスト鍵を Detector から削除します。

```
no host-keys ip-address-general
```

ip-address-general 引数には、リモートデバイスの IP アドレスを指定します。

- ステップ 4** Detector から、Guard モジュールと Detector との間に新しい SSL 通信チャネルを確立することにより、新しい SSL 証明書を再生成します。通信チャネルの確立については、『Cisco Traffic Anomaly Detector Module Configuration Guide』または『Cisco Traffic Anomaly Detector Configuration Guide』を参照してください。

SSH 通信チャネルパラメータの設定

トラフィック異常を検出したときに Detector が Guard モジュールをアクティブにすること以外に Detector と Guard モジュールが相互に通信する必要がない場合は、Detector と Guard モジュールの間の SSH 通信チャネルを設定します。SSH 通信チャネルでは、Detector が Guard モジュールに対して次のタスクを実行できません。

- Guard モジュールとゾーン設定を同期化する。
- Guard モジュールをポーリングして、ゾーンに対する攻撃が終了したことを確認する。Detector で検出およびラーニングプロセスをイネーブルにした場合、Detector は、ゾーンに対する攻撃を検出すると、ラーニングプロセス（しきい値調整）を停止します。Detector は、Guard モジュールをポーリングできず、攻撃がいつ終了したかを確認できないため、攻撃が終了しても自動的にラーニングプロセスを再開できません。
- Guard モジュールとの通信を監視し、リモート処理（Guard モジュールによるゾーン保護のアクティブ化など）が失敗した場合に通知する。

Detector がこれらのタスクを実行できるようにするには、SSL 通信チャネルを設定する必要があります（P.4-18 の「SSL 通信チャネルパラメータの設定」を参照）。

安全な SSH 通信チャネルを確保するために、Detector は秘密 SSH 鍵と公開 SSH 鍵のペアを生成し、公開 SSH 鍵をリモート Guard リストにある Guard モジュールに配布します。

SSH 通信チャネルをイネーブルにした後、Detector と Guard モジュールの間で通信チャネルを確立する必要があります。これは、Detector から行います。

SSH 通信チャネルの片側で Guard モジュールを新しいものと交換した場合は、Detector が新しい Guard モジュールでの認証に成功するように、Detector で SSH 秘密（ホスト）鍵および SSH 公開鍵を再生成する必要があります。

この項では、次のトピックについて取り上げます。

- SSH 通信チャネルのイネーブル化
- SSH 通信チャネル鍵の再生成

SSH 通信チャネルのイネーブル化

Guard モジュールと Detector の間の SSH 通信チャネルをイネーブルにするには、**permit ssh** コマンドを入力して、Detector の IP アドレスから Guard モジュール上の SSH サービスにアクセスすることを許可します。



注意

Guard モジュールが TACACS+ 認証を使用してユーザを認証している場合、Detector が SSH 通信チャネルを確立できるようにするには、TACACS+ サーバに **riverhead** ユーザを定義する必要があります。

Guard モジュールと Detector の間の SSL 通信チャネルをイネーブルにした後、Guard と Detector の間で通信チャネルを確立できます。通信チャネルの確立については、『*Cisco Traffic Anomaly Detector Module Configuration Guide*』または『*Cisco Traffic Anomaly Detector Configuration Guide*』を参照してください。

SSH 通信チャネル鍵の再生成

Detector が SSH 通信チャネル経由で通信する Guard モジュールを新しいものと交換した場合は、次の手順を実行して、SSH 通信チャネル鍵を再生成する必要があります。

ステップ 1 Detector 上で `no host-keys ip-address-general` 設定モード コマンドを入力して、SSH ホスト鍵を Detector から削除します。

`ip-address-general` 引数には、リモート デバイスの IP アドレスを指定します。

Guard モジュールにリストされているホスト鍵を表示するには、`show host-keys` コマンドを使用します。

ステップ 2 次のいずれかの処理を実行して、リモート Guard モジュールで SSH 鍵を設定します。

- 新しい SSH 通信チャネルを Detector から確立します。
 - Detector の公開鍵をリモート Guard モジュールに手動で追加します。Detector の公開 SSH 鍵をコピーし、Guard モジュールが保持している SSH 鍵のリストにペーストすることができます。Detector の公開 SSH 鍵を表示するには、Detector 上で `show public-key` コマンドを使用します。Detector の公開 SSH 鍵を、Guard モジュールが保持している SSH 鍵のリストに追加するには、Guard モジュール上で `key add` コマンドを使用します。詳細については、[P.4-23](#) の「SSH 鍵の追加」を参照してください。
-

SSH 鍵の管理

Guard モジュールは、安全なリモート ログインのために SSH をサポートします。SSH 鍵のリストを追加すると、ログインとパスワードを入力しなくても、リモートデバイスから Guard モジュールに安全な通信ができます。

次の各項では、Guard モジュールの SSH 鍵リストの管理方法について説明します。

- [SSH 鍵の追加](#)
- [SSH 鍵の削除](#)

SSH 鍵の追加

Guard モジュールの SSH 鍵リストにリモート接続の SSH 公開鍵を追加することにより、ログイン名とパスワードを入力せずに SSH 接続をイネーブルにすることができます。

設定モードで次のコマンドを入力します。

```
key add [user-name] {ssh-dsa | ssh-rsa} key-string comment
```

表 4-11 に、key add コマンドの引数とキーワードを示します。

表 4-11 key add コマンドの引数とキーワード

パラメータ	説明
<i>user-name</i>	(オプション) SSH 鍵が追加されるユーザ名。他のユーザの SSH 鍵を追加できるのは管理者だけです。 デフォルトは、現在のユーザの SSH 鍵の追加です。
ssh-dsa	SSH2-DSA 鍵のタイプを指定します。
ssh-rsa	SSH2-RSA 鍵のタイプを指定します。
<i>key-string</i>	Cisco Traffic Anomaly Detector Module またはリモート端末で作成された公開 SSH 鍵。鍵ストリングは、8,192 ビットまでに制限されています。 鍵タイプの識別 (ssh-rsa または ssh-dsa) を除いた完全な鍵をコピーする必要があります。
<i>comment</i>	デバイスの説明。コメントの形式は、通常、鍵の生成に使用されるユーザとマシンを表す user@hostname になります。たとえば、Cisco Traffic Anomaly Detector Module で生成される SSH 公開鍵に使用されるデフォルトのコメントは、root@DETECTOR です。

次の例は、SSH RSA 鍵を追加する方法を示しています。

```
user@GUARD-conf# key add ssh-rsa 14513797528175730. . .user@Guard module.com
```

SSH 鍵の削除

リストから SSH 鍵を削除できます。SSH 鍵を削除すると、次に Guard モジュールと SSH セッションを確立するときには認証を受ける必要があります。

Guard モジュールから SSH 鍵を削除するには、設定モードで次のコマンドを使用します。

```
key remove [user-name] key-string
```

表 4-12 に、`key remove` コマンドの引数を示します。

表 4-12 `key remove` コマンドの引数

パラメータ	説明
<code>user-name</code>	(オプション) SSH 鍵が削除されるユーザ名。 他のユーザの SSH 鍵を削除できるのは管理者だけです。デフォルトは、現在のユーザの SSH 鍵の削除です。
<code>key-string</code>	削除する公開 SSH 鍵。 プロンプトに SSH 公開鍵をペーストします。識別フィールド (<code>ssh-rsa</code> または <code>ssh-dsa</code>) は除き、鍵だけをペーストしてください。

次の例は、`key remove` コマンドにカット アンド ペーストを行えるように、ユーザ鍵を表示する方法を示しています。

```
user@GUARD-conf# show keys Lilac
ssh-rsa 2352345234523456... user@Guard module.com
user@GUARD-conf# key remove Lilac 2352345234523456...
```


SFTP 接続および SCP 接続用の鍵の設定

SSH2 の最上層にある Secure File Transfer Protocol (SFTP; セキュア ファイル転送プロトコル)、および SSH に依存する Secure Copy Protocol (SCP) は、ファイルのコピー方式を提供します。このコピー方式は、安全で信頼できます。SFTP および SCP では、公開鍵による認証と強力なデータ暗号化を使用しています。したがって、ログイン、データ、およびセッションの情報が送信中に傍受されたり変更されたりすることを防止できます。

SFTP 接続および SCP 接続用の鍵を設定するには、次の手順を実行します。

ステップ 1 設定モードで **show public-key** コマンドを入力して、Guard モジュール上で Guard モジュールの公開鍵を表示します。

鍵が存在する場合は、**ステップ 2** を省略して**ステップ 3** に進みます。

鍵が存在しない場合は、**ステップ 2** に進みます。

ステップ 2 設定モードで **key generate** コマンドを入力して、Guard モジュール上で秘密および公開鍵ペアを生成します。

SSH の鍵ペアがすでに存在している場合は、次のメッセージが表示されます。

```
/root/.ssh/id_rsa already exists.  
Overwrite (y/n)?
```

y を入力して鍵を生成します。

Guard モジュールが、公開および秘密鍵ペアを作成します。Guard モジュールの公開鍵を表示するには、設定モードで **show public-key** コマンドを使用します。

ステップ 3 公開鍵を Guard モジュールからコピーし、ネットワーク サーバ上の鍵ファイル内にペーストします。

たとえば、Linux オペレーティング システムにインストールされている ネットワーク サーバに username というユーザ アカウントで接続している場合は、Guard モジュールの公開鍵を /home/username/.ssh/authorized_keys2 ファイルに追加します。

鍵は 1 行に収まるようにコピーしてください。鍵が 2 行としてコピーされた場合は、1 行目の末尾にある改行文字を削除します。



(注) 公開鍵をコピーしてネットワーク サーバ上の鍵ファイルにペーストしないと、自動エクスポート機能 (**export reports** コマンドなど) が使用できず、手動でネットワーク サーバに接続するたびに、パスワードを入力する必要があります。

ホスト名の変更

Guard モジュールのホスト名を変更できます。この変更はすぐに反映され、新しいホスト名は自動的に CLI プロンプト スtring に組み込まれます。

Guard モジュールのホスト名を変更するには、設定モードで次のコマンドを使用します。

hostname *name*

name 引数には、新しいホスト名を指定します。

次の例は、Guard モジュールのホスト名を変更する方法を示しています。

```
user@GUARD-conf# hostname CiscoGuard  
admin@CiscoGuard-conf#
```

SNMP トラップのイネーブル化

Guard モジュールが SNMP トラップを送信し、Guard モジュールで発生する重大なイベントを管理者に通知するように設定することができます。また、Guard モジュールの SNMP トラップ ジェネレータのパラメータを設定し、Guard モジュールが報告する SNMP トラップ情報の範囲を定義することもできます。

トラップのログは、Guard モジュールのイベント ログに記録され、トラップ条件が発生すると、SNMP エージェントがトラップを送信するかどうかに関係なく、イベント モニタに表示されます。

Guard モジュールが SNMP トラップを送信するように設定するには、次の手順を実行します。

- ステップ 1** 設定モードで次のコマンドを入力して、SNMP トラップ ジェネレータ サービスをイネーブルにします。

```
service snmp-trap
```

- ステップ 2** 次のコマンドを入力して、SNMP トラップ ジェネレータのパラメータ（トラップの宛先 IP アドレスとトラップ情報の範囲）を設定します。

```
snmp trap-dest ip-address [community-string [min-severity]]
```

表 4-13 に、snmp trap-dest コマンドの引数を示します。

表 4-13 snmp trap-dest コマンドの引数

パラメータ	説明
<i>ip-address</i>	宛先ホストの IP アドレス。
<i>community-string</i>	(オプション) トラップとともに送信されるコミュニティ スtring。この String は、宛先ホスト用に定義されたコミュニティ String と一致する必要があります。デフォルトのコミュニティ String は、 <i>public</i> です。1～15 文字の英数字文字列を入力します。この文字列にスペースを含めることはできません。
<i>min-severity</i>	(オプション) トラップ情報のスコープ。重大度レベルの範囲の下限を指定してスコープを定義します。この定義により、トラップは指定された重大度レベル以上のすべてのイベントを表示します。たとえば、Warnings を指定すると、トラップは Warnings から Emergencies までのすべての重大度レベルのイベントを表示します。重大度レベルのオプションを次に示します。 <ul style="list-style-type: none"> Emergencies : システムは使用不能 (重大度 = 0)。 Alerts : 即座に処置が必要 (重大度 = 1)。 Critical : 深刻な状態 (重大度 = 2)。 Errors : エラー状態 (重大度 = 3)。 Warnings : 警告状態 (重大度 = 4)。 Notifications : 正常ではあるが、重要な状態 (重大度 = 5)。 Informational : 情報通知のメッセージ (重大度 = 6)。 Debugging : デバッグ メッセージ (重大度 = 7)。 デフォルトでは、レポートにはすべての重大度レベルのイベントが表示されません。

■ SNMP トラップのイネーブル化

SNMP トラップ ジェネレータ パラメータを削除するには、**no snmp trap-dest** コマンドを使用します。すべての SNMP トラップ宛先パラメータを削除するには、アスタリスク (*) を入力します。

次の例は、errors 以上の重大度レベルのトラップが、SNMP コミュニティ スtring tempo とともに宛先 IP アドレス 192.168.100.52 に送信される例を示しています。

```
user@GUARD-conf# snmp trap-dest 192.168.100.52 tempo errors
```

表 4-14 に、Guard モジュールが生成する SNMP トラップを示します。

表 4-14 SNMP トラップ

SNMP トラップ	重大度	説明
rhExcessiveUtilizationTrap	EMERGENCY	すべての Guard モジュールゾーン上で 150,000 個以上の動的フィルタが同時にアクティブになっているため、Guard モジュールは新しい動的フィルタを追加できない。
rhExcessiveUtilizationTrap	EMERGENCY	異常検出エンジン メモリが制限値に達した (90% を超えた)。
rhGeneralTrap	EMERGENCY	Guard モジュールは、パケット アクティベーション方式によるゾーン保護のアクティブ化に失敗し、重大度が ALERT の後続トラップを送信する。
rhGeneralTrap	EMERGENCY	Guard モジュールは宛先変更のアクティブ化に失敗した。
rhGeneralTrap	EMERGENCY	Guard モジュールの宛先変更が復旧した。
rhGeneralTrap	EMERGENCY	インストールされたソフトウェア ライセンスが存在しないか形式に誤りがある。このデバイスは動作しません。
rhGeneralTrap	EMERGENCY	インストールされたソフトウェア ライセンスの有効性を確認できない。このデバイスは動作しません。
rhGeneralTrap	EMERGENCY	インストールされたソフトウェア ライセンスは明日 (<dd-mmm-yyyy>) 期限切れになる。
rhGeneralTrap	EMERGENCY	インストールされたソフトウェア ライセンスは今日期限切れになった。このデバイスは動作していません。
rhGeneralTrap	EMERGENCY	インストールされたソフトウェア ライセンスは無効:<失敗メッセージ>。
rhGeneralTrap	ALERT	Guard モジュールはパケット アクティベーション方式によるゾーン保護のアクティブ化に失敗し、重大度が EMERGENCY のトラップがすでに生成されている。
rhGeneralTrap	ALERT	ディスク スペースが 80% である。
rhGeneralTrap	ALERT	インストールされたソフトウェア ライセンスは、2～3 週間以内 (<dd-mmm-yyyy>) に期限切れになる。
rhGeneralTrap	ALERT	インストールされたソフトウェア ライセンスは、1 週間以内 (<dd-mmm-yyyy>) に期限切れになる。

表 4-14 SNMP トラップ (続き)

SNMP トラップ	重大度	説明
rhGeneralTrap	ALERT	インストールされたソフトウェア ライセンスは、30、6、5、4、3、または2日以内 (<dd-mmm-yyyy>) に期限切れになる。
rhExcessiveUtilizationTrap	CRITICAL	ギガビット インターフェイス リンクの使用率 (bps ¹ 単位) が 85% を超えている。
rhExcessiveUtilizationTrap	CRITICAL	メモリ使用率が 85% を超えている。
rhExcessiveUtilizationTrap	CRITICAL	アクセラレータ カード CPU 使用率が 85% を超えている。
rhDynamicFilterTrap	ERROR	保留中の動的フィルタ数が 1000 のため、新しい保留動的フィルタは廃棄される。
rhZoneGenericTrap	ERROR	Guard モジュールはゾーン設定の同期化に失敗した。
rhGeneralTrap	ERROR	Guard モジュールが次のようなゾーン保護のアクティブ化に失敗した。 <ul style="list-style-type: none"> 保護から、またはラーニングから保護とラーニングまで 保護とラーニングから保護またはラーニングへ Guard モジュールはゾーン保護とラーニング プロセスを非アクティブにする。
rhDynamicFilterTrap	WARNING	Guard モジュールは動的フィルタの追加に失敗した。
rhExcessiveUtilizationTrap	WARNING	Guard モジュールに、すべてのゾーン上で同時にアクティブになっている動的フィルタが 135,000 個以上ある。アクティブな動的フィルタの数が 150,000 に到達すると、Guard モジュールは新しい動的フィルタを追加できなくなります。
rhGeneralTrap	WARNING	ディスク スペースが 75% である。
rhPolicyConstructionTrap	WARNING	ラーニング プロセスのポリシー構築フェーズが失敗した。
rhProtectionTrap	WARNING	Guard モジュールはゾーン保護の開始に失敗した。
rhReloadTrap	WARNING	Guard モジュールは再起動した。トラップには、MIB2 ウォーム スタート トラップまたはコールド スタート トラップと、Guard モジュールが再起動した原因に関する情報が含まれています。
rhReloadTrap	WARNING	Guard モジュールはシャットダウンした。トラップには、MIB2 ウォーム スタート トラップまたはコールド スタート トラップと、Guard モジュールがシャットダウンした原因に関する情報が含まれています。
rhThresholdTuningTrap	WARNING	ラーニング プロセスのしきい値調整フェーズが失敗した。
rhAttackTrap	NOTIFICATIONS	攻撃が開始した。
rhAttackTrap	NOTIFICATIONS	攻撃が終了した。

表 4-14 SNMP トラップ (続き)

SNMP トラップ	重大度	説明
rhPolicyConstructionTrap	NOTIFICATIONS	ラーニング プロセスのポリシー構築フェーズが開始された。
rhPolicyConstructionTrap	NOTIFICATIONS	ラーニング プロセスのポリシー構築フェーズが受け入れられた。
rhPolicyConstructionTrap	NOTIFICATIONS	ラーニング プロセスのポリシー構築フェーズが停止された。
rhProtectionTrap	NOTIFICATIONS	ゾーン保護が開始した。
rhProtectionTrap	NOTIFICATIONS	ゾーン保護が終了した。
rhThresholdTuningTrap	NOTIFICATIONS	ラーニング プロセスのしきい値調整フェーズが開始された。
rhThresholdTuningTrap	NOTIFICATIONS	ラーニング プロセスのしきい値調整フェーズが受け入れられた。
rhThresholdTuningTrap	NOTIFICATIONS	ラーニング プロセスのしきい値調整フェーズが停止された。
rhZoneGenericTrap	NOTIFICATIONS	Guard モジュールはゾーン設定の同期化を開始した。
rhZoneTrap	NOTIFICATIONS	新しいゾーンが作成された。
rhZoneTrap	NOTIFICATIONS	ゾーンが削除された。
rhDynamicFilterControlTrap	INFO	Guard モジュールが特定のポリシーに送信しなかった攻撃検出イベントの数。
rhDynamicFilterControlTrap	INFO	Guard モジュールは、アクティブな動的フィルタが 1000 個以上あるため、削除する動的フィルタのトラップを送信しない。
rhDynamicFilterTrap	INFO	動的フィルタが追加された。
rhDynamicFilterTrap	INFO	動的フィルタが削除された。
rhDynamicFilterTrap	INFO	保留動的フィルタが追加された。

1. bps = bits per second (ビット/秒)

SNMP コミュニティ string の設定

Guard モジュールの SNMP サーバにアクセスすることにより、管理情報ベース 2 (MIB2) および Cisco Riverhead 専用 MIB で定義された情報を取得することができます。コミュニティ string はパスワードのように機能し、Guard モジュールの SNMP エージェントからの読み取りアクセスを許可します。Guard モジュールの SNMP コミュニティ string を設定して、異なる組織のクライアントがそれぞれ異なるコミュニティ string を使用して SNMP エージェントにアクセスできるようにすることができます。

SNMP コミュニティ string を追加するには、設定モードで次のコマンドを使用します。

```
snmp community community-string
```

community-string 引数には、目的の Guard モジュールのコミュニティ string を指定します。1 ~ 15 文字の英数字の文字列を入力します。この文字列にスペースを含めることはできません。Guard モジュールのデフォルトのコミュニティ string は riverhead です。コミュニティ名はいくつでも指定できます。コミュニティ string を削除するには、**no community string** コマンドを使用します。すべての SNMP コミュニティ string を削除するには、アスタリスク (*) を入力します。

次の例は、SNMP コミュニティ string を設定する方法を示しています。

```
user@GUARD-conf# snmp community tempo
```

ログインバナーの設定

ログインバナーとは、SSH セッション、コンソール ポート接続、または Guard モジュールへの WBM セッションを開いたときに、ユーザ認証の前の画面に表示されるテキストのことです。

認可されていないアクセスに対してユーザに警告したり、適切と見なされるシステムの使用法を説明したり、不適切な使用法や不正な活動を検出するためにシステムが監視されていることをユーザに警告したりするように、ログインバナーを設定できます。

Guard モジュールは、次の場所にログインバナーを表示します。

- CLI : パスワード ログインプロンプトの前、またはポップアップ ウィンドウとして (使用している SSH クライアントによって異なる)。
- WBM : Guard モジュールのログイン ウィンドウの右側。

この項では、次のトピックについて取り上げます。

- [CLI からのログインバナーの設定](#)
- [ログインバナーのインポート](#)
- [ログインバナーの削除](#)

CLI からのログインバナーの設定

login-banner コマンドを使用すると、単一または複数のメッセージ バナーを作成できます。複数のログインバナーを入力した場合、新しいログインバナーは、既存のログインバナーの最後に新しい行として追加されます。

ログインバナーを設定するには、設定モードで次のコマンドを使用します。

```
login-banner banner-str
```

banner-str 引数には、バナーのテキストを指定します。文字列の長さは最大 999 文字です。式にスペースを使用する場合は、式を引用符 (" ") で囲みます。

ログインバナーを表示するには、**show login-banner** コマンドを使用します。

次の例は、ログインバナーを設定して表示する方法を示しています。

```
user@GUARD-conf# login-banner "Welcome to the Cisco Anomaly Guard Module"
user@GUARD-conf# login-banner "Unauthorized access is prohibited."
user@GUARD-conf# login-banner "Contact sysadmin@corp.com for access."
user@GUARD-conf# show login banner
Welcome to the Cisco Anomaly Guard Module
Unauthorized access is prohibited.
Contact sysadmin@corp.com for access.
```

ログインバナーのインポート

グローバル モードまたは設定モードで次のいずれかのコマンドを入力して、ネットワーク サーバからテキスト ファイルをインポートし、既存のログインバナーを差し替えることができます。

- **copy ftp login-banner server full-file-name [login [password]]**
- **copy {sftp | scp} login-banner server full-file-name login**

インポートするファイル内の各行の最大長は、999 文字です。

SFTP および SCP は、安全な通信を行うために SSH を使用します。したがって、**sftp** オプションまたは **scp** オプションを指定して **copy** コマンドを入力したときに、Guard モジュールで使用される鍵が設定されていないと、Guard モジュールからパスワードの入力を求められます。Guard モジュール

ルが安全な通信のために使用する鍵を設定する方法の詳細については、P.4-25 の「SFTP 接続および SCP 接続用の鍵の設定」を参照してください。

表 4-15 に、`copy login-banner` コマンドの引数とキーワードを示します。

表 4-15 `copy login-banner` コマンドの引数とキーワード

パラメータ	説明
<code>ftp</code>	FTP を指定します。
<code>sftp</code>	SFTP を指定します。
<code>scp</code>	SCP を指定します。
<code>server</code>	ログインバナーファイルが存在するネットワークサーバの IP アドレス。IP アドレスをドット区切り 10 進表記で入力します (たとえば 192.168.10.2)。
<code>full-file-name</code>	ファイルの完全な名前。パスを指定しない場合、サーバはユーザのホームディレクトリからファイルをコピーします。
<code>login</code>	(オプション) サーバのログイン名。 <code>login</code> 引数は、FTP サーバを定義するときは省略可能です。ログイン名を入力しない場合、FTP サーバは匿名ログインであると想定し、パスワードを要求しません。
<code>password</code>	(オプション) リモート FTP サーバへのアクセスに必要なパスワード。パスワードを入力しない場合、Guard モジュールによってパスワードを要求されます。

次の例は、FTP サーバからログインバナーをインポートする方法を示しています。

```
user@GUARD-conf# copy ftp login-banner 10.0.0.191 /root/login-banner <user> <password>
```

ログインバナーの削除

ユーザ認証の前にメッセージを表示する必要がなくなった場合、ログインバナーを削除できます。

ログインバナーを削除するには、設定モードで `no login-banner` コマンドを使用します。

次の例は、ログインバナーを削除する方法を示しています。

```
user@GUARD-conf# no login-banner
```

Web-Based Manager ログの設定

エンドユーザ インターフェイスをカスタマイズするために、企業のロゴやカスタマイズされたロゴを Web-based Manager (WBM) Web ページに追加することができます。

新しいロゴは、次の場所に表示されます。

- Guard モジュールのログイン ページで、Cisco Systems ログの下。
- すべての WBM ページ (Guard モジュールのログイン ページは除く) で、Cisco Systems ログの右側。

新しいロゴは GIF 形式である必要があります。新しいロゴのサイズは、幅 = 87 ピクセル、高さ = 41 ピクセルにすることをお勧めします。

この項では、次のトピックについて取り上げます。

- [WBM ログのインポート](#)
- [WBM ログの削除](#)

WBM ログのインポート

WBM で使用するネットワーク サーバから新しいロゴをインポートするには、グローバル モードまたは設定モードで次のコマンドを入力します。

- `copy ftp wbm-logo server full-file-name [login [password]]`
- `copy {sftp | scp} wbm-logo server full-file-name login`

SFTP および SCP は安全な通信を行うために SSH を使用します。したがって、`sftp` オプションまたは `scp` オプションを指定して `copy` コマンドを入力したときに、Guard モジュールで使用される鍵が設定されていないと、Guard モジュールからパスワードの入力を求められます。Guard モジュールが安全な通信のために使用する鍵を設定する方法の詳細については、[P.4-25](#) の「[SFTP 接続および SCP 接続用の鍵の設定](#)」を参照してください。

[表 4-16](#) に、`copy wbm-logo` コマンドの引数とキーワードを示します。

表 4-16 `copy wbm-logo` コマンドの引数とキーワード

パラメータ	説明
<code>ftp</code>	FTP を指定します。
<code>sftp</code>	SFTP を指定します。
<code>scp</code>	SCP を指定します。
<code>server</code>	WBM ログファイルが存在するネットワーク サーバの IP アドレス。IP アドレスをドット区切り 10 進表記で入力します (たとえば 192.168.10.2)。
<code>full-file-name</code>	GIF ファイル拡張子を含む、完全なファイル名。パスを指定しない場合、サーバはユーザのホーム ディレクトリからファイルをコピーします。
<code>login</code>	(オプション) サーバのログイン名。 <code>login</code> 引数は、FTP サーバを定義するときは省略可能です。ログイン名を入力しない場合、FTP サーバは匿名ログインであると想定し、パスワードを要求しません。
<code>password</code>	(オプション) リモート FTP サーバのパスワード。パスワードを入力しない場合、Guard モジュールによってパスワードを要求されます。

次の例は、FTP サーバから WBM ログ ファイルをインポートする方法を示しています。

```
user@GUARD-conf# copy ftp wbm-logo 10.0.0.191 /root/WBMlogo.gif <user> <password>
```

WBM ログの削除

WBM ログを削除するには、設定モードで **no wbm-logo** コマンドを使用します。

次の例は、WBM ログを削除する方法を示しています。

```
user@GUARD-conf# no wbm-logo
```

セッションタイムアウトの設定

セッションタイムアウトとは、アクティビティが何もない状態でセッションがアクティブでいられる時間のことです。設定された時間内に何もアクティビティがなかった場合、タイムアウトが発生し、再びログインする必要があります。セッションタイムアウトは、デフォルトではディセーブルになっています。

セッションタイムアウトは CLI にのみ適用され、WBM には適用されません。

設定モードで次のコマンドを入力して、Guard モジュールがアイドルセッションを自動的に切断するまでの分数を設定できます。

session-timeout *timeout-val*

timeout-val 引数には、Guard モジュールが自動的にアイドルセッションを切断するまでの分数を指定します。有効な値は、1～1,440 分（1 日）です。

次の例は、Guard モジュールが 10 分後にアイドルセッションを切断するように設定する方法を示しています。

```
user@GUARD-conf# session-timeout 10
```

Guard モジュールが自動的にアイドルセッションを切断しないようにするには、**no session-timeout** コマンドを使用します。

セッションタイムアウトの値を表示するには、**show session-timeout** コマンドを使用します。

