セキュアなシステムコンフィギュレーション ファイル

機能の概要と変更履歴

要約データ

該当製品または機能エリア	すべて
該当プラットフォーム	ASR 5500 VPC-DI VPC-SI
機能のデフォルト	無効
このリリースでの関連する変更点	N/A
関連資料	• *ASR 5500 System Administration Guide* • *VPC-DI* システム管理ガイド • *VPC-SI System* アドミニストレーション ガイド

マニュアルの変更履歴

改訂の詳細	リリース
最初の導入。	21.3

機能説明

システム設定ファイルには、オペレータのネットワークのセットアップと運用に使用される重要な設定情報が含まれています。設定ファイルは、ネットワークに悪影響を及ぼす可能性のあるファイルへの不正な変更を避けるために、ロード前に適切に承認されている必要があります。

この機能により、ロードされる前にコンフィギュレーションファイルの整合性と信頼性を確保するために、システム設定ファイルを RSA キーで署名できるようになります。オペレータは秘密キーを使用して、コンフィギュレーションファイルに署名できます。システムは公開キーを使用して、署名されたコンフィギュレーション ファイルをロードする前に検証します。

システム コンフィギュレーション ファイルの保護方法

デジタル署名の作成

オペレータは、次の手順を使用して設定ファイルに署名できます。

1. 設定ファイル上で SHA512 ハッシュを実行して、メッセージダイジェストを作成します。

   例（Linux/OpenSSL）：

   ```
   openssl dgst -sha512 -binary -out digest cfg_file
   ```

2. RSA 秘密キーを使用してメッセージダイジェスト値を暗号化することによって、デジタル署名を作成します。

   例（Linux/OpenSSL）：

   ```
   openssl pkeyutl -sign -in digest -inkey pri_key.pem -out sig \
   -pkeyopt digest:sha512 -pkeyopt rsa_padding_mode:pss \
   -pkeyopt rsa_pss_saltlen:-2
   ```

3. デジタル署名を base64 形式に変換します（先頭に「#」が追加され、最後に新しい行が追加されます）。

   例（Linux/OpenSSL）：

   ```
   echo -n "#" > sig_base64
   base64 sig -w 0 >> sig_base64
   echo "" >> sig_base64
   ```

4. デジタル署名を使用して元の設定ファイルを追加します。

   例（Linux/OpenSSL）：

   ```
   cat sig_base64 cfg_file > signed_cfg_file
   ```

公開キーと秘密キーの生成

RSA 公開キーは PEM 形式（.pem ファイル）に保存され、次の例の OpenSSL コマンドのいずれかを使用して生成できます。

```
openssl rsa -in pri_key.pem - pubout -out pub_key.pem
--or--
openssl rsa -in pri_key.pem -RSAPublicKey_out -out pub_key.pem
```

PEM 形式の RSA 秘密キーは、次の例の OpenSSL コマンドを使用して生成できます。

```
openssl genrsa -out pri_key.pem 2048
```

openssl rsa コマンドと **openssl genrsa** コマンドの詳細については、それぞれの OpenSSL のマニュアルページを参照してください。

デジタル署名の検証

署名の検証が有効になっている場合、システムが起動し、コンフィギュレーションファイル（または設定ファイルがロードされた時点) をロードするときに、デジタル署名の検証が行われます。システムは、セキュアディレクトリで *enable_cfg_pubkey* ファイルを検索して、署名検証が有効（または無効）になっているかどうかを判断します。詳細については、署名検証の有効化または無効化 （4 ページ）を参照してください。

システムは、次の手順を使用して、署名されたコンフィギュレーションファイルを検証します。

1. フラッシュから RSA 公開署名キーを抽出します。
2. コンフィギュレーションファイルのデジタル署名（最初の行）を抽出します。
3. base64 形式からバイナリ形式に署名を変換します。
4. RSA 公開キーを使用して署名を復号します。
5. プレーンコン フィギュレーション ファイルの SHA512 ハッシュを計算して、メッセージダイジェストが生成されるようにします。
6. 復号された署名の値と新しく計算されたメッセージダイジェストを比較します。一致する場合、コンフィギュレーションファイルは正常に検証されます。

署名検証の設定

検証用の RSA 公開キーのインポート

署名済みの設定ファイルを確認するには、RSA 公開キー（PEM 形式）をインポートする必要があります。RSA 公開キーをインポートするには、次のコマンドを使用します。

重要 このコマンドは、コンソールからのみ実行できます。

cfg-security import public-key url *url_address*

注：

- コマンドが実行されると、既存の .pem ファイルが新しい .pem ファイルに置き換えられます。
- *url_address* はローカルファイルまたはリモートファイルを参照します。また、次の形式を使用して入力する必要があります。

  [file:]{/flash | /usb1 | /hd-raid | /sftp}[*/directory***]/***filename*

  tftp://*host***[:***port***][/<***directory***>]/***filename*

  ftp://[*username***[:***password***]@]***host***[:***port***][/***directory***]/***filename*

  sftp://[*username***[:***password***]@]***host***[:***port***][/***directory***]/***filename*

  http://[*username***[:***password***]@]***host***[:***port***][/***directory***]/***filename*

  https://[*username***[:***password***]@]***host***[:***port***][/***directory***]/***filename*

署名検証の有効化または無効化

コンフィギュレーション ファイルで署名の検証を有効化（または無効化）するには、次のコマンドを使用します。

重要 このコマンドは、コンソールからのみ実行できます。

[no] cfg-security sign

注：

- 署名の検証（**cfg-security sign** コマンド）を有効にすると、PEM ファイルが存在するディレクトリ内に *enable_cfg_pubkey* という名前の空のファイルが作成されます。
- コンフィギュレーション ファイルの署名の検証を無効にするには、**no cfg-security sign** コマンドを使用します。署名の検証（**no cfg-security sign** コマンド）を無効にすると、*enable_cfg_pubkey* ファイルが削除されます。
- システムは、署名の検証が有効か無効かを判断するために、*enable_cfg_pubkey* ファイルを検索します。