

PEM 8.1 -PEM Security Best Practice Guide

EnterpriseDB

PEM Security Guide

このドキュメントでは、PEMを構成する際に考慮する必要があるセキュリティに関する情報を提供します。 PEM機能では、これらのプラクティスを実施する必要はありません。ただし、EDBはこれらの方法を推奨して、システム全体のセキュリティを確保します。

PEMは、これらのコンポーネントを確保する保証に、Apache ウェブサーバなどのベンダ・リポジトリからサードパーティに依存OpenSSLには、snmp ++、libcurlを、などである日にアップしている、あなたは、オペレーティングシステムを更新し、定期的に任意のを避けるために、セキュリティ更新を適用する必要がありますセキュリティの脆弱性。最新のセキュリティパッチがなければ、システムは潜在的にサイバー攻撃に対して脆弱です。セキュリティパッチは、最新の更新を適用して最新の脅威から保護することにより、デバイスとそのデータを保護します。

セキュリティパッチを定期的に適用することには、次のような利点があります。

- サイバー攻撃の危険性を軽減
- 生産性の低下の回避
- マルウェア（ランサムウェアなど）からのデータ保護
- セキュリティの抜け穴を使用してネットワーク上に広がるワーム感染を回避する

apache_httpd_security_configuration pem_application_configurationの結論

</div>

Apache HTTPD Security Configurations

Windowsシステムでは、Apache HTTPDの名前付けはPEM HTTPDで、Apache httpd構成ファイル（ pme.conf ）は <Apache_Installation_Path>/conf/addons ディレクトリです。 ssl構成ファイル（ httpd-ssl-pem.conf ）は、 <Apache_Installation_Path>/conf/addons ディレクトリです。

Linuxシステムでは、apache httpd構成ファイル（ edb-pem.conf ）は <Apache_Installation_Path>/conf.d ディレクトリにあり、SSL構成ファイル（ edb-ssl-pem.conf ）は <Apache_Installation_Path>/conf.d ディレクトリです。

SSLv2およびSSLv3を無効にする

SSLバージョンSSLv2、SSLv3、TLS 1、およびTLS 1.1を無効にする必要があります。これらのバージョンは最も脆弱であり、暗号化の問題の影響を受けるためです。

バージョンを無効にするには、次のコマンドをapache httpd設定ファイルに追加してください：

```
SSLProtocol -ALL +TLSv1.2
```

構成ファイルに変更を適用するには、Webサーバーをリスタートする必要があります。

デフォルトでは、PEMはSSL構成ファイルに次の行を追加して、セキュリティのためにTLS 1.2を使用できるようにします。

```
SSLProtocol -All TLSv1.2
```

```
SSLProxyProtocol -All TLSv1.2
```

SSL証明書で安全なhttpd

EDBは、Webアプリケーション用にSSLセキュリティの追加レイヤーを持つことを推奨しています。

デフォルトでは、PEMのインストール中に、PEMはApache / HTTPDサーバーの自己署名証明書を生成して使用します。PEMに独自のSSL証明書を使用する場合は、Apache構成ファイルを更新する必要があります。

Linuxシステムでは、ファイルに対する書き込みパーミッションを持つユーザとして、テキストエディタでApache httpd構成ファイル（`edb-ssl-pem.conf`）を開く必要があります。また、構成ファイル内のサーバー名前とファイル名を変更して、証明書ファイルとマッチせる必要があります。

PEM VirtualHostセクション内には、更新する必要がある2つのSSLディレクティブがあります。

- `SSLCertificateFile` はDigiCert証明書ファイル（たとえば、`your_domain_name.crt`）です。
- `SSLCertificateKeyFile` は、あなたがCSRを作成したときに生成したキーファイル（例えば、`your_private`。キー）。

例、更新：

```
SSLEngine on
```

```
SSLCertificateFile /path/to/your_domain_name.crt
```

```
SSLCertificateKeyFile /path/to/your_private.key
```

Postgres Enterprise Managerで、httpd自己署名SSL証明書を信頼できるCA署名付き証明書に置き換えることもできます。手順については、以下に示すリンクに従ってください。

<https://www.enterprisedb.com/postgres-tutorials/how-replacing-httpd-self-signed-ssl-certificates-trusted-ca-signed-certificates>

Webサーバーの情報漏えいを無効にする

EDBは、ウェブサーバのセキュリティのパートとして、すべてのウェブサーバの署名を無効にすることをお勧めします。ウェブサーバはソフトウェアシグネチャを公開します。シグネチャ

を無効にするには、Apache httpd構成ファイルに次のパラメーターを追加します。デフォルトでは、PEMは以下のパラメーターをApache httpd構成ファイルに追加することにより、情報の公開を無効にします。

```
ServerTokens Prod
```

```
ServerSignature Off
```

ServerTokens ディレクティブは、クライアントに返されるサーバー応答ヘッダフィールドを制御します。Apache httpd構成ファイルにこのパラメータを追加して、Apacheサーババージョンにすることをお勧めします。

ServerSignature ディレクティブには、サーバーで作成されたドキュメントのフッターが含まれます。フッターには、Apacheやオペレーティングシステムのバージョンなど、Apacheの構成に関する情報が含まれています。このような情報の表示を制限するには、Apache httpd構成ファイルでこのディレクティブを無効にすることをお勧めします。

Apache httpd構成ファイルに変更を適用するには、ウェブサーバをリスタートする必要があります。

ディレクトリ一覧の無効化

ディレクトリのリストにより、攻撃者は完全なディレクトリの内容をビューできます。デフォルトでは、ウェブサーバはこのオプションを有効にしており、攻撃者は任意のファイルを発見およびビューできます。このリストは、攻撃者がアプリケーションをリバースエンジニアリングしてソースコードを取得し、考えられるセキュリティ上の欠陥を分析し、アプリケーションに関する詳細情報を発見する可能性があります。

これを回避するには、Apache httpd構成ファイルで **Options** ディレクティブを設定して、ディレクトリリストを無効にする必要があります。デフォルトでは、PEMはウェブサーバ設定ファイルで以下のオプションを設定することにより、ディレクトリリストを無効にします。

```
<Directory /application/directory> Options -Indexes </Directory>
```

構成ファイルに加えた変更を適用するには、ウェブサーバをリスタートする必要があります。

ネットワークまたはIPアドレスへのアクセスを制限する

Apacheは、クライアントのホスト名またはIPアドレスに基づいてアクセス制御を提供します。特定のIPアドレスまたはネットワーク別にアプリケーションをビューするには、ユーザはApache構成ファイルを次のように変更して、**Allow** ディレクティブ内でネットワークアドレスを提供できます。

```
<Directory /application/hostname>
```

```
Options None
```

```
AllowOverride None
```

```
Order deny,allow
```

```
Deny from all
```

```
Allow from 192.168.0.0/24
```

```
</Directory>
```

PEMは、アプリケーション構成ファイルの `ALLOWED_HOSTS` 設定パラメータを使用して、許可されたホストのIPアドレスのリストを提供します。アプリケーション構成の `config_local.py` ファイルは `<PEM_INSTALLATION_PATH>/web` にあります。

デフォルトでは、PEMはすべてのホストがアプリケーションに接続できるようにします。

例：

以下のように、構成ファイルでIPアドレスのレンジを設定できます。

```
ALLOWED_HOSTS = ['225.0.0.0/8', '226.0.0.0/7', '228.0.0.0/6']
```

以下のように、構成ファイルでサブネットレベルのホストを許可するようにIPアドレスを設定できます。

```
ALLOWED_HOSTS = ['192.0.2.0/28', ':::192.0.2.0/124']
```

以下のように、構成ファイルで特定の個々のホストアドレス（IPアドレスに基づいて）を設定できます。

```
ALLOWED_HOSTS = ['127.0.0.1', '192.168.0.1']
```

変更をアプリケーション構成ファイルに適用するには、ウェブサーバをリスタートする必要があります。

クロスサイトトレース

ウェブサーバ接続をデバッグするには、TRACEとTRACKの2つのHTTPメソッドがあります。HTTP TRACE要求がそれをサポートするウェブサーバに送信されると、そのサーバは応答し、HTTPヘッダーを含む、渡されたデータをエコーします。Apache構成内でこれらのメソッドを無効にすることをお勧めします。

すべての仮想ホストでTRACEメソッドを無効にするには、次の行をApache httpd構成ファイルに追加します。

```
TraceEnable off
```

特定の仮想ホストでこれらの方法を無効にするには、Apache構成ファイルで各仮想ホストに次の行を追加します。PEMは、Apache httpd構成ファイルに次の行を追加します。

```
RewriteEngine on
```

```
RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK|OPTIONS)
```

```
RewriteRule .* - [F]
```

非特権ユーザーアカウントからWebサーバーを実行する

ルートユーザとしてApache ウェブサーバを実行すると、セキュリティの問題が発生します。ウェブサーバを一意的な非特権ユーザとして実行することを常にお勧めします。これにより、セキュリティ違反が発生したイベントに実行されている他のサービスを保護できます。

PEMはWSGIアプリケーションとして実行されます。実行中のWSGIアプリケーションを委任するには、`WSGIDaemonProcess` ディレクティブを使用して個別のデーモンプロセスを作成できます。

Linuxでは、Apache ウェブサーバはルートユーザとして起動しますが、デーモンはどのPEMアプリケーションが `pem` ユーザとして実行するかを処理します。Windowsでは、`WSGIDaemonProcess` ディレクティブとその機能は使用できません。PEM HTTPDはインストール時にサービスとしてインストールされており、ユーザは、サービスのインストールが成功するための `Administrators` グループのメンバーでニーズ。

デフォルトでは、Apacheサービスはシステムユーザ（`LocalSystem` アカウント）として実行するように登録されています。

PEM WebServerでセキュリティHTTPヘッダーをカスタマイズする

PEMには、次のセキュリティ問題を修正するための独自の構成ファイルが含まれています。`config.py` ではなく、`config_local.py` の構成のみをオーバーライドすることをお勧めします。ほとんどの場合、`config_local.py` はシステムに存在しません。したがって、ユーザーはアプリケーションレベルの構成を上書きするために作成する必要があります。PEMのアップグレード、`config_local.py` は上書きされませんが、`config.py` と `config_distro.py` の変更は上書きされることにノートしてください。PEMをアンインストールした後、ユーザーは `config_local.py` を削除する必要があります。

デフォルトでは、`config_local.py` はLinuxでは `/usr/edb/pem/web` に、Windows `C:\ProgramFiles\edb\pem\server\share\web` にあります。

ホストヘッダーインジェクション攻撃

HTTPホストヘッダー攻撃は、ホストヘッダー値を安全でない方法でハンドルする脆弱なWebサイトを悪用します。サーバーが暗黙的にホストヘッダーを信頼し、適切に検証またはエスケープできない場合、攻撃者はこの入力を使用して、サーバサイドの動作を操作する有害なペイロードを挿入できる可能性があります。通常、Webアプリケーションは、セットアップ中に構成ファイルで指定されない限り、デプロイされているドメインを認識しません。例、現在のドメインを知る必要がある場合、ホストヘッダーからドメインを取得して絶対URLを生成することに頼ることがあります。ホストヘッダーは他の脆弱性、最も顕著なWebキャッシュ>ポイズニング&SQLインジェクションのレンジを活用するための潜在的なベクトルです。

X-Frame-Options

X-Frame-Optionsを使用して、ブラウザーがページを<iframe>タグでレンダリングできるようにするかどうかを示すことができます。クリックジャックから保護するために特別に設計されました。PEMには、このような種類の攻撃を防ぐためのホストバリデーション `X_FRAME_OPT`

IONS オプションがあり、`config_local.py` ファイルで構成できます。デフォルトは次のとおりです。

```
X_FRAME_OPTIONS = "SAMEORIGIN"
```

コンテンツセキュリティポリシー

Content-Security-PolicyはHTML5標準のパートであり、X-Frame-Optionsヘッダー（置き換えられる）よりもレンジ保護を提供します。ウェブサイトの作成者は、リソース（スクリプト、スタイルシート、フォントなど）を読み込むことができる個々のドメイン、およびページの埋め込みを許可されているドメインをホワイトリストに登録できるように設計されています。

PEMには、このような種類の攻撃を防ぐためのホストバリデーション

CONTENT_SECURITY_POLICY オプションがあり、`config_local.py` ファイルで構成できます。デフォルトは次のとおりです。

```
CONTENT_SECURITY_POLICY = "default-src https: data: blob: 'unsafe-inline' 'unsafe-val';"
```

厳格な輸送セキュリティ

Strict-Transport-Security応答ヘッダー（多くの場合HSTSと略されます）を使用すると、WebサイトまたはWebアプリケーションは、HTTPではなくHTTPSを使用してのみアクセスする必要があります。このオプションを使用すると、中間者攻撃を防ぐことができます。デフォルトは次のとおりです。

```
STRICT_TRANSPORT_SECURITY = "max-age=31536000;includeSubDomains"
```

Note * このパラメータを追加すると、構成が変更された場合に問題が発生する可能性があります。したがって、PEMのインストールが完了してテストされた後にのみ、これを追加することをお勧めします。

X-Content-Type-Options

X-Content-Type-Options応答HTTPヘッダーは、Content-TypeヘッダーでアドバタイズされたMIMEタイプを変更して従わないことを示すためにサーバーが使用するマーカーです。これは、MIMEタイプのスニффイングをオプトアウトする方法、つまり、MIMEタイプが意図的に設定されていると言う方法です。デフォルトは次のとおりです。

```
X_CONTENT_TYPE_OPTIONS = "nosniff"
```

X-XSS-Protection

クロスサイトスクリプティング（XSS）は、Webサーバーで最も一般的なアプリケーション層の脆弱性の1つです。XSSを使用すると、攻撃者はクライアント側のスクリプトを他のユーザーが表示するウェブページに挿入できます。ヘッダーに対するHTTP X-XSS-Protectionの応答は、Internet Explorer、Chrome、およびSafariの機能であり、クロスサイトスクリプティング（XSS）の反射攻撃を検出するとページのローディングを停止します。インラインJavaScript（「unsafe-inline」）の使用を無効にする強力なContent-Security-Policyをサイトが実装している場合、これらの保護は現代のブラウザではほとんど不要ですが、CSPをまだサポートしていない古いWebブラウザのユーザーに保護を提供できます。デフォルトは次のと

おりです。

```
X_XSS_PROTECTION = "1; mode=block"
```

これを回避するには、Apache構成ファイルに次のオプションを追加する必要があります。

```
<IfModule mod_headers.c>
```

```
Header set X-XSS-Protection "1; mode=block"
```

```
</IfModule>
```

構成ファイルの変更を適用するには、ウェブサーバのリスタートが必要です。

デフォルトでは、PEMは `/usr/edb/pem/web/config.py` にあるアプリケーション構成ファイルに `X-XSS-Protection to "1; mode=block"` を設定します。

この変更を有効にオーダーには、Apacheサービスのリスタートが必要です

`config.py` ファイルの詳細については、[PEM Online Help](#)を参照してください。

Cookieセキュリティ

Cookieは、サーバーがブラウザに送信して構成データを保存できるデータの小さなパケットです。ブラウザはそれらをすべてのリクエストとともに同じサーバーに自動的に送信するため、Cookieを保護する方法を知ることが重要です。そこは `config.py` で参照することができクッキーを確保makeにPEMによって提供されるマルチプル構成オプションがありますが、続く3つが最も重要です。

1. **** SESSION_COOKIE_SECURE ****-セキュアフラグを設定すると、Cookieが暗号化されていない接続で送信されることを防ぎます。基本的に、暗号化されたチャネルを使用しないサーバーへのリクエストにCookieを追加しないようにブラウザに指示します。ブラウザは、HTTPSなどの接続にのみCookieを追加します。デフォルトは次のとおりです。
 - `SESSION_COOKIE_SECURE = True`
2. **** SESSION_COOKIE_HTTPONLY ****-デフォルトでは、CookieのコンテンツはJavaScriptを介して読み取ることができます。 `HTTPOnly` フラグは、スクリプトがCookieを読み取れないようにします。 `HTTPOnly` という名前が示すように、ブラウザはHTTP (S) 要求でのみCookieを使用します。これにより、ハッカーがXSS脆弱性を使用してCookieのコンテンツを学習することを防ぎます。例、 `sessionId` クッキーのためにそう `sessionId` クッキーのために、あなたは常に `HTTPOnly` フラグを設定することができ、クライアント側のスクリプトとクッキーを読み取るために必要なことはありません。デフォルトは次のとおりです。
 - `SESSION_COOKIE_HTTPONLY = True`
3. **** ENHANCED_COOKIE_PROTECTION ****このオプションを `True` に設定すると、IPアドレスとユーザーエージェントに従ってトークンが生成されます。後続のすべてのリクエストでは、トークンが再計算され、最初のリクエストに対して計算されたトークンと照合されます。セッションCookieが盗まれ、攻撃者が別の場所からそれを使用しようとすると、生成されるトークンは異なります。その場合、拡張機能はセッションをクリアし、リクエス

トをブロックします。デフォルトは次のとおりです。

- `ENHANCED_COOKIE_PROTECTION = True`
- サーバは、このようなKubernetesなどの環境を、ホスティング、動的IPアドレスやロードバランサの背後に展開されている場合!!! Note
- このオプションは、問題を引き起こす可能性があります。そのような場合、このオプションは `False` に設定する必要があります。
- この変更を有効にオーダーには、Apacheサービスのリスタートが必要です。
- `config.py` ファイルの詳細については、[PEM Online Help](#)を参照してください。

PEM application Security Configurations

セッションタイムアウト

Webアプリケーションによるセッションの有効期限が不十分であると、攻撃者が有効なセッションIDを再利用して関連するセッションを乗っ取ることができるため、他のセッションベースの攻撃の危険性が高まります。セッションインターバルが短いほど、攻撃者が有効なセッションIDを使用する必要がある時間が短くなります。このセキュリティ問題を回避するために、Webアプリケーションの非アクティブタイムアウトを低い値に設定することをお勧めします。

Postgres Enterprise Managerは、ユーザセッションのタイムアウト値を設定する方法を提供します。Webコンソールに指定された期間ユーザのアクティビティがない場合、PEMはWebコンソールからユーザをログします。PEM管理者は、非アクティブな時間の長さを設定できます。この値は、個々のユーザではなく、アプリケーションごとです。タイムアウト期間を構成するには、`<PEM_INSTALLATION_PATH>/web` ディレクトリにある `config_local.py` ファイルの `USER_INACTIVITY_TIMEOUT` パラメータを変更します。デフォルトでは、この機能は無効になっています。

例、アプリケーションが非アクティブの15分後にユーザをログするように指定するには、セット：

```
USER_INACTIVITY_TIMEOUT = 900
```

Note * タイムアウト値は秒単位で指定されます。

この変更を有効にオーダーには、Apacheサービスのリスタートが必要です。

`config.py` ファイルの詳細については、[PEM Online Help](#)を参照してください。

RestAPIヘッダーのカスタマイズ

要件に応じてRestAPIトークンヘッダーをカスタマイズします。デフォルト値は `config.py` ファイルによって公開されません。 `config_local.py` ファイルの次のヘッダーをカスタマイズします。

PEM_HEADER_SUBJECT_TOKEN_KEY

この構成オプションを使用すると、HTTPヘッダー名前を変更して、生成されたトークンを取得できます。デフォルトでは、ユーザがトークンを作成する要求を送信すると、サーバー応答には「X-Subject-Token」ヘッダーが含まれます。これには、新しく生成されたトークンの値が含まれます。ヘッダー名前をカスタマイズする場合は、次のように `config_local.py` ファイルを更新できます。

```
PEM_HEADER_SUBJECT_TOKEN_KEY = 'Pem-RestAPI-Generate-Token'
```

次の出力が得られます。

```
curl -ik -X POST -d '{"username":"enterprisedb","password":"edb"}' -H "Content-Type: application/json" https://localhost:8443/pem/api/token/
```

```
HTTP/1.1 201 CREATED
```

```
Date: Thu, 29 Oct 2020 11:03:48 GMT
```

```
Server: Apache
```

```
Content-Length: 326
```

```
Pem-RestAPI-Generate-Token: 997aef95-d46d-4d84-932a-a80146eaf84f
```

PEM_HEADER_TOKEN_KEY

この構成オプションを使用すると、トークンをPEMサーバーに送信する際に使用するHTTP要求ヘッダー名前を変更できます。デフォルトでは、ユーザがトークンを生成するリクエストを送信すると、トークンヘッダーは `X-Auth-Token` になります。RestAPIリクエストヘッダー名前をカスタマイズする場合は、次のように `config_local.py` ファイルを更新できます。

```
PEM_HEADER_TOKEN_KEY = 'Pem-Token'
```

これにより、トークンを送信できます。

```
$ curl -Lk -X GET -H "Pem-Token: gw5rzaloxyp91tttd1c97w24b5sv60cllc24sxy9" https://localhost:8443/pem/api/v4/agent
```

PEM_TOKEN_EXPIRY

この構成オプションを使用すると、PEM RestAPIトークンの生成後の有効期限を変更できます。デフォルトでは、トークンの有効期限は20分（1200秒）に設定されています。トークンの有効期限を10分に変更する場合は、次のように `config_local.py` ファイルを更新できます。

```
PEM_TOKEN_EXPIRY = 600
```

この変更を有効にオーダーには、Apacheサービスのリスタートが必要です。

PEMでの役割ベースのアクセス制御

役割ベースのアクセス制御（RBAC）は、組織内のユーザーのロールに基づいてアプリケーション

ンアクセスを制限し、アクセス制御のプライマリ方法の1つです。RBACのロールは、ユーザーがアプリケーションに対して持つアクセスのレベルを指します。ユーザーは、唯一効果的にジョブを実行するために必要な情報にアクセスすることを許可されています。PEMのロールは、添字ではなく継承可能で付加的です。シンプルに言えば、PEM管理者として、ユーザに最低レベルのロールを付与し、ユーザがそれぞれのタスクを実行するために必要な役割を付与する必要があります。例、唯一のSQLへのアクセスを提供します

```
CREATE ROLE user_sql_profiler WITH LOGIN NOSUPERUSER NOCREATEDB NOCREATEROLE INHER  
NOREPLICATION CONNECTION LIMIT -1 PASSWORD 'xxxxxx';
```

```
GRANT pem_user, pem_comp_sqlprofiler TO user_sql_profiler;
```

ロールの詳細については、[PEM Roles](#)をご覧ください。

SQL/ Protectプラグイン

SQLインジェクション攻撃の防止は、通常、アプリケーション開発者の責任です。データベース管理者は通常、潜在的な脅威をほとんどまたはまったく制御できません。データベース管理者にとっての困難は、アプリケーションが適切にファンクションするためにデータにアクセスできる必要があることです。

SQL/Protect は、データベース管理者がSQLインジェクション攻撃からデータベースを保護できるようにするモジュールです。SQL / Protectは、標準的なデータベースセキュリティポリシーに加えて、一般的なSQLインジェクションプロファイルの着信クエリを調べることであり、セキュリティのレイヤーを提供します。

SQLインジェクション攻撃を実行するために使用されるいくつかの異なる手法があります。特定のシグネチャが各手法を特徴づけます。SQL/ Protectは、不正な関係、ユーティリティコマンド、SQLトートロジー、無制限のDMLステートメントのクエリを調べます。SQL/ Protectは、潜在的に危険なクエリを管理者に警告してブロックすることにより、データベース管理者に制御を戻します。

Note * このプラグインはEPASサーバーでのみ機能するため、PEMデータベースがEPASサーバーでホストされている場合にのみ有効です。

SQL Profilerプラグインの詳細については、[PEM Online Help - SQL Profiler](#)を参照してください。

パスワード管理

PEM管理ユーザーのセキュリティのヒントの1つは、PEMログインパスワードを定期的に新しいものに変更することです。パスワードを変更すると、次のような多くの危険を回避できます。

- マルチプルアカウントの侵害
- 定数アクセスを防止
- 物理的に保護されていないシステムで保存されたパスワードの使用を防止します
- キーストロークロガーによって得られるアクセスを制限する

非rootユーザーでpemAgentジョブを実行する

ほとんどの場合、`pemAgent` はルートユーザとしてインストールされ、ルート特権を持つデーモンプロセスとして実行されます。デフォルトでは、`PEM`はスケジュールされたジョブ/タスクの実行を無効にします。`PEM`は、`pemAgent`構成ファイルを変更することにより、非ルートユーザとしてスケジュールされたジョブを実行するためのサポートを提供します。

スケジュールされたジョブを非ルートユーザとして実行するには、`agent.cfg` ファイルの `batch_script_user` パラメータのエントリを変更し、スクリプトの実行に使用するユーザを指定します。非ルートユーザまたはルートユーザIDを指定できます。ユーザを指定しない場合、または指定したユーザが存在しない場合、スクリプトは実行されません。ファイルを変更した後、エージェントを再起動します。非ルートユーザが `pemagent` を実行している場合、`batch_script_user` の値は無視され、`pemagent` の実行に使用される同じ非ルートユーザがスクリプトを実行します。

Windowsシステムでスクリプトを呼び出すには、`AllowBatchJobSteps` のレジストリエントリを `true` に設定し、`PEM`エージェントをリスタートします。`PEM`レジストリエントリは次の場所にあります。

```
HKEY_LOCAL_MACHINE\Software\Wow6432Node\EnterpriseDB\PEM\agent
```

pemAgentおよびPEMバックエンドデータベースサーバー証明書の変更

デフォルトでは、`PEM`をインストールすると、インストーラは`pemAgent`および`PEM`データベースサーバの自己署名証明書を生成して使用します。`PemAgent`は、`PEM`データベースサーバへの接続時にこれらの証明書を使用します。`pemAgent`および`PEM`データベースサーバに独自のSSL証明書を使用するに

は、`PEM Administrators's Guide` の `Managing Certificates` セクションに記載されている手順に従います。

Note * `PEM`は、SSL CA証明書をカスタムの場所に配置することをサポートしていません。`ca_certificate.crt` と `ca_key.key` の場所を変更しないでください。