その他の注意事項

Hybrid Data Security に関する既知の問題

- (ハイブリッドデータ セキュリティ でクラスタを削除するか、すべてのノードをシャットダウンして）Control Hub クラスタをシャットダウンした場合、構成 ISO ファイルが失われた場合、またはキーストア データベースにアクセスできなくなった場合、Webex アプリ ユーザは、KMS でキーを使用して作成された [ユーザ（People）] リストに含まれるスペースを使用できなくなります。これは、トライアルと実稼働の両方の導入に当てはまります。現在この問題の回避策や修正方法はないため、アクティブなユーザアカウントを処理した後で HDS サービスをシャット ダウンしないことを強くお勧めします。

- すでに ECDH で KMS に接続しているクライアントは、一定期間（1 時間程度）その接続を保持します。ユーザが ハイブリッド データ セキュリティ トライアルのメンバーになると、そのユーザのクライアントは既存の ECDH 接続をタイムアウトするまで使用し続けます。または、ユーザは Webex アプリ アプリからサインアウトしてから再びサインインすることで、場所を更新し、アプリが暗号キーを照会できるようにすることもできます。

  組織のトライアルを実稼働に移行したときも、同じ現象が発生します。以前のデータ セキュリティ サービスに対する既存の ECDH 接続を使用するすべての非トライアル ユーザは、（タイムアウトまたはサインアウトと再サインインによって）ECDH 接続が再ネゴシエートされるまで、これらのサービスを使用し続けます。

OpenSSL を使用した PKCS12 ファイルの生成

始める前に

- OpenSSL は、HDS セットアップ ツールでの読み込みに適した形式で PKCS12 ファイルを作成するために使用できるツールの 1 つです。他にも使用できる手段はありますが、いずれかの手段をサポートまたは優先することはありません。
- OpenSSL を使用する場合は、「X.509 証明書の要件」で説明している x.509 証明書の要件を満たすファイルを作成できるよう、ガイドラインとして以下の手順に従ってください。ファイルを作成する前に、適用される要件を理解する必要があります。
- サポートされている環境に OpenSSL をインストールします。ソフトウェアおよびドキュメントについては、https://www.openssl.orgを参照してください。
- 秘密キーを作成します。
- 認証局（CA）からサーバ証明書を受け取った後、以下の手順に従います。

手順

ステップ 1 CA からサーバ証明書を受け取ったら、`hdsnode. pem` として保存します。

ステップ 2 証明書をテキストとして表示し、詳細を確認します。

```
openssl x509 -text -noout -in hdsnode.pem
```

ステップ 3 テキスト エディタを使用して、`hdsnode-bundle.pem` という名前の証明書バンドル ファイルを作成します。バンドルファイルには、サーバ証明書、中間 CA 証明書、およびルート CA 証明書が次の形式で含まれている必要があります。

```
-----BEGIN CERTIFICATE-----
### Server certificate. ###
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
###  Intermediate CA certificate. ###
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
###  Root CA certificate. ###
-----END CERTIFICATE-----
```

ステップ 4 フレンドリ名 `kms-private-key` を使用して .p12 ファイルを作成します。

```
openssl pkcs12 -export -inkey hdsnode.key -in hdsnode-bundle.pem -name kms-private-key
-caname kms-private-key -out hdsnode.p12
```

ステップ 5 サーバ証明書の詳細を確認します。

a) `openssl pkcs12 -in hdsnode.p12`

b) プロンプトが表示されたらパスワードを入力して秘密キーを暗号化し、暗号化された状態で出力されるようにします。次に、秘密キーと最初の証明書に **friendlyName: kms-private-key** という行が含まれていることを確認します。

例：

```
bash$ openssl pkcs12 -in hdsnode.p12
Enter Import Password:
MAC verified OK
Bag Attributes
    friendlyName: kms-private-key
    localKeyID: 54 69 6D 65 20 31 34 39 30 37 33 32 35 30 39 33 31 34
Key Attributes: <No Attributes>
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----BEGIN ENCRYPTED PRIVATE KEY-----
<redacted>
-----END ENCRYPTED PRIVATE KEY-----
Bag Attributes
    friendlyName: kms-private-key
    localKeyID: 54 69 6D 65 20 31 34 39 30 37 33 32 35 30 39 33 31 34
subject=/CN=hds1.org6.portun.us
issuer=/C=US/O=Let's Encrypt/CN=Let's Encrypt Authority X3
-----BEGIN CERTIFICATE-----
<redacted>
-----END CERTIFICATE-----
Bag Attributes
    friendlyName: CN=Let's Encrypt Authority X3,O=Let's Encrypt,C=US
subject=/C=US/O=Let's Encrypt/CN=Let's Encrypt Authority X3
issuer=/O=Digital Signature Trust Co./CN=DST Root CA X3
-----BEGIN CERTIFICATE-----
<redacted>
-----END CERTIFICATE-----
```

次のタスク

「Hybrid Data Security の前提条件への対応」に戻ります。「HDS ホストの構成 ISO の作成」では、この `hdsnode.p12` ファイルと、このファイルに設定したパスワードを使用します。

（注） これらのファイルを再利用して、元の証明書の有効期限が切れたときに新しい証明書を要求できます。

トピック 2.1

HDS ノードとクラウド間のトラフィック

メトリック収集のアウトバウンド トラフィック

ハイブリッド データ セキュリティ ノードは特定のメトリックを Webex クラウドに送信します。これには、最大ヒープ、使用ヒープ、CPU 負荷、スレッド カウントに関するシステム メトリック、同期および非同期スレッドのメトリック、暗号化接続、遅延、または要求キュー長のしきい値に関するアラートのメトリック、データストアのメトリック、および暗号化接続のメトリックが含まれます。ノードは、アウトオブバンド（要求とは別の）チャネルを介して暗号化されたキー マテリアルを送信します。

インバウンドトラフィック

ハイブリッド データ セキュリティ ノードは、Webex クラウドから次のタイプのインバウンド トラフィックを受信します。

- 暗号化サービスによってルーティングされるクライアントからの暗号化要求
- ノード ソフトウェアのアップグレード

Hybrid Data Security の Squid プロキシの構成

HTTPS トラフィックを検査する Squid プロキシは、Hybrid Data Security に必要な WebSocket（`wss:`）接続の確立に干渉する場合があります。ここでは、サービスが適切に動作するよう、さまざまなバージョンの Squid で `wss:` トラフィックを無視するように構成する方法を説明します。

Squid 4 および 5

`squid.conf` に `on_unsupported_protocol` ディレクティブを追加します。

```
on_unsupported_protocol tunnel all
```

Squid 3.5.27

次のルールを `squid.conf` に追加して Hybrid Data Security をテストした結果、正しく動作することが確認されています。新しく開発された機能で Webex クラウドが更新されると、これらのルールが変更される可能性があります。

```
acl wssMercuryConnection ssl::server_name_regex mercury-connection

ssl_bump splice wssMercuryConnection

acl step1 at_step SslBump1
acl step2 at_step SslBump2
acl step3 at_step SslBump3
ssl_bump peek step1 all
```

その他の注意事項

```
ssl_bump stare step2 all
ssl_bump bump step3 all
```

その他の注意事項

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。