



# 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](#)
- [Cisco Traffic Anomaly Detector Module との通信の確立](#)
- [日付と時刻の設定](#)
- [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](#)
- [WBM ロゴの設定](#)
- [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 モジュールでアクティブにするサービスを定義することができます。正しい機能をイネーブルにするためには、サービスをイネーブルにして、そのサービスへのアクセスを許可する必要があります。Guard モジュールのサービスのアクティベーションを制御し、特定の IP アドレスに対してアクセス権を付与または拒否することにより、Guard モジュールにアクセスして制御する IP アドレスを制限することができます。

表 4-1 で、Guard モジュールのサービスについて説明します。

表 4-1 Guard モジュール サービス

サービス	説明
<b>internode-comm</b>	ノード間通信サービス。Guard モジュールは、Cisco Traffic Anomaly Detector Module との通信チャネルを確立するときにこのサービスを使用します。  詳細については、P.4-27 の「Cisco Traffic Anomaly Detector Module との通信の確立」を参照してください。
<b>snmp-server</b>	SNMP 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、Riverhead の専用 MIB、MIB2、および UC Davis MIB で定義された情報を取得することができます。  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。   <b>(注)</b> 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。
<b>snmp-trap</b>	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.4-40 の「SNMP トラップのイネーブル化」を参照してください。

表 4-1 Guard モジュール サービス (続き)

サービス	説明
ssh	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、 <a href="#">P.3-23 の「SSH を使用した Guard モジュールへのアクセス」</a> および <a href="#">P.4-35 の「SSH 鍵の管理」</a> を参照してください。
wbm	Web-Based Management (WBM) サービス。Web ブラウザを使用して Web から Guard モジュールを制御できます。詳細については、 <a href="#">P.3-21 の「Web-Based Manager による Guard モジュールの管理」</a> を参照してください。

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 | snmp-server | snmp-trap | wbm}
```

Guard モジュールのサービスの説明については、[表 4-1](#)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。

- ステップ 2** 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 | snmp-server | snmp-trap | ssh | wbm}  
ip-address-general [ip-mask]
```

[表 4-2](#)に、`permit` コマンドの引数を示します。

表 4-2 permit コマンドの引数

パラメータ	説明
<i>service</i>	アクセスと操作の対象となるサービス。Guard モジュールのサービスの説明については、表 4-1 を参照してください。
<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)。すべての IP アドレスからのアクセスを許可するには、アスタリスク (*) を使用します。
<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。

**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 (\* と入力する) ことはお勧めしません。

次の例は、サービスをアクティブ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、誰に対して Guard モジュールへのアクセスを許可するか、またアクセス後にどのサービスを許可するか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。
- **アカウントング**: 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、Cisco Traffic Anomaly Detector Module 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。

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。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。

次の各項では、アクセス コントロールの設定方法について説明します。

- [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名およびイネーブル パスワードが認証に使用されます。この認証方式がデフォルトです。詳細については、P.4-8 の「ローカル認証の設定」を参照してください。
- TACACS+ 認証：1つの TACACS+ サーバまたは複数の TACACS+ サーバのリストを使用してユーザが認証されます。

ユーザを1つの TACACS+ サーバのみに設定する場合、その TACACS+ サーバでユーザに対して認可も設定する必要があります。設定しないと、そのユーザは `show` コマンドにしかアクセスできません。

シーケンシャルな認証リストを設定することができます。認証リストとは、ユーザ認証に使用する認証方式を定義したもので、1つ以上の認証方式を指定でき、最初の認証方式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、この順次認証リスト上の最初の認証方式を使用してユーザを認証します。認証方式からの応答がない場合は、Guard モジュールはリスト上の2番目の認証方式を使用します。認可方式が両方とも成功しなかった場合のみ、ユーザ認証は失敗します。

分散認証方式を設定し、ユーザを別々の認証データベースに定義することができます。Guard モジュールは、最初の TACACS+ サーバを使用してユーザを認証します。認証で拒否が返された場合、Guard モジュールは TACACS+ サーバリスト、および、存在する場合は代替の認証方式 (ローカル) をスキャンします。リスト上のすべての認証方式が失敗した場合にのみ、認証が失敗します。このオプションは、`first-hit` オプションを設定していない場合にのみ有効です。



(注)

ユーザ データベースが、複数の TACACS+ サーバに分散している、または1つの TACACS+ サーバとローカル ユーザ データベースに分散している場合は、`no tacacs-server first-hit` コマンドを使用してください。これを使用しないと、最初の認証方式から拒否が返された後、認証が失敗します。

この項では、次のトピックについて取り上げます。

- [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.4-21](#) 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
<code>enable</code>	Guard モジュールは、上位の特権レベルに入るときに認証を行います。
<code>login</code>	Guard モジュールへのログイン時に認証が行われます。
<code>local</code>	Guard モジュールは、ローカル データベースを使用してユーザを認証します。
<code>tacacs+</code>	TACACS+ サーバによってユーザが認証されます。
<code>tacacs+   local</code>	(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設定します。

## ■ AAA を使用したアクセス コントロールの設定

コンソール セッションから Guard モジュールにアクセスした場合、Guard モジュールは、定義済みの認証方式に関係なく、ローカル ユーザ データベースを使用して認証します。

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、管理者特権を持つユーザ名があらかじめ設定されています。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.4-6](#) 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>	ユーザ名。1～63 文字の英数字の文字列です。大文字と小文字が区別され、先頭は英字である必要があります。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。
<b>admin</b>	すべての操作にアクセスできます。
<b>config</b>	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。
<b>dynamic</b>	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。 <b>dynamic</b> 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。
<b>show</b>	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) パスワード。6～24 文字の文字列を入力します。スペースは使用できず、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、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 1234
```

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.4-10 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。新しいパスワードをもう一度入力し、確認するように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。

特定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を持つユーザです。

**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。新しいパスワードをもう一度入力し、確認するように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せん。

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：TACACS+ サーバを使用してユーザが認可されます。後続のサーバが定義されている場合は、1つのサーバとの通信が失敗した場合にのみ、その後続のサーバへのアクセスが開始されます。

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。これは、**copy running-config** コマンドを使用すると、設定ファイル内で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の実行が許可されるためです。

- 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 プロファイルが使用されます。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 モジュールのローカル認可は、TACACS+ サーバへの通信に失敗した場合に実行することができます。

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。

Guard モジュールが認証拒否を最終的なものと見なし、それ以上他の TACACS+ サーバやローカル ユーザ データベースを検索しないように設定するために、TACACS+ サーバ アトリビュートを設定できます。詳細については、[P.4-21](#) の「[TACACS+ サーバ アトリビュートの設定](#)」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。

- [ローカル認可の設定](#)
- [認可方式の設定](#)
- [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 のサービス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サービス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、要求したサービス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](#)

## ■ AAA を使用したアクセスコントロールの設定

## 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。他のユーザ特権レベルのパスワードを設定する前に、そのユーザ特権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` コマンドの引数を示します。

表 4-5 enable password コマンドの引数

パラメータ	説明
<code>level level</code>	<p>(オプション) ユーザの特権レベル。特権レベルは次のいずれかになります。</p> <ul style="list-style-type: none"> <li>• <b>admin</b> : すべての操作にアクセスできる。</li> <li>• <b>config</b> : 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。</li> <li>• <b>dynamic</b> : 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。<b>dynamic</b> 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。</li> <li>• <b>show</b> : 監視操作と診断操作にアクセスできる。</li> </ul> <p>デフォルトのレベルは <b>admin</b> です。</p>
<code>password</code>	<p>(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。</p>

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 <password>
```

## ユーザ特権レベル間の移動

認可されたユーザは、ユーザ特権レベル間を移動することができます。

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]
```

`level` 引数には、ユーザの特権レベルを指定します。特権レベルは次のいずれかになります。

- **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。

**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>
```

下位の特権レベル (`show`) に戻る場合は、**disable** コマンドを使用します。

## ■ AAA を使用したアクセスコントロールの設定

## 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.4-21 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 {local | tacacs+} [local]`

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 4-6 に、`aaa authorization` コマンドの引数とキーワードを示します。

表 4-6 `aaa authorization` コマンドの引数とキーワード


パラメータ	説明
<code>exec</code>	<p>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。</p> <p> <b>注意</b> 認可を設定する前に、TACACS+ サーバにそのユーザを設定しておく必要があります。これを行わないと、Guard モジュールにアクセスできない場合があります。</p>
<code>commands</code>	<p>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。</p>



表 4-6 aaa authorization コマンドの引数とキーワード (続き)

パラメータ	説明
<i>level</i>	指定された特権レベルの認可を定義します。特権レベルは次のいずれかになります。 <ul style="list-style-type: none"> <li>• <b>admin</b> : すべての操作にアクセスできる。</li> <li>• <b>config</b> : 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。</li> <li>• <b>dynamic</b> : 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。<b>dynamic</b> 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。</li> </ul>
<b>local</b>	Guard モジュールのローカル データベースでユーザのアクセス権を確認します。
<b>tacacs+</b>	TACACS+ サーバでユーザのアクセス権を確認します。
<b>local</b>	(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設定します。

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

ゾーン設定へのアクセスを認可されたユーザのみに制限するには、ゾーン名のタブ補完をディセーブルにします。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.4-21](#) の「[TACACS+ サーバアトリビュートの設定](#)」を参照してください。

## ■ AAA を使用したアクセスコントロールの設定

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

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

表 4-7 に、**aaa accounting** コマンドのキーワードを示します。

**表 4-7 aaa accounting コマンドのキーワード**

パラメータ	説明
<b>show   dynamic   config   admin</b>	指定された特権レベルのアカウントिंग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)。
<b>stop-only</b>	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。
<b>tacacs+</b>	アカウントING情報の記録に TACACS+ サーバのデータベースを使用します。
<b>local</b>	アカウントING情報を保存しません。

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

パフォーマンス影響する可能性があるため、アカウントING管理は **config** ユーザ特権レベルにのみイネーブルにすることをお勧めします。

特権レベルのアカウントING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ING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、またはアカウントिंग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` 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.4-22](#) 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.4-23](#) 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.4-24](#) 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.4-25](#) 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。

## ■ AAA を使用したアクセスコントロールの設定

**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.4-26](#) 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

## 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、およびアカウントリングに使用するように設定できます。Guard モジュールは、リストされた TACACS+ サーバを使用して、ユーザを認証または認可したり、アカウントリング イベントを送信したりします。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.4-24](#) 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
```

*ip-address* 引数には、TACACS+ サーバの IP アドレスを指定します。

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
```

## TACACS+ サーバの暗号鍵の設定

TACACS+ サーバにアクセスするには、暗号鍵を設定する必要があります。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、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

Guard モジュールが認証拒否を最終的なものと見なし、それ以上他の TACACS+ サーバやローカル ユーザ データベースを検索しないように設定するには、**tacacs-server first-hit** コマンドを使用します。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的なものと見なし、他の TACACS+ サーバまたはローカル ユーザ データベースを使用したそのユーザ認証の試行を停止します。

**tacacs-server first-hit** コマンドを入力して認証拒否を最終的なものと見なすように TACACS+ 検索方式を設定していない場合、Guard モジュールはデフォルトでリスト内のすべての TACACS+ サーバを使用してユーザ認証を試みます。ユーザ認証として **first-hit** 検索方式をイネーブにする (**no tacacs-server first-hit** コマンドを入力する) と、Guard モジュールは、リストの最初にある TACACS+ サーバを使用してユーザを認証します。最初のサーバが応答しなかった、またはユーザの認証に失敗した場合は、Guard モジュールはリストにある次のサーバを選択します。リストにあるすべての TACACS+ サーバが応答しなかった場合、またはユーザ認証に失敗した場合、ローカルの認証方式が設定されていないと、そのユーザ認証は失敗します。TACACS+ 検索方式は認証のみに適用できるもので、認可やアカウントिंगには影響しません。

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認証するように設定するには、設定モードで **tacacs-server first-hit** コマンドを使用します。

**first-hit** 検索方式をディセーブルにし、Guard モジュールがリスト内のすべての TACACS+ サーバを使用してユーザ認証を試みるようにするには、設定モードで **no tacacs-server first-hit** コマンドを使用します。

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-config# 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

TACACS+ サーバの統計情報を表示できます。Guard モジュールは、サーバごとに統計データを提供します。

TACACS+ 関連の統計情報を表示するには、設定モードで **show tacacs statistics** コマンドを使用します。

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 4-8 に、**show tacacs statistics** コマンド出力のフィールドを示します。

表 4-8 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、アクセス権を拒否された回数。
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。

## Cisco Traffic Anomaly Detector Module との通信の確立

次のタスクをイネーブルにする場合は、Guard モジュールと Cisco Traffic Anomaly Detector Module (Detector モジュール) の間に安全な通信チャネルを確立できません。

- Remote activation of zone protection : Detector モジュールはゾーン 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、Guard モジュールによるゾーン保護をアクティブにします。
- Synchronization of zone configuration information : Detector モジュールと Guard モジュールは、通信チャネルを介してゾーン設定情報を交換します。

Detector モジュールは Guard モジュールとの接続を確立し、通信チャネルの確保に必要な暗号鍵と証明書を交換します。その後 Detector モジュールは接続を閉じ、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。

Guard モジュールは、次に示す2つのタイプの通信チャネルをサポートします。

- Secure Sockets Layer (SSL) : Remote activation of zone protection および Synchronization of zone configuration information をイネーブルにします。
- セキュア シェル 2 (SSH2) : Remote activation of zone protection のみをイネーブルにします。

Detector モジュールは Guard モジュールのリスト (リモート Guard リストと呼ばれる) を保持し、このリスト上の Guard モジュールのゾーン保護をアクティブにしたり、これらの Guard とゾーン設定を同期させたりします。Detector モジュールは、リモート Guard リストに設定されている各 Guard モジュールと通信チャネルを確立します。Detector モジュールは、SSL 通信チャネルを確立する前に各 Guard モジュールと SSH2 通信チャネルを確立します。SSL リモート Guard リストおよび SSH リモート Guard リストを設定する場合は、SSH2 通信チャネルを確立する必要はありません。

この項では、次のトピックについて取り上げます。

- [SSL 通信チャネルの設定](#)
- [SSH 通信チャネルの設定](#)

## SSL 通信チャネルの設定

Guard モジュールと Detector モジュールは、通信チャネルに Secure Sockets Layer (SSL) 接続を使用します。SSL とは、認証とデータの暗号化を組み合わせることにより安全な接続を提供するもので、デジタル証明書、秘密と公開の鍵交換ペア、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。

各 Guard モジュールおよび各 Detector モジュールは、通信チャネル経由で通信を試みるデバイスに対し、デジタル証明書やそのデバイス固有の情報（デバイスの IP アドレスなど）を使用して認証を行います。

安全な接続を確保するために、Detector モジュールは秘密および公開鍵ペアを生成し、公開鍵をリモート Guard リスト内の Guard モジュールに配布します。

Guard モジュール上で通信チャネルサービスをイネーブルにしたら、Detector モジュールから通信チャネルを確立します。Detector モジュールはまず、Guard モジュール上のユーザ *riverhead* と SSH2 通信チャネルを確立します。次に Detector モジュールは、安全な SSH2 通信チャネルを使用して SSL 接続鍵を交換します。

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、Guard モジュールおよび Detector モジュールの両方で次の手順を実行します。



### 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 モジュールが SSH2 接続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。

**ステップ 1** 設定モードで **permit ssh ip-address-general [ip-mask]** を入力して、Detector モジュールの IP アドレスから Guard モジュール上の SSH サービスへのアクセスを許可します。

引数 *ip-address-general* および *ip-mask* で、Guard モジュールへのアクセスを許可する Detector モジュールの IP アドレスを定義します。



**(注)** SSH サービスはすでにイネーブルになっているので、ここでイネーブルにする必要はありません。

**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、Detector モジュールの IP アドレスから通信チャネル サービスへのアクセスを許可します。

引数 *ip-address-general* および *ip-mask* で、Guard モジュールへのアクセスを許可する Detector モジュールの IP アドレスを定義します。



**(注)** SSL 証明書にある Guard モジュールと Detector モジュールの ID は、IP アドレスに関連付けられます。通信チャネルの片側で Guard モジュールまたは Detector モジュールの IP アドレスを変更する場合は、SSL 証明書を再生成する必要があります。詳細については、P.4-30 の「[SSL 証明書の再生成](#)」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 モジュールを識別する鍵は、IP アドレスに関連付けられます。

次のような場合、通信チャネルの両側で Guard モジュールと Detector モジュールの新しい 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 証明書を生成する前に、まず、現在使用している証明書を両方のデバイスで削除する必要があります。

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 モジュールから削除します。

```
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 モジュールの 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 モジュールの 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 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。

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 モジュールから 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 モジュールから削除します。

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

*ip-address* 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** 新しい SSL 証明書を再生成するには、Guard モジュールと Detector モジュールとの間に新しい SSL 通信チャネルを確立する必要があります。この処理は Detector モジュールから実行する必要があります。

## SSH 通信チャネルの設定

トラフィックの異常を検出すると、Detector モジュールはそのイベントをログに記録するか、SSH 通信チャネルを使用して Guard モジュールによるゾーン保護をアクティブにします。SSH 通信チャネルを使用する場合、Detector モジュールでは次のタスクを実行できません。

- ゾーン設定を同期化する。
- Guard モジュールを監視して、ゾーンに対する攻撃が終了したことを確認する。異常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 モジュールはゾーンに対する攻撃が終了したことを確認できず、リモート Guard をアクティブにした後は、ゾーン トラフィックのラーニングを続行しません。
- 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。

安全な SSH2 通信チャネルを確保するために、Detector モジュールは秘密および公開 SSH 鍵ペアを生成し、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ら、Detector モジュールから SSH 通信チャネルを確立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。

- [SSH2 通信チャネルのイネーブル化](#)
- [SSH2 通信チャネル鍵の再生成](#)

## SSH2 通信チャネルのイネーブル化

Guard モジュールと Detector モジュールの間の SSH2 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 モジュールの IP アドレスからの Guard モジュール上の SSH サービスへのアクセスを許可します。



### 注意

Guard が TACACS+ 認証を使用してユーザを認証している場合、Detector モジュールが SSH2 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。

SSH 通信チャネルをイネーブルにしたら、Detector モジュールから SSH 通信チャネルを確立する必要があります。

Guard モジュールデバイスを交換 (スワップアウト) する場合は、SSH2 通信チャネルを再生成する必要があります。P.4-32 の「[SSH2 通信チャネル鍵の再生成](#)」を参照してください。

## SSH2 通信チャネル鍵の再生成

Guard モジュールデバイスを交換した場合は、次の手順を実行して通信チャネルを再生成します。

- ステップ 1** Detector モジュール上で **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 モジュールから削除します。

*ip-address* 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 で SSH 鍵を設定します。

- 新しい SSH2 通信チャネルを Detector モジュールから確立します。
- Detector モジュールの公開鍵をリモート Guard に手動で追加します。Detector モジュールの公開 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 モジュールの公開 SSH 鍵を表示するには、Detector モジュール上で **show public-key** コマンドを使用します。

Detector モジュールの公開 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.4-35](#) の「SSH 鍵の追加」を参照してください。

---

## 日付と時刻の設定

時刻と日付を設定するには、設定モードで次のコマンドを使用します。

```
date MMDDhhmm[[CC]YY][.ss]
```

表 4-9 に、**date** コマンドの引数を示します。

**表 4-9** date コマンドの引数

パラメータ	説明
<i>MM</i>	数字で表した月。
<i>DD</i>	月の日付。
<i>hh</i>	24 時間表記の時間。
<i>mm</i>	分。
<i>CC</i>	(オプション) 年の最初の 2 桁 (たとえば <b>2005</b> )。
<i>YY</i>	(オプション) 年の最後の 2 桁 (たとえば <b>2005</b> )。
<i>.ss</i>	(オプション) 秒 (小数点が必要)。

次の例は、日付を 2003 年 10 月 8 日に、時刻を午後 5 時 10 分 (1710) 17 秒に設定する方法を示しています。

```
user@GUARD-conf# date 1008171003.17
Wed Oct 8 17:10:17 EDT 2003
```

## 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

ログイン名とパスワードを入力しない SSH 接続をイネーブルにするには、Guard モジュールの SSH 鍵リストにリモート接続の SSH 公開鍵を追加します。

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 4-10 に、`key add` コマンドの引数とキーワードを示します。

表 4-10 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) 指定されたユーザの SSH 鍵を追加します。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、現行ユーザの SSH 鍵の追加です。
<b>ssh-dsa</b>	SSH2-DSA 鍵のタイプ。
<b>ssh-rsa</b>	SSH2-RSA 鍵のタイプ。
<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。

表 4-10 key add コマンドの引数とキーワード (続き)

パラメータ	説明
<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 <code>user@hostname</code> 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、 <code>root@DETECTOR</code> 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 4-11 に、`key remove` コマンドの引数を示します。

表 4-11 key remove コマンドの引数

パラメータ	説明
<i>user-name</i>	(オプション) 指定されたユーザの SSH 鍵を削除します。  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、現行ユーザの SSH 鍵の削除です。
<i>key-string</i>	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 の最上層にあるセキュア FTP (SFTP)、および SSH に依存する Secure Copy (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。これにより、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。

SSH2 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 プロンプト文字列に組み込まれます。

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 4-12 に、`snmp trap-dest` コマンドの引数を示します。

**表 4-12 snmp trap-dest コマンドの引数**

パラメータ	説明
<code>ip-address</code>	宛先ホストの IP アドレス。
<code>community-string</code>	(オプション) トラップとともに送信されるコミュニティ ストリング。このストリングは、宛先ホスト用に定義されたコミュニティ ストリングと一致する必要があります。デフォルトのコミュニティ ストリングは、 <code>public</code> 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。



表 4-12 snmp trap-dest コマンドの引数 (続き)

パラメータ	説明
<i>min-severity</i>	<p>(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。</p> <ul style="list-style-type: none"> <li>• Emergencies : システムは使用不能 (重大度 = 0)。</li> <li>• Alerts : 即座に処置が必要 (重大度 = 1)。</li> <li>• Critical : 深刻な状態 (重大度 = 2)。</li> <li>• Errors : エラー状態 (重大度 = 3)。</li> <li>• Warnings : 警告状態 (重大度 = 4)。</li> <li>• Notifications : 正常ではあるが、重要な状態 (重大度 = 5)。</li> <li>• Informational : 情報通知のメッセージ (重大度 = 6)。</li> <li>• Debugging : デバッグ メッセージ (重大度 = 7)。</li> </ul> <p>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す。</p>

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 4-13 に、Guard モジュールが生成する SNMP トラップを示します。

表 4-13 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	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。
rhExcessiveUtilizationTrap	CRITICAL	ギガビット インターフェイスリンクの使用率 (bps <sup>1</sup> 単位) が 85% を超えている。

表 4-13 SNMP トラップ (続き)

SNMP トラップ	重大度	説明
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"> <li>保護またはラーニングから保護とラーニングまで</li> <li>保護とラーニングから保護またはラーニングへ</li> </ul> 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化する。
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。

表 4-13 SNMP トラップ (続き)

SNMP トラップ	重大度	説明
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。
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。

表 4-13 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 スtring 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 スtring 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 スtring 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 スtring 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 スtring 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 スtring を指定します。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。Guard モジュールのデフォルトのコミュニティ スtring 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 スtring 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 スtring 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 スtring 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。

## ■ ログイン バナーの設定

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。

セキュア FTP (SFTP) および Secure Copy (SCP) は安全な通信を SSH に依存しています。よって、**copy** コマンドに **sftp** または **scp** オプションを指定して入力する前に、Guard モジュールが使用する鍵を設定していない場合、Guard モジュールからパスワードを入力するよう要求されます。Guard モジュール がセキュア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.4-38](#) 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 4-14](#) に、**copy login-banner** コマンドの引数とキーワードを示します。

**表 4-14** copy login-banner コマンドの引数とキーワード

パラメータ	説明
<b>ftp</b>	Guard モジュールは FTP ネットワーク サーバからログイン バナー ファイルをインポートします。
<b>sftp</b>	Guard モジュールは SFTP ネットワーク サーバからログイン バナー ファイルをインポートします。



表 4-14 copy login-banner コマンドの引数とキーワード (続き)

パラメータ	説明
<code>scp</code>	Guard モジュールは SCP ネットワーク サーバからログイン バナー ファイルをインポートします。
<code>server</code>	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>	サーバのログイン名。  <i>login</i> 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かった場合、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
```

## WBM ロゴの設定

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（またはカスタマイズされたロゴ）を 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 ロゴの下。
- 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 ロゴの右側。

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 ロゴのインポート](#)
- [WBM ロゴの削除](#)

## WBM ロゴのインポート

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**

セキュア FTP (SFTP) および Secure Copy (SCP) は安全な通信を SSH に依存しているため、**copy** コマンドに **sftp** または **scp** オプションを指定して入力する前に、Guard モジュールが使用する鍵を設定していない場合、Guard モジュールからパスワードを入力するよう要求されます。Guard モジュールがセキュア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.4-38](#) 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 wbm-logo` コマンドの引数とキーワードを示します。

表 4-15 `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>	Guard モジュールは FTP ネットワーク サーバから WBM ロゴ ファイルをインポートします。
<code>sftp</code>	Guard モジュールは SFTP ネットワーク サーバから WBM ロゴ ファイルをインポートします。
<code>scp</code>	Guard モジュールは SCP ネットワーク サーバから WBM ロゴ ファイルをインポートします。
<code>server</code>	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>	サーバのログイン名。  <i>login</i> 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かった場合、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 ロゴ 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 ロゴの削除

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

次の例は、WBM ロゴ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接続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。