ASA および ACS における RSA トークン サーバおよび SDI プロトコルの使用

内容

概要

このドキュメントでは、Cisco 適応型セキュリティ アプライアンス（ASA）と Cisco Secure Access Control Server（ACS）と統合できる RSA 認証マネージャのトラブルシューティング手順を説明します。

RSA 認証マネージャは、認証用のワンタイム パスワード（OTP）を提供するソリューションです。パスワードは 1 回だけ使用でき、60 秒ごとに変更されます。これはハードウェアとソフトウェアの両方のトークンをサポートします。

前提条件

要件

次の項目に関する基本的な知識が推奨されます。

- Cisco ASA CLI 設定
- Cisco ACS の設定

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアのバージョンに基づいています。

- Cisco ASA ソフトウェア バージョン 8.4 以降
- Cisco ACS ソフトウェア バージョン 5.3 以降

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。対象のネットワークが実稼働中である場合には、どのようなコマンドについても、その潜在的な影響について確実に理解しておく必要があります。

理論

RSA サーバは、RADIUS または独自 RSA のプロトコルである SDI を使用してアクセスできます。ASA と ACS では、どちらも両方のプロトコル（RADIUS、SDI）を使用して RSA にアクセスできます。

ソフトウェアのトークンが使用されている場合には、RSA は Cisco AnyConnect セキュア モビリティ クライアントに統合できることに注意してください。このドキュメントでは、ASA と ACS の統合に焦点を当てています。AnyConnect の詳細については、『Cisco AnyConnect Secure Mobility Client Administrator Guide, Release 3.1（Cisco AnyConnect セキュア モビリティ クライアント管理者ガイド、リリース 3.1）』の「Using SDI Authentication（SDI 認証の使用）」を参照してください。

RADIUS 経由の RSA

RADIUS には、SDI に比べて 1 つの大きなメリットがあります。RSA では、ユーザに特定のプロファイル（ACS でグループと呼ばれる）を割り当てることができます。これらのプロファイルには、特定の RADIUS 属性が定義されています。認証に成功すると、RSA から返される RADIUS-Accept メッセージにそれらの属性が記載されます。ACS はそれらの属性に基づいて、追加の決定を行います。最も一般的なシナリオでは、ACS グループ マッピングを使用して、ACS の特定のグループに、RSA のプロファイルに関連する特定の RADIUS 属性をマッピングすることを決定します。このロジックにより、承認プロセス全体を RSA から ACS に移動しながら、RSA と同様に詳細なロジックを維持することが可能です。

SDI 経由の RSA

SDI には、RADIUS に比べて 2 つの大きなメリットがあります。1 つは、セッション全体が暗号化されていることです。もう 1 つは、SDI のエージェントが提供する有益なオプションです。それにより、失敗の原因が認証または認可の失敗にあるか、ユーザが見つからなかったためかを判定できます。

この情報は、識別のための操作で ACS で使用されます。たとえば、「user not found」の場合は続行し、「authentication failed」の場合は拒否するなどです。

RADIUS と SDI にはもう 1 つ違いがあります。ASA などのネットワーク アクセス デバイスが SDI を使用する場合、ACS は認証だけを実行します。RADIUS を使用する場合は、ACS が認証、許可、アカウンティング（AAA）を実行します。 ただしこれは大きな違いではありません。同じセッションについて、SDI を認証用に、RADIUS をアカウンティング用に設定することは可能です。

SDI プロトコル

デフォルトでは、SDI は User Datagram Protocol（UDP）5500 を使用します。SDI は、RADIUS キーと同じように対称暗号キーを使用して、セッションを暗号化します。そのキー SDI クライアントごとに異なり、ノード シークレット ファイルに保存されます。ファイルは手動または自動で導入されます。

> **注**：ACS/ASA では手動導入はサポートされていません。

自動導入ノードでは、シークレット ファイルは最初の認証成功後に自動的にダウンロードされます。ノード シークレットはユーザのパスコードとその他の情報から取得したキーで暗号化されます。これはセキュリティ上の問題につながる可能性があるため、最初の認証をローカルで行い、暗号化されたプロトコル（Telnet ではなく Secure Shell（SSH））を使用して、攻撃者がそのファイルを代行受信して復号化できないようにする必要があります。

コンフィギュレーション

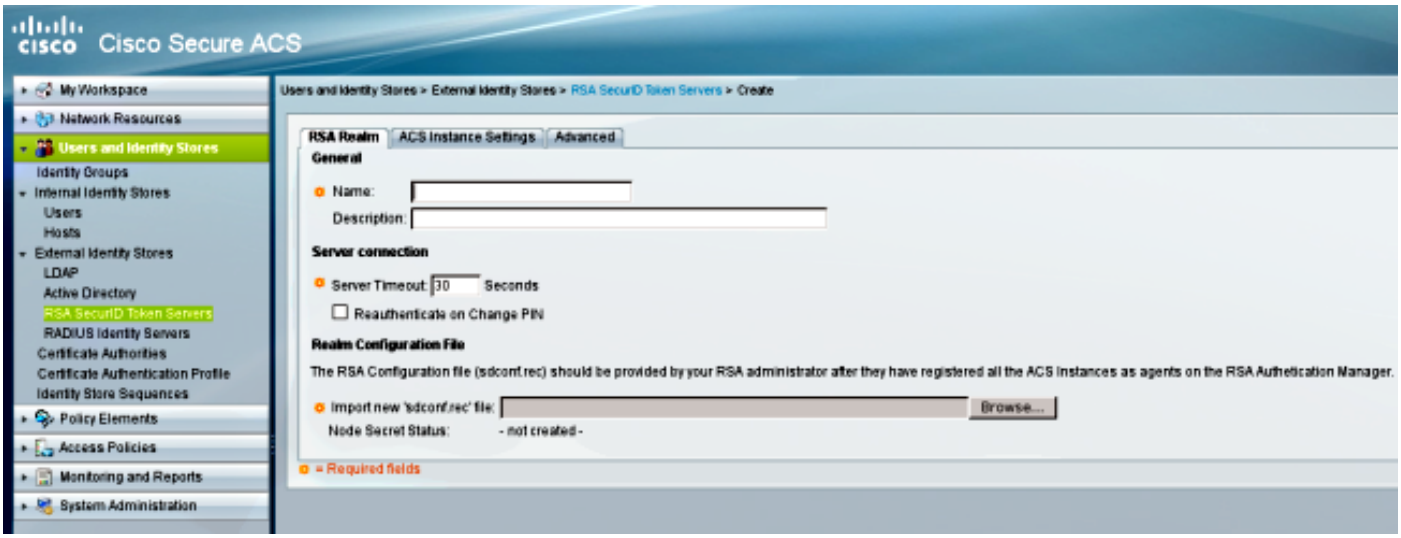
> **注**：
>
> このセクションで使用されるコマンドの詳細については、Command Lookup Tool（登録ユーザ専用）を使用してください。
>
> アウトプット インタープリタ ツール（登録ユーザ専用）は、特定の show コマンドをサポートしています。show コマンドの出力の分析を表示するには、Output Interpreter Tool を使用します。
>
> **debug コマンドを使用する前に、**「デバッグ コマンドの重要な情報」を参照してください。

ACS の SDI

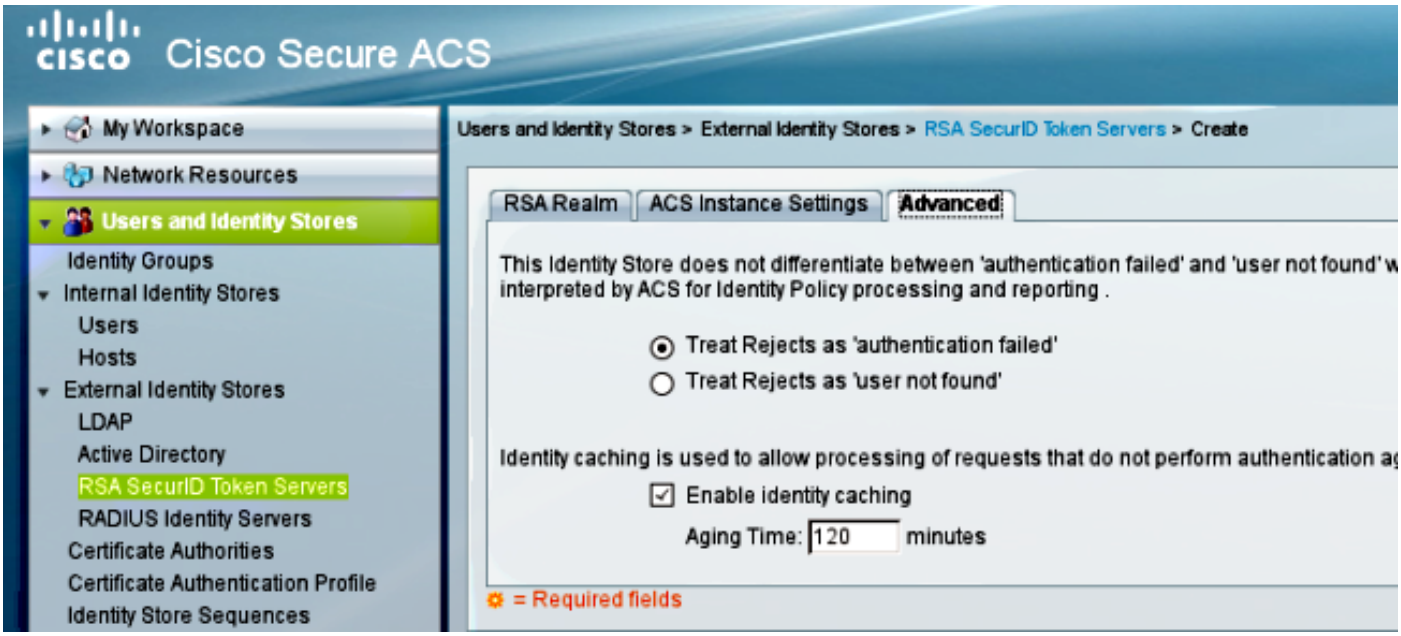
これは、[Users and Identity Stores] > [External Identity Store] > [RSA Secure ID Token Servers] で設定されています。

RSA には、ACS のセカンダリ サーバなど、複数のレプリカ サーバがあります。すべてのアドレスをそのサーバに置く必要はなく、RSA 管理者が提供する **sdconf.rec ファイルだけを配置します**。このファイルには、プライマリ RSA サーバの IP アドレスが含まれています。最初の認証ノー

ドに成功すると、すべての RSA レプリカの IP アドレスとともにシークレット ファイルがダウンロードされます。



「user not found」と「authentication failure」を識別するには、[Advanced] **タブで設定を選択します。**



複数の RSA サーバ（プライマリとレプリカ）間で、デフォルトのルーティング（ロード バランシング）メカニズムを変更することも可能です。 これは、RSA 管理者が提供する **sdopts.rec ファイルを使用して変更します。**ACSでは、[Users and Identity Stores] > [External Identity Store] > [RSA Secure ID Token Servers] > [ACS Instance Settings]にアップロードされます。

クラスタの導入では、設定を複製する必要があります。最初の認証に成功すると、各 ACS ノードは、プライマリ RSA サーバからダウンロードされた独自のノード シークレットを使用します。クラスタ内のすべての ACS ノードの RSA を設定するように注意してください。

ASA の SDI

ASA は **sdconf.rec ファイルのアップロードを許可しません。**ACS と同様に、自動導入のみを許可します。ASA は、プライマリ RSA サーバをポイントするように手動で設定する必要があります。パスワードは不要です。最初の認証ノードに成功すると、シークレット ファイル（フラッシ

ユの .sdi ファイル) がインストールされ、それ以降の認証セッションが保護されます。その他の RSA サーバの IP アドレスもダウンロードされます。

以下が一例です。

```
aaa-server SDI protocol sdi
aaa-server SDI (backbone) host 1.1.1.1
debug sdi 255
test aaa auth SDI host 1.1.1.1 user test pass 321321321
```

認証に成功すると、show aaa-server protocol sdi または show aaa-server <aaa-server-group> コマンドを実行するとすべての RSA サーバ (複数ある場合) が表示され、show run コマンドを実行するとプライマリ IP アドレスだけが表示されます。

```
bsns-asa5510-17# show aaa-server RSA
Server Group:    RSA
Server Protocol: sdi
Server Address:  10.0.0.101
Server port:     5500
Server status:   ACTIVE (admin initiated), Last transaction at
10:13:55 UTC Sat Jul 27 2013
Number of pending requests              0
Average round trip time                 706ms
Number of authentication requests       4
Number of authorization requests        0
Number of accounting requests           0
Number of retransmissions               0
Number of accepts                       1
Number of rejects                       3
Number of challenges                    0
Number of malformed responses           0
Number of bad authenticators            0
Number of timeouts                      0
Number of unrecognized responses        0


SDI Server List:
       Active Address:          10.0.0.101
       Server Address:          10.0.0.101
       Server port:             5500
       Priority:                0
       Proximity:               2
        Status:                  OK
       Number of accepts                     0
       Number of rejects                     0
       Number of bad next token codes        0
       Number of bad new pins sent           0
       Number of retries                     0
       Number of timeouts                    0

       Active Address:          10.0.0.102
       Server Address:          10.0.0.102
       Server port:             5500
       Priority:                8
       Proximity:               2
        Status:                  OK
       Number of accepts                     1
       Number of rejects                     0
       Number of bad next token codes        0
       Number of bad new pins sent           0
```

```
        Number of retries                       0
        Number of timeouts                      0
```

トラブルシュート

ここでは、設定のトラブルシューティングに使用できる情報を示します。

RSA におけるエージェントの未設定

新しい ASA をインストールするか ASA IP アドレスを変更した場合には、RSA でも同じ変更を行うことを忘れがちです。RSA のエージェント IP アドレスは、RSA にアクセスするすべてのクライアントについて更新する必要があります。それによって、新しいノード シークレットが生成されます。ACS、特にセカンダリ ノードの場合も同様です。それぞれ IP アドレスが異なり、RSA がそれらを信頼する必要があるためです。

シークレット ノードの破損

ASA または RSA のシークレット ノード ファイルが破損することがあります。その場合は、RSA のエージェントの設定を削除し、再度追加すべきです。また ASA/ACS でも、設定を再度削除して追加するという同じ処理を行う必要があります。さらに、フラッシュの .sdi ファイルを削除して、次の認証で新しい .sdi ファイルがインストールされるようにします。これが完了すると、ノード シークレットの自動導入が行われます。

待機モードのノード

サーバからの応答がないために、いずれかのノードが一時停止モードになる場合があります。

```
asa# show aaa-server RSA
<.....output ommited"
SDI Server List:
Active Address: 10.0.0.101
Server Address: 10.0.0.101
Server port: 5500
Priority: 0
Proximity: 2
  Status:                  SUSPENDED
```

一時停止モードでは、ASA はそのノードにパケットを送信しません。それについてはステータスが [OK] **になっている必要があります。**障害が発生したサーバは、デッド タイマー後に再度アクティブ モードになります。詳細については、『Cisco ASA Series Command Reference, 9.1 guide (Cisco ASA シリーズ コマンド リファレンス 9.1 ガイド) 』の reactivation-mode コマンドのセクションを参照してください。

このようなシナリオで、サーバを再度アクティブ モードにするには、そのグループ用の AAA サーバ設定を削除して追加することが最適です。

アカウントはロックされています

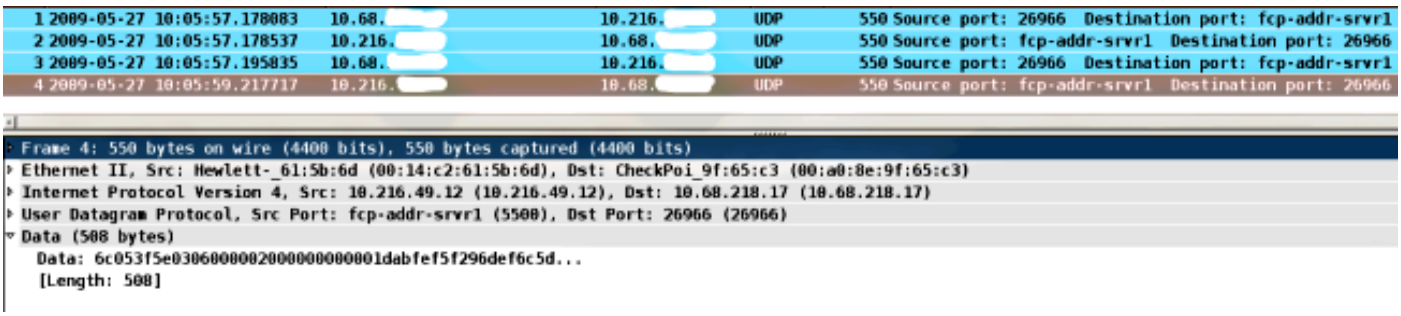
複数回のリトライ後に、RSA がアカウントからロック アウトされる場合があります。これは RSA のレポートで簡単に確認できます。ASA/ACS では、レポートに「failed authentication」とだけ表示されます。

最大伝送ユニット（MTU）の問題およびフラグメンテーション

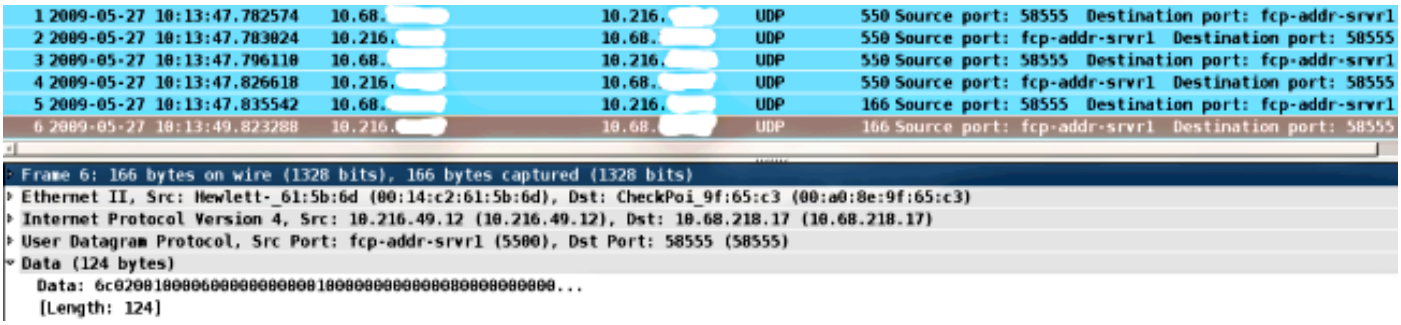
SDI では、UDP が MTU パスの検出ではなくトランスポートとして使用されます。また UDP トラフィックには Don't Fragment（DF; フラグメントなし）ビットがデフォルトでは設定されていません。大きなパケットの場合は、フラグメンテーションの問題が発生する可能性があります。RSA では、トラフィックのスニッフィングを簡単に行うことができます（アプライアンスと仮想マシン（VM）のどちらも Windows と Wireshark を使用）。 ASA/ACS で同じプロセスを実行して比較してください。また、RSA で RADIUS または WebAuthentication のテストを行い、SDI と比較してください（問題を絞り込むため）。

ACS のパケットおよびデバッグ

SDI のペイロードは暗号化されているため、応答のサイズを比較することが、キャプチャのトラブルシューティングを行う唯一の方法です。この値が 200 バイト未満の場合は、問題がある可能性があります。一般的な SDI 交換では、それぞれが 550 バイトの 4 つのパケットが使用されますが、RSA サーバ バージョンでは異なる場合があります。



問題が発生した場合は、多くの場合、サイズの小さいパケットが 5 つ以上交換されています。



また、ACS ログは非常に明瞭です。ACS での SDI の一般的なログを次に示します。

```
EventHandler,11/03/2013,13:47:58:416,DEBUG,3050957712,Stack: 0xa3de560
Calling backRSAIDStore: Method MethodCaller<RSAIDStore, RSAAgentEvent> in
thread:3050957712,EventStack.cpp:242

AuthenSessionState,11/03/2013,13:47:58:416,DEBUG,3050957712,cntx=0000146144,
sesn=acs-01/150591921/1587,user=mickey.mouse,[RSACheckPasscodeState
::onEnterState],RSACheckPasscodeState.cpp:23
```

```
EventHandler,11/03/2013,13:47:58:416,DEBUG,3002137488,Stack: 0xa3de560
Calling RSAAgent:Method MethodCaller<RSAAgent, RSAAgentEvent> in thread:
3002137488,EventStack.cpp:204

RSAAgent,11/03/2013,13:47:58:416,DEBUG,3002137488,cntx=0000146144,sesn=
acs-01/150591921/1587,user=mickey.mouse,[RSAAgent::handleCheckPasscode],
RSAAgent.cpp:319

RSASessionHandler,11/03/2013,13:47:58:416,DEBUG,3002137488,[RSASessionHandler::
checkPasscode] call AceCheck,RSASessionHandler.cpp:251

EventHandler,11/03/2013,13:48:00:417,DEBUG,2965347216,Stack: 0xc14bba0
Create newstack, EventStack.cpp:27

EventHandler,11/03/2013,13:48:00:417,DEBUG,3002137488,Stack: 0xc14bba0 Calling
RSAAgent: Method MethodCaller<RSAAgent, RSAServerResponseEvent> in
thread:3002137488,EventStack.cpp:204

RSAAgent,11/03/2013,13:48:00:417,DEBUG,3002137488,cntx=0000146144,sesn=acs-01
/150591921/1587,user=mickey.mouse,[RSAAgent::handleResponse] operation completed
with ACM_OKstatus,RSAAgent.cpp:237

EventHandler,11/03/2013,13:48:00:417,DEBUG,3002137488,Stack: 0xc14bba0
EventStack.cpp:37

EventHandler,11/03/2013,13:48:00:417,DEBUG,3049905040,Stack: 0xa3de560 Calling
back RSAIDStore: Method MethodCaller<RSAIDStore, RSAAgentEvent> in thread:
3049905040,EventStack.cpp:242

AuthenSessionState,11/03/2013,13:48:00:417,DEBUG,3049905040,cntx=0000146144,sesn=
acs-01/150591921/1587,user=mickey.mouse,[RSACheckPasscodeState::onRSAAgentResponse]
Checkpasscode succeeded, Authentication passed,RSACheckPasscodeState.cpp:55
```

関連情報

- RSA 認証マネージャのリソース
- 『Cisco ASA 5500 シリーズ設定ガイド（CLI、8.4、8.6 使用）』の「RSA/SDI サーバのサポート」セクション
- 『Cisco Secure Access Control System 5.4 ユーザ ガイド）』の「RSA SecurID サーバ」セクション
- テクニカル サポートとドキュメント – Cisco Systems