



## ユーザ管理

---

- ユーザアカウント (1 ページ)
- ユーザ名に関するガイドライン (3 ページ)
- パスワードに関するガイドライン (3 ページ)
- リモート認証のガイドライン (4 ページ)
- ユーザの役割 (6 ページ)
- ローカル認証されたユーザのパスワードプロファイル (7 ページ)
- デフォルト認証サービスの選択 (8 ページ)
- セッションタイムアウトの設定 (10 ページ)
- 絶対セッションタイムアウトの設定 (11 ページ)
- リモートユーザのロールポリシーの設定 (12 ページ)
- ローカル認証されたユーザのパスワードの強度チェックの有効化 (13 ページ)
- ログイン試行の最大回数の設定 (14 ページ)
- ユーザロックアウトステータスの表示およびクリア (15 ページ)
- 変更間隔のパスワード変更の最大数の設定 (15 ページ)
- 最小パスワード長チェックの設定 (16 ページ)
- パスワードの変更禁止間隔の設定 (17 ページ)
- パスワード履歴カウントの設定 (18 ページ)
- ローカルユーザアカウントの作成 (19 ページ)
- ローカルユーザアカウントの削除 (21 ページ)
- ローカルユーザアカウントのアクティブ化または非アクティブ化 (22 ページ)
- ローカル認証されたユーザのパスワード履歴のクリア (23 ページ)

## ユーザアカウント

ユーザアカウントは、システムにアクセスするために使用されます。最大 48 のローカルユーザアカウントを設定できます。各ユーザアカウントには、一意のユーザ名とパスワードが必要です。

## 管理者アカウント

管理者アカウントはデフォルトユーザアカウントであり、変更や削除はできません。このアカウントは、システム管理者またはスーパーユーザアカウントであり、すべての権限が与えられています。管理者アカウントには、デフォルトのパスワードは割り当てられません。初期システムセットアップ時にパスワードを選択する必要があります。

管理者アカウントは常にアクティブで、有効期限がありません。管理者アカウントを非アクティブに設定することはできません。

## ローカル認証されたユーザアカウント

ローカル認証されたユーザアカウントは、シャージを通じて直接認証され、管理者権限またはAAA権限があれば誰でも有効化または無効化できます。ローカルユーザアカウントを無効にすると、ユーザはログインできません。データベースは無効化されたローカルユーザアカウントの設定の詳細を削除しません。無効なローカルユーザアカウントを再度有効にすると、アカウントは既存の設定で再びアクティブになりますが、。

## リモート認証されたユーザアカウント

リモート認証されたユーザアカウントとは、LDAP、RADIUS、またはTACACS+を通じて認証されたユーザアカウントのことです。

ユーザがローカルユーザアカウントとリモートユーザアカウントを同時に保持する場合、ローカルユーザアカウントで定義されたロールがリモートユーザアカウントに保持された値を上書きします。

リモート認証のガイドラインの詳細や、リモート認証プロバイダーの設定および削除方法については、次のトピックを参照してください。

- [リモート認証のガイドライン \(4 ページ\)](#)
- [LDAP プロバイダーの設定](#)
- [RADIUS プロバイダーの設定](#)
- [TACACS+ プロバイダーの設定](#)

## ユーザアカウントの有効期限

ユーザアカウントは、事前に定義した時間に有効期限が切れるように設定できます。有効期限の時間になると、ユーザアカウントは無効になります。

デフォルトでは、ユーザアカウントの有効期限はありません。

ユーザアカウントに有効期限を設定した後、「有効期限なし」に再設定することはできません。ただし、使用できる最新の有効期限日付でアカウントを設定することは可能です。

## ユーザ名に関するガイドライン

ユーザ名は、Firepower Chassis Manager および FXOS CLI のログイン ID としても使用されます。ユーザアカウントにログイン ID を割り当てるときは、次のガイドラインおよび制約事項を考慮してください。

- ログイン ID には、次を含む 1 ～ 32 の文字を含めることができます。
  - 任意の英字
  - 任意の数字
  - \_ (アンダースコア)
  - - (ダッシュ)
  - . (ドット)
- ログイン ID は一意である必要があります。
- ログイン ID は、英文字で開始する必要があります。数字やアンダースコアなどの特殊文字から始めることはできません。
- ログイン ID では、大文字と小文字が区別されます。
- すべて数字のログイン ID は作成できません。
- ユーザアカウントの作成後は、ログイン ID を変更できません。ユーザアカウントを削除し、新しいユーザアカウントを作成する必要があります。

## パスワードに関するガイドライン

ローカル認証された各ユーザアカウントにパスワードが必要です。admin または AAA 権限を持つユーザについては、ユーザパスワードのパスワード強度チェックを実行するようにシステムを設定できます。パスワード強度チェックをイネーブルにすると、各ユーザが強力なパスワードを使用する必要があります。

各ユーザが強力なパスワードを設定することを推奨します。ローカル認証されたユーザのパスワード強度チェックを有効にすると、Firepower eXtensible Operating System は次の要件を満たしていないパスワードを拒否します。

- 少なくとも 8 文字を含み、最大 127 文字であること



---

(注) コモンクライテリア要件に準拠するために、オプションでシステムの最小文字数 15 文字の長さのパスワードを設定できます。詳細については、[最小パスワード長チェックの設定 \(16 ページ\)](#)を参照してください。

---

- アルファベットの大文字を少なくとも 1 文字含む。
- アルファベットの小文字を少なくとも 1 文字含む。
- 英数字以外の文字（特殊文字）を少なくとも 1 文字含む。
- aaabbb など連続して 3 回を超えて繰り返す文字を含まない。
- passwordABC や password321 などの 3 つの連続した数字や文字をどのような順序であっても含まない。
- ユーザ名と同一、またはユーザ名を逆にしたものではない。
- パスワードディクショナリチェックに合格する。たとえば、辞書に記載されている標準的な単語に基づくパスワードを指定することはできません。
- 次の記号を含まない。\$（ドル記号）、?（疑問符）、=（等号）。



---

(注) この制限は、パスワードの強度チェックが有効になっているかどうかにかかわらず適用されます。

---

- ローカルユーザアカウントおよび admin アカウントの場合は空白にしない。

## リモート認証のガイドライン

システムを、サポートされているリモート認証サービスのいずれかに設定する場合は、そのサービス用のプロバイダーを作成して、Firepower 4100/9300 シャーシがそのシステムと通信できるようにする必要があります。ユーザ認証に影響する注意事項は次のとおりです。

### リモート認証サービスのユーザ アカウント

ユーザアカウントは、Firepower 4100/9300 シャーシにローカルに存在するか、またはリモート認証サーバに存在することができます。

リモート認証サービスを介してログインしているユーザの一時的なセッションを、Firepower Chassis Manager または FXOS CLI から表示できます。

### リモート認証サービスのユーザ ロール

リモート認証サーバでユーザアカウントを作成する場合は、ユーザが Firepower 4100/9300 シャーシで作業するために必要なロールをそれらのアカウントに含めること、およびそれ

らのロールの名前を **FXOS** で使用される名前と一致させることが必要です。ロールポリシーによっては、ユーザがログインできない場合や読み取り専用権限しか付与されない場合があります。

### リモート認証プロバイダーのユーザ属性

RADIUS および TACACS+ 構成では、ユーザが Firepower Chassis Manager または FXOS CLI へのログインに使用する各リモート認証プロバイダーに Firepower 4100/9300 シャーシ用のユーザ属性を設定する必要があります。このユーザ属性には、各ユーザに割り当てられたロールとロケールが含まれています。

ユーザがログインすると、FXOS は次を実行します。

1. リモート認証サービスに問い合わせます。
2. ユーザを検証します。
3. ユーザが検証されると、そのユーザに割り当てられているロールとロケールをチェックします。

次の表は、FXOS でサポートしているリモート認証プロバイダーのユーザ属性要件を比較したものです。

認証プロバイダー	カスタム属性	スキーマの拡張	属性 ID 要件
LDAP	オプション	次のいずれかを実行するように選択できます。 <ul style="list-style-type: none"> <li>• LDAP スキーマを拡張せず、要件を満たす既存の未使用の属性を設定します。</li> <li>• LDAP スキーマを拡張して、CiscoAVPair などの一意の名前でカスタム属性を作成します。</li> </ul>	シスコの LDAP の実装では、Unicode タイプの属性が必要です。 CiscoAVPair カスタム属性を作成する場合、属性 ID として 1.3.6.1.4.1.9.287247.1 を使用します 次の項で、サンプル OID を示します。
RADIUS	オプション	次のいずれかを実行するように選択できます。 <ul style="list-style-type: none"> <li>• RADIUS スキーマを拡張せず、要件を満たす既存の未使用属性を使用します。</li> <li>• RADIUS スキーマを拡張して、cisco-avpair などの一意の名前でカスタム属性を作成します。</li> </ul>	シスコによる RADIUS の実装のベンダー ID は 009 であり、属性のベンダー ID は 001 です。 次の構文例は、cisco-avpair 属性を作成する場合に複数のユーザロールとロケールを指定する方法を示しています。 shell:roles="admin,aaa" shell:locales="L1,abc"。複数の値を区切るには、区切り文字としてカンマ「,」を使用します。

認証プロバイダー	カスタム属性	スキーマの拡張	属性 ID 要件
TACACS+	必須	スキーマを拡張し、 <code>cisco-av-pair</code> という名前のカスタム属性を作成する必要があります。	<p><code>cisco-av-pair</code> 名は、TACACS+ プロバイダーの属性 ID を提供する文字列です。</p> <p>次の構文例は、<code>cisco-av-pair</code> 属性を作成するときに複数のユーザロールとロケールを指定する方法を示しています。</p> <pre>cisco-av-pair=shell:roles="admin aaa" shell:locales*"L1 abc"</pre> <p><code>cisco-av-pair</code> 属性構文でアスタリスク (*) を使用すると、ロケールがオプションとして指定され、同じ認可プロファイルを使用する他のシスコデバイスで認証の失敗を防ぐことができます。複数の値を区切るには、区切り文字としてスペースを使用します。</p>

### LDAP ユーザ属性のサンプル OID

カスタム `CiscoAVPair` 属性のサンプル OID は、次のとおりです。

```
CN=CiscoAVPair,CN=Schema,
CN=Configuration,CN=X
objectClass: top
objectClass: attributeSchema
cn: CiscoAVPair
distinguishedName: CN=CiscoAVPair,CN=Schema,CN=Configuration,CN=X
instanceType: 0x4
uSNCreated: 26318654
attributeID: 1.3.6.1.4.1.9.287247.1
attributeSyntax: 2.5.5.12
isSingleValued: TRUE
showInAdvancedViewOnly: TRUE
adminDisplayName: CiscoAVPair
adminDescription: UCS User Authorization Field
oMSyntax: 64
LDAPDisplayName: CiscoAVPair
name: CiscoAVPair
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,CN=X
```

## ユーザの役割

システムには、次のユーザ ロールが用意されています。

### 管理者

システム全体に対する完全な読み取りと書き込みのアクセス権。デフォルトの admin アカウントは、デフォルトでこのロールが割り当てられ、変更はできません。

### 読み取り専用

システム設定に対する読み取り専用アクセス権。システム状態を変更する権限はありません。

### 操作

NTP の設定、Smart Licensing のための Smart Call Home の設定、システム ログ (syslog サーバとエラーを含む) に対する読み取りと書き込みのアクセス権。システムの残りの部分に対する読み取りアクセス権。

### AAA アドミニストレータ

ユーザ、ロール、および AAA 設定に対する読み取りと書き込みのアクセス権。システムの残りの部分に対する読み取りアクセス権。

## ローカル認証されたユーザのパスワード プロファイル

パスワードのプロファイルには、ローカル認証されたユーザすべてのパスワード履歴やパスワード変更間隔プロパティが含まれます。ローカル認証されたユーザのそれぞれに異なるパスワード プロファイルを指定することはできません。

### パスワード履歴カウント

パスワード履歴のカウントにより、ローカル認証されたユーザが何度も同じパスワードを再利用しないようにすることができます。このプロパティが設定されている場合、Firepower シャーシは、ローカル認証されたユーザがこれまでに使用した最大 15 個のパスワードを保存します。パスワードは最近のものから時系列の逆順で格納され、履歴カウントがしきい値に達した場合に、最も古いパスワードだけを再利用可能にします。

あるパスワードが再利用可能になる前に、ユーザはパスワード履歴カウントで設定された数のパスワードを作成して使用する必要があります。たとえば、パスワード履歴カウントを 8 に設定した場合、ローカル認証されたユーザは 9 番目のパスワードが期限切れになった後まで、最初のパスワードを再利用できません。

デフォルトでは、パスワード履歴は 0 に設定されます。この値は、履歴のカウントをディセーブルにし、ユーザはいつでも前のパスワードを使用できます。

必要に応じて、ローカル認証されたユーザについてパスワード履歴カウントをクリアし、以前のパスワードの再利用をイネーブルにできます。

### パスワード変更間隔

パスワード変更間隔は、ローカル認証されたユーザが特定の時間内に行えるパスワード変更回数を制限することができます。次の表で、パスワード変更間隔の2つの設定オプションについて説明します。

間隔の設定	説明	例
パスワード変更禁止	このオプションを設定すると、ローカル認証されたユーザは、パスワードを変更してから指定された時間内はパスワードを変更できなくなります。  1 ~ 745 時間の変更禁止間隔を指定できます。デフォルトでは、変更禁止間隔は 24 時間です。	たとえば、ローカル認証されたユーザが 48 時間の間パスワードを変更できないようにする場合、次のように設定します。  <ul style="list-style-type: none"> <li>• [間隔中の変更 (Change During Interval) ] を無効にする</li> <li>• [変更禁止間隔 (No Change Interval) ] を 48 に設定する</li> </ul>
変更間隔内のパスワード変更許可	このオプションは、ローカル認証されたユーザのパスワードを事前に定義された時間内に変更できる最大回数を指定します。  変更間隔を 1 ~ 745 時間で、パスワード変更の最大回数を 0 ~ 10 で指定できます。デフォルトでは、ローカル認証されたユーザに対して、48 時間間隔内で最大 2 回のパスワード変更が許可されます。	たとえば、ローカル認証されたユーザがパスワードを変更した後 24 時間以内に最大 1 回そのパスワードを変更できるようにするには、次のように設定します。  <ul style="list-style-type: none"> <li>• [間隔中の変更 (Change During Interval) ] を有効にする</li> <li>• [変更カウント (Change Count) ] を 1 に設定する</li> <li>• [変更間隔 (Change Interval) ] を 24 に設定する</li> </ul>

## デフォルト認証サービスの選択

### 手順

**ステップ 1** セキュリティ モードを開始します。

Firepower-chassis # **scope security**

**ステップ 2** デフォルト認証セキュリティ モードを開始します。

Firepower-chassis /security # **scope default-auth**

**ステップ 3** デフォルト認証を指定します。



```
Firepower-chassis /security/default-auth # set realm auth-type
```

*auth-type* は、次のキーワードのいずれかです。

- **ldap** : LDAP 認証を指定します。
- **local** : ローカル認証を指定します。
- **none** : ローカル ユーザはパスワードを指定せずにログインできます。
- **radius** : RADIUS 認証を指定します。
- **tacacs** : TACACS+ 認証を指定します。

(注) デフォルトの認証とコンソール認証の両方が同じリモート認証プロトコル (RADIUS、TACACS+、または LDAP) を使用するように設定されている場合、そのサーバの設定の特定の側面を変更することは (たとえば、サーバの削除や、割り当ての順序の変更)、これらのユーザ設定を更新することなしではできません。

**ステップ 4** (任意) 関連付けられたプロバイダー グループを指定します (存在する場合)。

```
Firepower-chassis /security/default-auth # set auth-server-group auth-serv-group-name
```

**ステップ 5** (任意) このドメインのユーザに許可する更新要求間隔の最大時間数を指定します。

```
Firepower-chassis /security/default-auth # set refresh-period seconds
```

0 ~ 600 の整数を指定します。デフォルトは 600 秒です。

この時間制限を超えると、FXOS は Web セッションを非アクティブと見なしますが、そのセッションを終了することはありません。

**ステップ 6** (任意) FXOS が Web セッションを終了したと見なすまでの、最後の更新要求後からの最大経過時間を指定します。

```
Firepower-chassis /security/default-auth # set session-timeout seconds
```

0 ~ 600 の整数を指定します。デフォルトは 600 秒です。

(注) RADIUS または TACACS+ レルムに対して二要素認証を設定する場合は、リモートユーザが頻繁に再認証する必要がないよう、**セッションの更新時間およびセッションのタイムアウト時間を増やす**ことを検討してください。

**ステップ 7** (任意) 認証方式をレルムの二要素認証に設定します。

```
Firepower-chassis /security/default-auth # set use-2-factor yes
```

(注) 二要素認証は、RADIUS および TACACS+ レルムにのみ適用されます。

**ステップ 8** トランザクションをシステム設定にコミットします。

```
commit-buffer
```

**例**

次の例では、デフォルトの認証を RADIUS に設定し、デフォルトの認証プロバイダグループを `provider1` に設定し、二要素認証を有効にし、更新間隔を 300 秒（5 分）に設定し、セッションのタイムアウト間隔を 540 秒（9 分）に設定し、二要素認証を有効にします。その後で、トランザクションを確定します。

```
Firepower-chassis# scope security
Firepower-chassis /security # scope default-auth
Firepower-chassis /security/default-auth # set realm radius
Firepower-chassis /security/default-auth* # set auth-server-group provider1
Firepower-chassis /security/default-auth* # set use-2-factor yes
Firepower-chassis /security/default-auth* # set refresh-period 300
Firepower-chassis /security/default-auth* # set session-timeout 540
Firepower-chassis /security/default-auth* # commit-buffer
Firepower-chassis /security/default-auth #
```

## セッションタイムアウトの設定

FXOS CLI を使用することにより、ユーザアクティビティなしで経過可能な時間を指定できます。この時間が経過した後、Firepower4100/9300 シャーシはユーザセッションを閉じます。コンソールセッションと、HTTPS、SSH、および Telnet セッションとで、異なる設定を行うことができます。

タイムアウトとして 3600 秒（60 分）以下の値を設定できます。デフォルト値は 600 秒です。この設定を無効にするには、セッションタイムアウト値を 0 に設定します。

**手順**

**ステップ 1** セキュリティ モードを開始します。

```
Firepower-chassis # scope security
```

**ステップ 2** デフォルト認証セキュリティ モードを開始します。

```
Firepower-chassis /security # scope default-auth
```

**ステップ 3** HTTPS、SSH、および Telnet セッションのアイドル タイムアウトを設定します。

```
Firepower-chassis /security/default-auth # set session-timeout seconds
```

**ステップ 4** （任意）コンソールセッションのアイドル タイムアウトを設定します。

```
Firepower-chassis /security/default-auth # set con-session-timeout seconds
```

**ステップ 5** トランザクションをシステム設定にコミットします。

```
Firepower-chassis /security/default-auth # commit-buffer
```

**ステップ 6** （任意）セッションおよび絶対セッションタイムアウトの設定を表示します。

```
Firepower-chassis /security/default-auth # show detail
```

例：

```
Default authentication:
Admin Realm: Local
Operational Realm: Local
Web session refresh period(in secs): 600
Idle Session timeout (in secs) for web, ssh, telnet sessions: 600
Absolute Session timeout (in secs) for web, ssh, telnet sessions: 3600
Serial Console Session timeout(in secs): 600
Serial Console Absolute Session timeout(in secs): 3600
Admin Authentication server group:
Operational Authentication server group:
Use of 2nd factor: No
```

## 絶対セッションタイムアウトの設定

Firepower4100/9300 シャーシには絶対セッションタイムアウト設定があり、セッションの使用状況に関係なく、絶対セッションタイムアウト期間が経過するとユーザセッションは閉じられます。この絶対タイムアウト機能は、シリアルコンソール、SSH、HTTPS を含むすべての形式のアクセスに対してグローバルに適用されます。

シリアルコンソールセッションの絶対セッションタイムアウトを個別に設定できます。これにより、デバッグニーズに応えるシリアルコンソール絶対セッションタイムアウトは無効にしながら、他の形式のアクセスのタイムアウトは維持することができます。

絶対タイムアウト値のデフォルトは 3600 秒（60 分）であり、FXOS CLI を使用して変更できます。この設定を無効にするには、絶対セッションタイムアウト値を 0 に設定します。

### 手順

**ステップ 1** セキュリティ モードを開始します。

```
Firepower-chassis # scope security
```

**ステップ 2** デフォルト認証セキュリティ モードを開始します。

```
Firepower-chassis /security # scope default-auth
```

**ステップ 3** 絶対セッションタイムアウトを設定します。

```
Firepower-chassis /security/default-auth # set absolute-session-timeout seconds
```

**ステップ 4** （任意）別個のコンソールセッションタイムアウトを設定します。

```
Firepower-chassis /security/default-auth # set con-absolute-session-timeout seconds
```

**ステップ 5** トランザクションをシステム設定にコミットします。

```
Firepower-chassis /security/default-auth # commit-buffer
```

**ステップ6** (任意) セッションおよび絶対セッションタイムアウトの設定を表示します。

```
Firepower-chassis /security/default-auth # show detail
```

例 :

```
Default authentication:
Admin Realm: Local
Operational Realm: Local
Web session refresh period(in secs): 600
Idle Session timeout (in secs) for web, ssh, telnet sessions: 600
Absolute Session timeout (in secs) for web, ssh, telnet sessions: 3600
Serial Console Session timeout(in secs): 600
Serial Console Absolute Session timeout(in secs): 3600
Admin Authentication server group:
Operational Authentication server group:
Use of 2nd factor: No
```

## リモートユーザのロールポリシーの設定

デフォルトでは、LDAP、RADIUS、またはTACACS+プロトコルを使用してリモートサーバから Firepower Chassis Manager または FXOS CLI にログインするすべてのユーザに読み取り専用アクセス権が付与されます。セキュリティ上の理由から、確立されたユーザロールに一致するユーザにアクセスを制限することが望ましい場合があります。

リモートユーザのロールポリシーは、次の方法で設定できます。

### **assign-default-role**

ユーザがログインしようとしたときにリモート認証プロバイダーが認証情報付きのユーザロールを提供しなかった場合、そのユーザは読み取り専用ユーザロールでのログインが許可されます。

これはデフォルトの動作です。

### **no-login**

ユーザがログインしようとしたときにリモート認証プロバイダーが認証情報付きのユーザロールを提供しなかった場合は、アクセスが拒否されます。

### 手順

**ステップ1** セキュリティモードを開始します。

```
Firepower-chassis # scope security
```

**ステップ2** Firepower Chassis Manager および FXOS CLI へのユーザアクセスをユーザロールに基づいて制限するかどうかを指定します。

```
Firepower-chassis /security # set remote-user default-role {assign-default-role | no-login}
```

**ステップ3** トランザクションをシステム設定にコミットします。

```
Firepower-chassis /security # commit-buffer
```

---

#### 例

次の例では、リモートユーザのロールポリシーを設定し、トランザクションをコミットします。

```
Firepower-chassis# scope security  
Firepower-chassis /security # set remote-user default-role no-login  
Firepower-chassis /security* # commit-buffer  
Firepower-chassis /security #
```

## ローカル認証されたユーザのパスワードの強度チェックの有効化

パスワードの強度チェックが有効になっている場合、Firepower eXtensible Operating System では、強力なパスワードのガイドラインを満たしていないパスワードを選択できません ([パスワードに関するガイドライン \(3 ページ\)](#) を参照)。

#### 手順

---

**ステップ1** セキュリティ モードを開始します。

```
Firepower-chassis # scope security
```

**ステップ2** パスワード強度チェックを有効化するかディセーブルにするかを指定します。

```
Firepower-chassis /security # set enforce-strong-password {yes | no}
```

---

#### 例

次に、パスワード強度チェックを有効にする例を示します。

```
Firepower-chassis# scope security  
Firepower-chassis /security # set enforce-strong-password yes  
Firepower-chassis /security* # commit-buffer  
Firepower-chassis /security #
```

## ログイン試行の最大回数の設定

ロックアウト前にユーザに許可されるログイン試行の最大回数を指定します。この回数を超えると、指定した時間だけ Firepower 4100/9300 シャーシからロックアウトされることとなります。ユーザは、設定した最大回数を超過してログインを試行すると、システムからロックされます。ユーザがロックアウトされたことを示す通知は表示されません。これが起きると、ユーザは次にログインを試行できるようになるまで、指定された時間だけ待機する必要があります。

ログイン試行の最大数を設定するには、次の手順を実行します。



- (注)
- どのタイプのユーザアカウントであっても（管理者を含む）、ログイン試行の最大数を超過してログインを試行すると、システムからロックアウトされます。
  - 失敗できるログイン試行のデフォルトの最大回数は0です。ユーザがログイン試行の最大数を超過したときにシステムからロックアウトされるデフォルトの時間は、30分（1800秒）です。
  - ユーザのロックアウトのステータスを表示し、ユーザのロックアウト状態をクリアする手順については、[ユーザロックアウトステータスの表示およびクリア（15 ページ）](#)を参照してください。

このオプションは、システムのコモンクライテリア認定への準拠を取得するために提示される数の1つです。詳細については、[セキュリティ認定コンプライアンス](#)を参照してください。

### 手順

**ステップ 1** FXOS CLI から、セキュリティ モードに入ります。

```
scope security
```

**ステップ 2** 失敗できるログイン試行の最高回数を設定します。

```
set max-login-attempts num_attempts
```

*num\_attempts* の値は、0 ~ 10 の範囲内の任意の整数です。

**ステップ 3** ログイン試行の最高回数に達した後、ユーザがシステムからロックアウトされる時間（秒単位）を指定します。

```
set user-account-unlock-time
```

```
unlock_time
```

**ステップ 4** 設定をコミットします。

```
commit-buffer
```

## ユーザ ロックアウト ステータスの表示およびクリア

管理者ユーザは、失敗の回数が [最大ログイン試行回数 (Maximum Number of Login Attempts) ] CLI 設定で指定されたログイン最大試行回数を超えた後、Firepower 4100/9300 シャーシからロックアウトされているユーザのロックアウトステータスを表示およびクリアできます。詳細については、[ログイン試行の最大回数の設定 \(14 ページ\)](#) を参照してください。

### 手順

**ステップ 1** FXOS CLI から、セキュリティ モードに入ります。

```
scope security
```

**ステップ 2** 該当するユーザのユーザ情報 (ロックアウト ステータスを含む) を次のように表示します。

```
Firepower-chassis /security # show local-user user detail
```

例 :

```
□□□□ □□□□□□□□
□□
□□
□□□□□□
□□□
□□□□□□□□
Password:
□□□ □□□ □□□□□□□□
□□□□□ □□□□□□□□□□
□□□ □□□□
□□□□□□□□□□
□□□ SSH □□□□□
```

**ステップ 3** (任意) ユーザのロックアウト ステータスをクリアします。

```
Firepower-chassis /security # scope local-user user
```

```
Firepower-chassis /security/local-user # clear lock-status
```

## 変更間隔のパスワード変更の最大数の設定

### 手順

**ステップ 1** セキュリティ モードを開始します。

```
Firepower-chassis # scope security
```

**ステップ 2** パスワード プロファイル セキュリティ モードを開始します。

```
Firepower-chassis /security # scope password-profile
```

**ステップ 3** ローカル認証されたユーザが指定した時間内にパスワードを変更できる回数を制限します。

```
Firepower-chassis /security/password-profile # set change-during-interval enable
```

**ステップ 4** ローカル認証されたユーザが、[Change Interval] の間に自分のパスワードを変更できる最大回数を指定します。

```
Firepower-chassis /security/password-profile # set change-count pass-change-num
```

この値は、0 ~ 10 から自由に設定できます。

**ステップ 5** [Change Count] フィールドで指定したパスワード変更回数が適用される最大時間数を指定します。

```
Firepower-chassis /security/password-profile # set change-interval num-of-hours
```

この値は、1 ~ 745 時間から自由に設定できます。

たとえば、このフィールドが 48 に設定され、[Change Count] フィールドが 2 に設定されている場合、ローカル認証されたユーザは 48 時間以内に 2 回を超えるパスワード変更を実行することはできません。

**ステップ 6** トランザクションをシステム設定にコミットします。

```
Firepower-chassis /security/password-profile # commit-buffer
```

## 例

次の例は、change during interval オプションを有効にし、変更回数を 5 回、変更間隔を 72 時間に設定し、トランザクションをコミットします。

```
Firepower-chassis # scope security
Firepower-chassis /security # scope password-profile
Firepower-chassis /security/password-profile # set change-during-interval enable
Firepower-chassis /security/password-profile* # set change-count 5
Firepower-chassis /security/password-profile* # set change-interval 72
Firepower-chassis /security/password-profile* # commit-buffer
Firepower-chassis /security/password-profile #
```

## 最小パスワード長チェックの設定

最小パスワード長チェックを有効にした場合は、指定した最小文字を使用するパスワードを作成する必要があります。たとえば、`min_length` オプションを 15 に設定した場合、パスワードは 15 文字以上を使用して作成する必要があります。このオプションは、システムのコモンクライテリア認定への準拠のための数の 1 つです。詳細については、[セキュリティ認定コンプライアンス](#)を参照してください。



最小パスワード長チェックを設定するには、次の手順を実行します。

#### 手順

---

**ステップ 1** FXOS CLI から、セキュリティ モードに入ります。

```
scope security
```

**ステップ 2** パスワードの最小の長さを指定します。

```
set min-password-length min_length
```

**ステップ 3** 設定をコミットします。

```
commit-buffer
```

---

## パスワードの変更禁止間隔の設定

#### 手順

---

**ステップ 1** セキュリティ モードを開始します。

```
Firepower-chassis # scope security
```

**ステップ 2** パスワードプロファイルセキュリティ モードを開始します。

```
Firepower-chassis /security # scope password-profile
```

**ステップ 3** 間隔中の変更機能をディセーブルにします。

```
Firepower-chassis /security/password-profile # set change-during-interval disable
```

**ステップ 4** ローカル認証されたユーザが、新しく作成したパスワードを変更する前に待機する最小時間数を指定します。

```
Firepower-chassis /security/password-profile # set no-change-interval min-num-hours
```

この値は、1 ~ 745 時間の範囲で自由に設定できます。

この間隔は、[間隔中の変更 (Change During Interval)] プロパティが [無効 (Disable)] に設定されていない場合は無視されます。

**ステップ 5** トランザクションをシステム設定にコミットします。

```
Firepower-chassis /security/password-profile # commit-buffer
```

---

### 例

次に、間隔中の変更オプションを無効にし、変更禁止間隔を72時間に設定し、トランザクションをコミットする例を示します。

```
Firepower-chassis # scope security
Firepower-chassis /security # scope password-profile
Firepower-chassis /security/password-profile # set change-during-interval disable
Firepower-chassis /security/password-profile* # set no-change-interval 72
Firepower-chassis /security/password-profile* # commit-buffer
Firepower-chassis /security/password-profile #
```

## パスワード履歴カウンタの設定

### 手順

**ステップ1** セキュリティ モードを開始します。

```
Firepower-chassis # scope security
```

**ステップ2** パスワードプロファイルセキュリティ モードを開始します。

```
Firepower-chassis /security # scope password-profile
```

**ステップ3** ローカル認証されたユーザが、以前に使用したパスワードを再利用できるようになるまでに、作成する必要がある一意のパスワードの数を指定します

```
Firepower-chassis /security/password-profile # set history-count num-of-passwords
```

この値は、0 ~ 15 から自由に設定できます。

デフォルトでは、[履歴 (History Count) ] フィールドは0に設定されます。これにより、履歴カウンタが無効になるため、ユーザはいつでも以前に使用していたパスワードを再利用できます。

**ステップ4** トランザクションをシステム設定にコミットします。

```
Firepower-chassis /security/password-profile # commit-buffer
```

### 例

次の例は、パスワード履歴カウンタを設定し、トランザクションをコミットします。

```
Firepower-chassis # scope security
Firepower-chassis /security # scope password-profile
Firepower-chassis /security/password-profile # set history-count 5
Firepower-chassis /security/password-profile* # commit-buffer
Firepower-chassis /security/password-profile #
```

# ローカル ユーザ アカウントの作成

## 手順

**ステップ 1** セキュリティ モードを開始します。

```
Firepower-chassis# scope security
```

**ステップ 2** ユーザ アカウントを作成します。

```
Firepower-chassis /security # create local-user local-user-name
```

ここで *local-user-name* は、このアカウントにログインするときを使用されるアカウント名です。この名前は、固有であり、ユーザアカウント名のガイドラインと制限を満たしている必要があります ([ユーザ名に関するガイドライン \(3 ページ\)](#) を参照)。

ユーザを作成した後は、ログイン ID を変更できません。ユーザ アカウントを削除し、新しいユーザ アカウントを作成する必要があります。

**ステップ 3** ローカル ユーザ アカウントを有効化するかディセーブルにするかを指定します。

```
Firepower-chassis /security/local-user # set account-status {active|inactive}
```

**ステップ 4** ユーザ アカウントのパスワードを設定します。

```
Firepower-chassis /security/local-user # set password
```

パスワードを入力します。 *password*

パスワードを確認します。 *password*

パスワード強度チェックを有効にした場合は、ユーザパスワードを強固なものにする必要があります。Firepower eXtensible Operating System は強度チェック要件を満たしていないパスワードを拒否します ([パスワードに関するガイドライン \(3 ページ\)](#) を参照)。

(注) パスワードには次の記号を含めることはできません。\$ (ドル記号)、? (疑問符)、= (等号)。この制限は、パスワードの強度チェックが有効になっているかどうかにかかわらず適用されます。

**ステップ 5** (任意) ユーザの名を指定します。

```
Firepower-chassis /security/local-user # set firstname first-name
```

**ステップ 6** (任意) ユーザの姓を指定します。

```
Firepower-chassis /security/local-user # set lastname last-name
```

**ステップ 7** (任意) ユーザアカウントが期限切れになる日付を指定します。 *month* 引数は、月の英名の最初の 3 文字です。

```
Firepower-chassis /security/local-user # set expiration month day-of-month year
```

(注) ユーザアカウントに有効期限を設定した後、「有効期限なし」に再設定することはできません。ただし、使用できる最新の有効期限日付でアカウントを設定することは可能です。

**ステップ 8** (任意) ユーザの電子メールアドレスを指定します。

```
Firepower-chassis /security/local-user # set email email-addr
```

**ステップ 9** (任意) ユーザの電話番号を指定します。

```
Firepower-chassis /security/local-user # set phone phone-num
```

**ステップ 10** (任意) パスワードレス アクセス用の SSH キーを指定します。

```
Firepower-chassis /security/local-user # set sshkey ssh-key
```

**ステップ 11** すべてのユーザはデフォルトで *read-only* ロールに割り当てられ、このロールは削除できません。ユーザに割り当てる追加の各ロールに対して、以下を実行します。

```
Firepower-chassis /security/local-user # create role role-name
```

ここで *role-name* は、ユーザアカウントに割り当てる特権を表すロールです ([ユーザの役割 \(6 ページ\)](#) を参照)。

(注) ユーザ ロールおよび権限の変更は次回のユーザ ログイン時に有効になります。ユーザアカウントへの新しいロールの割り当てや既存のロールの削除を行うときにユーザがログインしている場合、アクティブなセッションは以前のロールや権限を引き続き使用します。

**ステップ 12** 割り当てられたロールをユーザから削除するには、以下を実行します。

```
Firepower-chassis /security/local-user # delete role role-name
```

すべてのユーザはデフォルトで *read-only* ロールに割り当てられ、このロールは削除できません。

(注) ユーザロールを削除すると、そのユーザの現在のセッション ID が取り消されます。つまり、すべてのユーザのアクティブセッション (CLI と Web の両方) がただちに終了します。

**ステップ 13** トランザクションをコミットします。

```
Firepower-chassis security/local-user # commit-buffer
```

---

## 例

次の例は、*kikipopo* という名前のユーザアカウントを作成し、ユーザアカウントを有効にし、*foo12345* にパスワードを設定し、管理ユーザロールを割り当て、トランザクションを確定します。

```

Firepower-chassis# scope security
Firepower-chassis /security # create local-user kikipopo
Firepower-chassis /security/local-user* # set account-status active
Firepower-chassis /security/local-user* # set password
Enter a password:
Confirm the password:
Firepower-chassis /security/local-user* # create role admin
Firepower-chassis /security/local-user* # commit-buffer
Firepower-chassis /security/local-user #

```

次の例は、lincey という名前のユーザアカウントを作成し、ユーザアカウントを有効にし、パスワードレス アクセス用の OpenSSH キーを設定し、AAA および操作ユーザロールを割り当て、トランザクションを確定します。

```

Firepower-chassis# scope security
Firepower-chassis /security # create local-user lincey
Firepower-chassis /security/local-user* # set account-status active
Firepower-chassis /security/local-user* # set sshkey "ssh-rsa
AAAAAB3NzaC1yc2EAAAABIwAAAIEAuo9VQ2CmWBI9/S1f30klCWjnV3lgdXMz00WU15iPw85lkdQqap+NFuNmHcb4K
iaQB8X/PDdmtlxQQcawclj+k8f4VcOelBx1sGk5luq51s1ob1VOIEwcKEL/h5lrdbN1I8y3SS9I/gGiBZ9ARlop9LDpD
m8HPh2LOgyH7Ei1MI8="
Firepower-chassis /security/local-user* # create role aaa
Firepower-chassis /security/local-user* # create role operations
Firepower-chassis /security/local-user* # commit-buffer
Firepower-chassis /security/local-user #

```

次の例は、jforlenz という名前のユーザアカウントを作成し、ユーザアカウントを有効にし、パスワードレスアクセス用のセキュア SSH キーを設定し、トランザクションを確定します。

```

Firepower-chassis# scope security
Firepower-chassis /security # create local-user jforlenz
Firepower-chassis /security/local-user* # set account-status active
Firepower-chassis /security/local-user* # set sshkey
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
User's SSH key:
> ---- BEGIN SSH2 PUBLIC KEY ----
>AAAAAB3NzaC1yc2EAAAABIwAAAIEAuo9VQ2CmWBI9/S1f30klCWjnV3lgdXMz00WU15iPw8
>5lkdQqap+NFuNmHcb4KiaQB8X/PDdmtlxQQcawclj+k8f4VcOelBx1sGk5luq51s1ob1VO
>IEwcKEL/h5lrdbN1I8y3SS9I/gGiBZ9ARlop9LDpDm8HPh2LOgyH7Ei1MI8=
> ---- END SSH2 PUBLIC KEY ----
> ENDOFBUF
Firepower-chassis /security/local-user* # commit-buffer
Firepower-chassis /security/local-user #

```

## ローカル ユーザ アカウントの削除

### 手順

**ステップ 1** セキュリティ モードを開始します。

```

Firepower-chassis# scope security

```

ステップ2 ローカルユーザ アカウントを削除します。

```
Firepower-chassis /security # delete local-user local-user-name
```

ステップ3 トランザクションをシステム設定にコミットします。

```
Firepower-chassis /security # commit-buffer
```

---

#### 例

次に、foo というユーザ アカウントを削除し、トランザクションをコミットする例を示します。

```
Firepower-chassis# scope security  
Firepower-chassis /security # delete local-user foo  
Firepower-chassis /security* # commit-buffer  
Firepower-chassis /security #
```

## ローカルユーザアカウントのアクティブ化または非アクティブ化

ローカルユーザアカウントをアクティブ化または非アクティブ化できるのは、admin 権限または AAA 権限を持つユーザのみです。

#### 手順

---

ステップ1 セキュリティ モードを開始します。

```
Firepower-chassis# scope security
```

ステップ2 アクティブ化または非アクティブ化するユーザに対してローカルユーザ セキュリティ モードを開始します。

```
Firepower-chassis /security # scope local-user local-user-name
```

ステップ3 ローカル ユーザ アカウントをアクティブ化するか非アクティブ化するかを指定します。

```
Firepower-chassis /security/local-user # set account-status {active | inactive}
```

(注) admin ユーザ アカウントは常にアクティブに設定されます。変更はできません。

ステップ4 トランザクションをシステム設定にコミットします。

```
Firepower-chassis /security/local-user # commit-buffer
```

---

### 例

次に、**accounting** というローカル ユーザ アカウントを有効にする例を示します。

```
Firepower-chassis# scope security
Firepower-chassis /security # scope local-user accounting
Firepower-chassis /security* # commit-buffer
Firepower-chassis /security #
```

## ローカル認証されたユーザのパスワード履歴のクリア

### 手順

**ステップ 1** セキュリティ モードを開始します。

```
Firepower-chassis # scope security
```

**ステップ 2** 指定したユーザ アカウントに対してローカル ユーザ セキュリティ モードを開始します。

```
Firepower-chassis /security # scope local-user user-name
```

**ステップ 3** 指定したユーザ アカウントのパスワード履歴をクリアします。

```
Firepower-chassis /security/local-user # clear password-history
```

**ステップ 4** トランザクションをシステム設定にコミットします。

```
Firepower-chassis /security/local-user # commit-buffer
```

### 例

次に、パスワード履歴を消去し、トランザクションを確定する例を示します。

```
Firepower-chassis # scope security
Firepower-chassis /security # scope local-user admin
Firepower-chassis /security/local-user # clear password-history
Firepower-chassis /security/local-user* # commit-buffer
Firepower-chassis /security/local-user #
```

