クライアントレス SSL VPN ユーザ

パスワードの管理

必要に応じて、パスワードの期限切れが近づいたときにエンド ユーザに警告するように ASA を設定できます。

ASA は、RADIUS および LDAP プロトコルのパスワード管理をサポートしています。「password-expire-in-days」オプションは、LDAP に対してのみサポートされます。

IPsec リモート アクセスと SSL VPN トンネルグループのパスワード管理を設定できます。

パスワード管理を設定すると、ASA はリモート ユーザのログイン時に、現在のパスワードの期限切れが近づいていること、または期限が切れていることを通知します。それから ASA は、ユーザがパスワードを変更できるようにします。現行のパスワードが失効していない場合、ユーザはそのパスワードを使用してログインし続けることができます。

このコマンドは、この通知をサポートしている AAA サーバに対して有効です。

ASA のリリース 7.1 以降では、通常、LDAP による認証時または MS-CHAPv2 をサポートする RADIUS コンフィギュレーションによる認証時に、次の接続タイプに対するパスワード管理がサポートされます。

- AnyConnect VPN クライアント
- IPsec VPN クライアント
- クライアントレス SSL VPN

RADIUS サーバ（Cisco ACS など）は、認証要求を別の認証サーバにプロキシする場合があります。ただし、ASA からは RADIUS サーバとのみ通信しているように見えます。

始める前に

- ネイティブ LDAP には、SSL 接続が必要です。LDAP のパスワード管理を実行する前に、SSL 上での LDAP をイネーブルにする必要があります。デフォルトでは、LDAP はポート 636 を使用します。
- 認証に LDAP ディレクトリ サーバを使用している場合、パスワード管理は Sun JAVA System Directory Server（旧名称は Sun ONE Directory Server）および Microsoft Active Directory を使用してサポートされます。
  - Sun：Sun ディレクトリ サーバにアクセスするために ASA に設定されている DN が、サーバのデフォルトパスワードポリシーにアクセスできる必要があります。DN として、ディレクトリ管理者、またはディレクトリ管理者権限を持つユーザを使用することを推奨します。または、デフォルト パスワード ポリシーに ACI を設定できます。
  - Microsoft：Microsoft Active Directory でパスワード管理をイネーブルにするには、LDAP over SSL を設定する必要があります。
- MSCHAP をサポートする一部の RADIUS サーバは、現在 MSCHAPv2 をサポートしていません。このコマンドには MSCHAPv2 が必要なため、ベンダーに問い合わせてください。
- Kerberos/Active Directory（Windows パスワード）または NT 4.0 ドメインでは、これらの接続タイプのいずれについても、パスワード管理はサポートされません。
- LDAP でパスワードを変更するには、市販の LDAP サーバごとに独自の方法が使用されています。現在、ASA では Microsoft Active Directory および Sun LDAP サーバに対してのみ、独自のパスワード管理ロジックを実装しています。
- RADIUS または LDAP 認証が設定されていない場合、ASA ではこのコマンドが無視されます。
- password-management コマンドはパスワードの期限が切れるまでの日数を変更するものではありません。このコマンドは、ASA がパスワードの期限が近いことについてユーザへの警告を開始する、期限切れ前の日数を変更します。

手順

ステップ 1 一般属性モードに切り替えます。

tunnel-group general-attributes

ステップ 2 パスワードの期限切れが近づいていることをリモート ユーザに通知します。

password-management password-expire-in-days *days*

例：

```
hostname(config-general)# password-management password-expire-in-days 90
```

- password-expire-in-days キーワードを指定する場合は、日数も指定する必要があります。

- 日数を 0 に設定すると、このコマンドはオフになります。

この例では、ASA が有効期限の 90 日前にユーザへのパスワードの期限切れの警告を開始します。

（注） password-expire-in-days キーワードが設定されていない場合、ASA は期限切れが近いことをユーザに通知しませんが、ユーザは期限が切れた後にパスワードを変更できます。

クライアントレス SSL VPN でのシングル サインオンの使用

シングル サインオンのサポートを使用すると、クライアントレス SSL VPN のユーザは、ユーザ名とパスワードを 1 回入力するだけで、保護された複数のサービスや Web サーバにアクセスできます。一般に、SSO のメカニズムは AAA プロセスの一部として開始されるか、または AAA サーバのユーザ認証に成功した直後に開始されます。ASA で実行されるクライアントレス SSL VPN サーバは、認証サーバへのユーザ用プロキシとして動作します。ユーザがログインすると、クライアントレス SSL VPN サーバは、ユーザ名とパスワードを含む SSO 認証要求を認証サーバに送信します。サーバが認証要求を受け入れた場合は、クライアントレス SSL VPN サーバに SSO 認証クッキーを戻します。ASA は、ユーザの代わりにこのクッキーを保持し、ユーザの認証にこのクッキーを使用して、SSO サーバにより保護されているドメイン内の Web サイトの安全を確保します。

SiteMinder による SSO 認証の設定

この項では、SiteMinder を使用して SSO をサポートするための ASA の設定について説明します。ユーザの Web サイトのセキュリティ インフラストラクチャにすでに SiteMinder を組み込んでいる場合は、SiteMinder を使用して SSO を実装するのが一般的です。この方式では、SSO 認証は AAA とは分離され、AAA プロセスが完了するとこの認証が 1 回行われます。

始める前に

- SSO サーバの指定。
- ASA が SSO 認証要求を作成する SSO サーバの URL を指定。
- ASA と SSO サーバとの間でセキュアな通信を確立するための秘密キーを指定。このキーはパスワードのようなもので、ユーザが作成して保存し、Cisco Java プラグイン認証スキームを使用して ASA と SiteMinder ポリシー サーバの両方に入力します。

これらの必須のタスクに加えて、次のようなオプションの設定タスクを行うことができます。

- 認証要求のタイムアウトの設定。

- 認証要求のリトライ回数の設定。

（注） クライアントレス SSL VPN アクセスを行うユーザまたはグループに SSO を設定するには、まず RADIUS サーバや LDAP サーバなどの AAA サーバを設定する必要があります。

手順

ステップ 1 クライアントレス SSL VPN コンフィギュレーション モードに切り替えます。

webvpn

ステップ 2 Example of type siteminder という名前の SSO サーバを作成します。

sso-server *name* **type** *type*

例：

```
ciscoasa(config-webvpn)# sso-server Example type siteminder
```

ステップ 3 SiteMinder コンフィギュレーション モードに切り替えます。

config-webvpn-sso-siteminder

ステップ 4 SSO サーバの認証 URL を http://www.Example.com/webvpn として指定します。

web-agent-url

例：

```
hostname(config-webvpn-sso-siteminder)# web-agent-url http://www.Example.com/webvpn
```

ステップ 5 ASA と SiteMinder との間でセキュアな認証通信を確立するための秘密キーを指定します。

policy-server-secret *secret*

キーの長さは、標準またはシフト式英数字を使用した任意の文字長にできますが、ASA と SSO サーバの両方で同じキーを入力する必要があります。

例：

秘密キー AtaL8rD8! を作成します。

```
hostname(config-webvpn-sso-siteminder)# policy-server-secret AtaL8rD8!
```

ステップ 6 失敗した SSO 認証試行をタイムアウトさせるまでの秒数を設定します。

request-timeout *seconds*

デフォルトの秒数は 5 で、1 ～ 30 秒までの範囲で指定できます。

例：

この例では、要求がタイムアウトするまでの秒数を 8 に変更します。

```
hostname(config-webvpn-sso-siteminder)# request-timeout 8
```

ステップ 7 認証がタイムアウトするまでに ASA が失敗した SSO 認証を再試行する回数を設定します。

max-retry-attempts *number*

デフォルトの再試行回数は 3 で、1 回から 5 回までの範囲で指定できます。

例：

この例では、再試行回数を 4 に設定します。

```
hostname(config-webvpn-sso-siteminder)# max-retry-attempts 4
```

ステップ 8 グループまたはユーザの SSO 認証を指定します。

- ユーザの認証を指定する場合。

  username-webvpn

- グループの認証を指定する場合。

  group-policy-webvpn

ステップ 9 ユーザに SSO サーバを割り当てます。

sso-server value *value*

例：

```
hostname(config)# username Anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# sso-server value Example
```

この例では、Example という名前の SSO サーバを Anyuser という名前のユーザに割り当てます。

ステップ 10 SSO サーバの設定をテストします。

test sso-server *server* **username** *username*

例：

この例では、Example という名前の SSO サーバをユーザ名 Anyuser を使用してテストします。

```
hostname# test sso-server Example username Anyuser
INFO: Attempting authentication request to sso-server Example for user Anyuser
INFO: STATUS: Success
hostname#
```

シスコの認証スキームの SiteMinder への追加

SiteMinder による SSO を使用するための ASA の設定に加え、Java プラグインとして提供されているシスコの認証スキーム（シスコの Web サイトからダウンロード）を使用するようにユーザの CA SiteMinder ポリシー サーバを設定する必要もあります。

始める前に

SiteMinder ポリシー サーバを設定するには、SiteMinder の経験が必要です。

手順

ステップ 1 SiteMinder Administration ユーティリティを使用して、次の特定の引数を使用できるようにカスタム認証スキームを作成します。

- Library フィールドに、**smjavaapi** と入力します。
- [Secret] フィールドに、ASA に設定したものと同じ秘密キーを入力します。

コマンドライン インターフェイスで **policy-server-secret** コマンドを使用して、ASA に秘密キーを設定します。

- Parameter フィールドに、**CiscoAuthAPI** と入力します。

ステップ 2 Cisco.com にログインして、http://www.cisco.com/cisco/software/navigator.html から **cisco_vpn_auth.jar** ファイルをダウンロードして、SiteMinder サーバのデフォルトのライブラリ ディレクトリにコピーします。この .jar ファイルは、Cisco ASA CD にも含まれています。

SAML Browser Post Profile を使用した SSO 認証の設定

この項では、認可されたユーザに対して、Security Assertion Markup Language（SAML）、バージョン 1.1 POST プロファイル シングル サインオン（SSO）をサポートするための ASA の設定について説明します。

セッション開始後、ASA は設定されている AAA 方式に対してユーザを認証します。次に、ASA（アサーティング パーティ）は、SAML サーバが提供するコンシューマ URL サービスであるリライング パーティに対してアサーションを生成します。SAML の交換が成功すると、ユーザは保護されているリソースへのアクセスを許可されます。

始める前に

SAML Browser Post Profile を使用して SSO を設定するには、次のタスクを実行する必要があります。

- **sso-server** コマンドを使用した SSO サーバの指定
- 認証要求を行うための SSO サーバの URL の指定（**assertion-consumer-url** コマンド）
- 認証要求を発行するコンポーネントとして ASA ホスト名を指定（issuer コマンド）
- SAML Post Profile アサーションの署名に使用するトラストポイント証明書の指定（**trustpoint** コマンド）

これらの必須タスクに加えて、次のようなオプションの設定タスクを行うことができます。

- 認証要求のタイムアウトの設定（**request-timeout** コマンド）
- 認証要求のリトライ回数の設定（**max-retry-attempts** コマンド）
- SAML SSO は、クライアントレス SSL VPN セッション対してのみサポートされています。
- 現在 ASA は、SAML SSO サーバの Browser Post Profile タイプのみをサポートしています。
- SAML Browser Artifact プロファイル方式のアサーション交換はサポートされていません。

手順

ステップ 1 クライアントレス SSL VPN コンフィギュレーション モードに切り替えます。

webvpn

ステップ 2 SSO サーバを作成します。

sso-server type type

例：

```
hostname(config-webvpn)# sso-server sample type SAML-V1.1-post
```

この例では、タイプが SAML-V1.1-POST の Sample という名前の SSO サーバを作成します。

ステップ 3 クライアントレス SSL VPN sso-saml コンフィギュレーション モードに切り替えます。

sso saml

ステップ 4 SSO サーバの認証 URL を指定します。

assertion-consumer-url url

例：

```
hostname(config-webvpn-sso-saml)# assertion-consumer-url http://www.example.com/webvpn
hostname(config-webvpn-sso-saml)#
```

この例では、http://www.Example.com/webvpn という URL に認証要求を送信します。

ステップ 5 ASA でアサーションを生成する場合は、ASA 自体を識別します。

issuer string

例：

```
hostname(config-webvpn-sso-saml)# issuer myasa
```

通常、この issuer 名は ASA のホスト名になります。

ステップ 6 アサーションに署名するための ID 証明書を指定します。

trust-point

例：

```
hostname(config-webvpn-sso-saml)# trust-point mytrustpoint
```

ステップ 7 失敗した SSO 認証試行をタイムアウトさせるまでの秒数を設定します。

request-timeout

例：

```
hostname(config-webvpn-sso-saml)# request-timeout 8
```

この例では、要求がタイムアウトするまでの秒数を 8 に設定します。

デフォルトの秒数は 5 で、1 秒から 30 秒までの範囲で指定できます。

ステップ 8 認証がタイムアウトするまでに ASA が失敗した SSO 認証を再試行する回数を設定します。

max-retry-attempts

例：

```
hostname(config-webvpn-sso-saml)# max-retry-attempts 4
```

この例では、再試行回数を 4 に設定します。

デフォルトの再試行回数は 3 で、1 回から 5 回までの範囲で指定できます。

ステップ 9 クライアントレス SSL VPN コンフィギュレーション モードに切り替えます。

webvpn

SSO サーバをグループ ポリシーに割り当てる場合。

group-policy-webvpn

SSO サーバをユーザ ポリシーに割り当てる場合。

username-webvpn

ステップ 10 グループまたはユーザの SSO 認証を指定します。

sso-server value

例：

```
hostname(config)# username Anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# sso-server value sample
```

この例では、Example という名前の SSO サーバを Anyuser という名前のユーザに割り当てます。

ステップ 11 （特権 EXEC モード）SSO サーバの設定をテストします。

test sso-server

例：

```
hostname# test sso-server Example username Anyuser
INFO: Attempting authentication request to sso-server sample for user Anyuser
INFO: STATUS: Success
```

この例では、SSO サーバ Example をユーザ名 Anyuser を使用してテストします。

SAML POST SSO サーバの設定

サーバ ソフトウェア ベンダーが提供する SAML サーバのマニュアルに従って、SAML サーバを Relying Party モードで設定します。

手順

ステップ 1 アサーティング パーティ（ASA）を表す SAML サーバ パラメータを設定します。

- Recipient consumer URL（ASA で設定されるアサーション コンシューマ URL と同じ）
- Issuer ID（通常はアプライアンスのホスト名である文字列）
- Profile type：Browser Post Profile

ステップ 2 証明書を設定します。

ステップ 3 アサーティング パーティのアサーションには署名が必要なことを指定します。

ステップ 4 SAML サーバがユーザを特定する方法を、次のように選択します。

- Subject Name Type が DN
- Subject Name format が uid=<user>

HTTP Basic 認証または NTLM 認証による SSO の設定

この項では、HTTP Basic 認証または NTLM 認証を使用するシングル サインオンについて説明します。この方法のいずれかまたは両方を使用して SSO を実装するように ASA を設定することができます。**auto-sign-on** コマンドを使用すると、ASA はクライアントレス SSL VPN ユーザのログイン クレデンシャル（ユーザ名およびパスワード）を内部サーバに自動的に渡すように設定されます。複数の **auto-sign-on** コマンドを入力できます。ASA は複数のコマンドを入力順に処理します（先に入力されたコマンドを優先）。IP アドレスと IP マスク、または URI マスクのいずれかを使用してログインのクレデンシャルを受信するようにサーバに指定します。

クライアントレス SSL VPN コンフィギュレーション、クライアントレス SSL VPN グループ ポリシー モード、またはクライアントレス SSL VPN ユーザ名モードの 3 つのモードのいずれか

で、**auto-sign-on** コマンドを使用します。ユーザ名はグループより優先され、グループはグローバルより優先されます。認証に必要な範囲のモードを選択します。

モード	スコープ
webvpn configuration	クライアントレス SSL VPN ユーザ全員に対するグローバルな範囲
webvpn group-policy configuration	グループ ポリシーで定義されるクライアントレス SSL VPN ユーザのサブセット
webvpn username configuration	個々のクライアントレス SSL VPN ユーザ

例

- NTLM 認証を使用し、10.1.1.0 ～ 10.1.1.255 の IP アドレス範囲に存在するサーバに対するすべてのクライアントレス SSL VPN ユーザからのアクセスに auto-sign-on を設定します。

```
hostname(config-webvpn)# auto-sign-on allow ip 10.1.1.1 255.255.255.0 auth-type
ntlm
```

- 基本の HTTP 認証を使用するすべてのクライアントレス SSL VPN ユーザに対し、URI マスク https://*.example.com/* で定義されたサーバへのアクセスに auto-sign-on を設定します。

```
hostname(config-webvpn)# auto-sign-on allow uri https://*.example.com/* auth-type
```

- 基本認証または NTLM 認証を使用して、ExamplePolicy グループ ポリシーと関連付けられているクライアントレス SSL VPN セッションに対し、URI マスクで定義されたサーバへのアクセスに auto-sign-on を設定します。

```
hostname(config)# group-policy ExamplePolicy attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# auto-sign-on allow uri https://*.example.com/* auth-type
 all
```

- *Anyuser* というユーザが IP アドレス範囲 10.1.1.0 ～ 10.1.1.255 のサーバに、HTTP 基本認証によって自動サインオンするように設定します。

```
hostname(config)# username Anyuser attributes
hostname(config-username)# webvpn
hostname(config-username-webvpn)# auto-sign-on allow ip 10.1.1.1 255.255.255.0
auth-type basic
```

- 特定のポートで自動サインオンを設定し、認証のレルムを設定します。

```
smart-tunnel auto-sign-on host-list [use-domain] [realm realm string] [port port
num] [host host mask | ip address subnet mask]
```

HTTP Form プロトコルによる SSO の設定

この項では、SSO における HTTP Form プロトコルの使用について説明します。HTTP Form プロトコルは、SSO 認証を実行するための手段で、AAA 方式としても使用できます。このプロトコルは、クライアントレス SSL VPN のユーザおよび認証を行う Web サーバの間で認証情報を交換するセキュアな方法を提供します。RADIUS サーバや LDAP サーバなどの他の AAA サーバと組み合わせて使用することができます。

ASA は、ここでも認証 Web サーバに対するクライアントレス SSL VPN ユーザのプロキシとして機能しますが、この場合は、要求に対して HTTP Form プロトコルと POST 方式を使用します。フォーム データを送受信するように ASA を設定する必要があります。

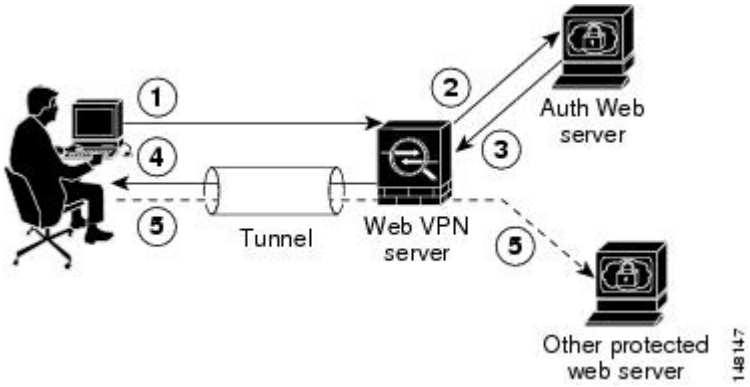
HTTP プロトコルを使用して SSO を正しく設定するには、認証と HTTP プロトコル交換についての詳しい実務知識が必要です。

これは、一般的なプロトコルとして、認証に使用する Web サーバ アプリケーションの次の条件に一致する場合にだけ適用できます。

- 認証クッキーは、正常な要求に対して設定され、未許可のログインに対して設定されないようにする必要があります。この場合、ASA は、失敗した認証から正常な要求を識別することはできません。

次の図は、後述する SSO 認証手順を示しています。

図 *1 : HTTP Form* を使用した *SSO* 認証



1. クライアントレス SSL VPN のユーザは、最初にユーザ名とパスワードを入力して ASA 上のクライアントレス SSL VPN サーバにログオンします。
2. ユーザのプロキシとして動作するクライアントレス SSL VPN サーバは、このフォーム データ（ユーザ名およびパスワード）を、POST 認証要求を使用して認証 Web サーバに転送します。
3. 認証 Web サーバがユーザのデータを承認した場合は、認証クッキーをユーザの代行で保存していたクライアントレス SSL VPN サーバに戻します。
4. クライアントレス SSL VPN サーバはユーザまでのトンネルを確立します。
5. これでユーザは、ユーザ名やパスワードを再入力しなくても、保護された SSO 環境内の他の Web サイトにアクセスできるようになります。

ユーザ名やパスワードなどの POST データを ASA によって含めるようにフォーム パラメータを設定しても、Web サーバに必要な非表示のパラメータが追加されたことに、当初、ユーザは気づかない可能性があります。認証アプリケーションの中には、ユーザ側に表示されず、ユーザが入力することもない非表示データを要求するものもあります。ただし、ASA を仲介役のプロキシとして使用せずに、ブラウザから Web サーバに直接認証要求を行うことによって、認証 Web サーバに必要な非表示のパラメータを見つけることができます。HTTP ヘッダー アナライザを使用して Web サーバの応答を分析すると、非表示パラメータが次のような形式で表示されます。

```
<param name>=<URL encoded value>&<param name>=<URL encoded>
```

非表示パラメータには、必須のパラメータとオプションのパラメータとがあります。Web サーバが非表示パラメータのデータを要求すると、Web サーバはそのデータを省略するすべての認証 POST 要求を拒否します。ヘッダーアナライザは、非表示パラメータが必須かオプションかについては伝えないため、必須のパラメータが判別できるまではすべての非表示パラメータを含めておくことをお勧めします。

HTTP Form プロトコルを使用した SSO を設定するには、次を実行する必要があります。

- フォーム データ（**action-uri**）を受信して処理するために、認証 Web サーバにユニフォーム リソース識別子を設定する。
- ユーザ名パラメータ（**user-parameter**）を設定する。
- ユーザ パスワード パラメータ（**password-parameter**）を設定する。

認証 Web サーバの要件によっては次のタスクが必要になる場合もあります。

- 認証 Web サーバがログイン前のクッキー交換を必要とする場合は、開始 URL（**start-url**）を設定する。
- 認証 Web サーバが必要とするあらゆる非表示認証パラメータ（**hidden-parameter**）を設定する。
- 認証 Web サーバによって設定される認証クッキーの名前（**auth-cookie-name**）を設定する。

手順

ステップ 1 AAA サーバ ホスト コンフィギュレーション モードに切り替えます。

aaa-server-host

ステップ 2 認証 Web サーバが要求する場合は、認証 Web サーバから事前ログイン クッキーを取得するための URL を指定します。

start-url

例：

```
hostname(config)# aaa-server testgrp1 protocol http-form
hostname(config)# aaa-server testgrp1 host 10.0.0.2
hostname(config-aaa-server-host)# start-url http://example.com/east/Area.do?Page-Grp1
```

この例では、http://example.com/east/Area.do?Page-Grp1 の URL 認証 Web サーバを、IP アドレス 10.0.0.2 の testgrp1 サーバ グループに指定します。

ステップ 3 認証 Web サーバ上の認証プログラムの URI を指定します。

action-uri

例：

```
http://www.example.com/auth/index.html/appdir/authc/forms/MCOlogin.fcc?TYPE=33554433
&REALMOID=06-000a1311-a828-1185-ab41-8333b16a0008&GUID=&SMAUTHREASON=0&METHOD=GET&SMAGENTNA
ME=$SM$5FZmjnk3DRNwNjk2KcqVCFbIrNT9%2bJ0H0KPshFtg6rB1UV2PxkHqLw%3d%3d&TARGET=https%3A%2F%2F
auth.example.com
```

この action URI を指定するには、次のコマンドを入力します。

```
hostname(config-aaa-server-host)# action-uri http://www.example.com/auth/index.htm
hostname(config-aaa-server-host)# action-uri l/appdir/authc/forms/MCOlogin.fcc?TYP
hostname(config-aaa-server-host)# action-uri 554433&REALMOID=06-000a1311-a828-1185
hostname(config-aaa-server-host)# action-uri -ab41-8333b16a0008&GUID=&SMAUTHREASON
hostname(config-aaa-server-host)# action-uri =0&METHOD=GET&SMAGENTNAME=$SM$5FZmjnk
hostname(config-aaa-server-host)# action-uri 3DRNwNjk2KcqVCFbIrNT9%2bJ0H0KPshFtg6r
hostname(config-aaa-server-host)# action-uri B1UV2PxkHqLw%3d%3d&TARGET=https%3A%2F
hostname(config-aaa-server-host)# action-uri %2Fauth.example.com
```

1 つの URI を連続する複数行にわたって入力することができます。1 行あたりの最大文字数は 255 です。URI 全体の最大文字数は 2048 です。

アクション URI にホスト名とプロトコルを含める必要があります。この例では、これらは http://www.example.com の URI の最初に表示されます。

ステップ 4 HTTP POST 要求の userid ユーザ名パラメータを設定します。

user-parameter

例：

```
hostname(config-aaa-server-host)# user-parameter userid
```

ステップ 5 HTTP POST 要求の user_password ユーザ パスワード パラメータを設定します。

password-parameter

例：

```
hostname(config-aaa-server-host)# password-parameter user_password
```

ステップ 6 認証 Web サーバと交換するための非表示パラメータを指定します。

hidden-parameter

例：

```
hostname(config)# aaa-server testgrp1 host example.com
hostname(config-aaa-server-host)# hidden-parameter SMENC=ISO-8859-1&SMLOCALE=US-EN&targe
hostname(config-aaa-server-host)# hidden-parameter t=https%3A%2F%2Fwww.example.com%2Femc
hostname(config-aaa-server-host)# hidden-parameter o%2Fappdir%2FAreaRoot.do%3FEMCOPageCo
hostname(config-aaa-server-host)# hidden-parameter de%3DENG&smauthreason=0
```

この例では、POST 要求から抜粋した非表示パラメータの例を示します。この非表示パラメータには、間を & で区切った 4 つの Form エントリとその値が含まれています。エントリとその値は次のとおりです。

- SMENC、値は ISO-8859-1。
- SMLOCALE、値は US-EN。
- target、値は https%3A%2F%2Fwww.example.com%2Femco%2Fappdir%2FAreaRoot.do。
- %3FEMCOPageCode%3DENG。
- smauthreason、値は 0。

ステップ 7 認証クッキーの名前を指定します。

auth-cookie-name *cookie-name*

例：

```
hostname(config-aaa-server-host)# auth-cookie-name SsoAuthCookie
```

この例では、SsoAuthCookie の認証クッキー名を指定します。

ステップ 8 トンネル グループ一般属性コンフィギュレーション モードに切り替えます。

tunnel-group general-attributes

ステップ 9 前の手順で設定された SSO サーバを使用するためのトンネル グループを設定します。

authentication-server-group

例：

```
hostname(config)# tunnel-group testgroup general-attributes
hostname(config-tunnel-general)#authentication-server-group testgrp1
```

この例では、/testgrp1/ という名前の SSO サーバを使用するための、/testgroup/ という名前のトンネル グループを設定します。

ステップ 10 AAA サーバ ホスト コンフィギュレーション モードに切り替えます。

aaa-server-host

ステップ 11 認証クッキーの名前を指定します。

auth-cookie-name *cookie-name*

例：

```
hostname(config-aaa-server-host)# auth-cookie-name SsoAuthCookie
```

この例では、SsoAuthCookie の認証クッキー名を指定します。

ステップ 12 トンネル グループ一般属性モードに切り替えます。

tunnel-group general-attributes

ステップ 13 前の手順で設定された SSO サーバを使用するためのトンネル グループを設定します。

authentication-server-group group

例：

```
hostname(config)# tunnel-group testgroup general-attributes
hostname(config-tunnel-general)#authentication-server-group testgrp1
```

この例では、/testgrp1/ という名前の SSO サーバを使用するための、/testgroup/ という名前のトンネル グループを設定します。

HTTP Form データの収集

この項では、必要な HTTP Form データを検出および収集する手順を示します。認証 Web サーバが要求するパラメータが何かわからない場合は、認証交換を分析するとパラメータ データを収集することができます。

始める前に

これらの手順では、ブラウザと HTTP ヘッダー アナライザが必要です。

手順

ステップ 1 ブラウザと HTTP ヘッダー アナライザを起動し、ASA を経由せずに、Web サーバのログイン ページに直接接続します。

ステップ 2 Web サーバのログイン ページがユーザのブラウザにロードされてから、ログイン シーケンスを検証して交換時にクッキーが設定されているかどうか判別します。Web サーバによってログイン ページにクッキーがロードされている場合は、このログイン ページの URL を *start-URL* として設定します。

ステップ 3 Web サーバにログオンするためのユーザ名とパスワードを入力して、Enter を押します。この動作によって、ユーザが検証する認証 POST 要求が HTTP ヘッダー アナライザを使用して生成されます。

次に、ホストの HTTP ヘッダーおよび本文が記載された POST 要求の例を示します。

```
POST
/emco/myemco/authc/forms/MCOlogin.fcc?TYPE=33554433&REALMOID=06-000430e1-7443-125c
-ac05-83846dc90034&GUID=&SMAUTHREASON=0&METHOD=GET&SMAGENTNAME=$SM$5FZmjnk3DRNwNjk
2KcqVCFbIrNT9%2bJ0H0KPshFtg6rB1UV2PxkHqLw%3d%3d&TARGET=https%3A%2F%2Fwww.example.c
om%2Femco%2Fmyemco%2FHTTP/1.1

Host: www.example.com

(BODY)

SMENC=ISO-8859-1&SMLOCALE=US-EN&USERID=Anyuser&USER_PASSWORD=XXXXXX&target=https
%3A%2F%2Fwww.example.com%2Femco%2Fmyemco%2F&smauthreason=0
```

ステップ 4 POST 要求を検証してプロトコル、ホストをコピーし、URL を入力して action-uri パラメータを設定します。

ステップ 5 POST 要求の本文を検証して、次の情報をコピーします。

a) ユーザ名パラメータ。上記の例では、このパラメータは *USERID* で、値 *anyuser* ではありません。

b) パスワード パラメータ。上記の例では、このパラメータは *USER_PASSWORD* です。

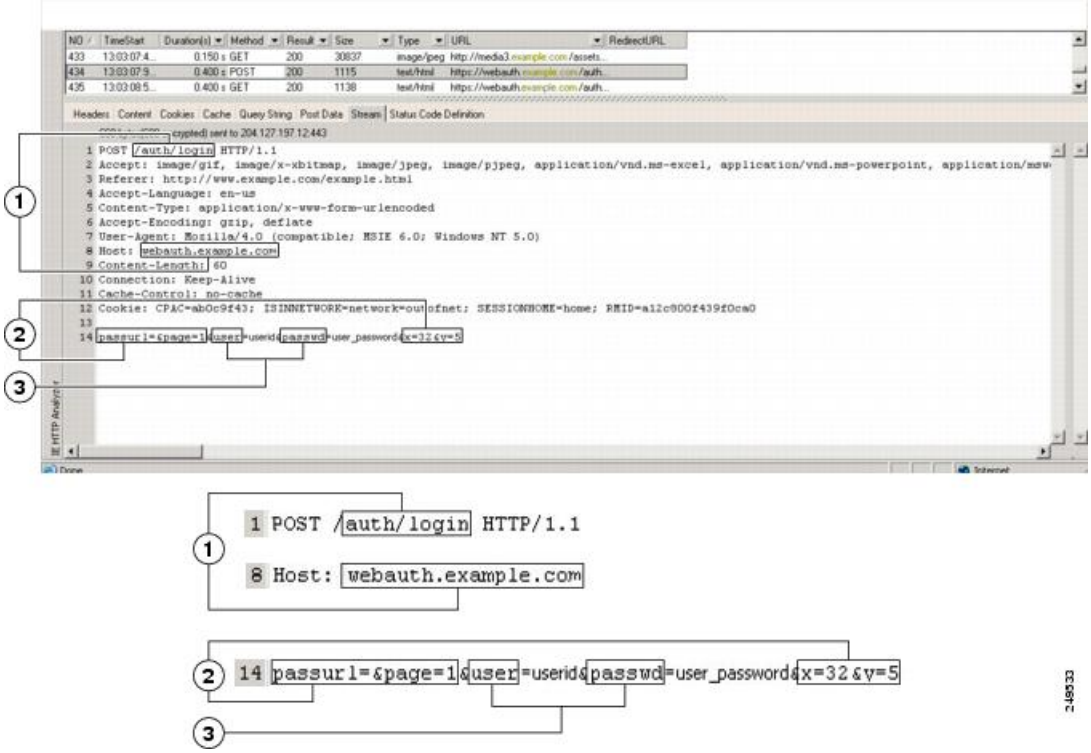
c) 非表示パラメータ。

このパラメータは、POST 本文からユーザ名パラメータとパスワード パラメータを除くすべてです。前の例の非表示パラメータは次のとおりです。

```
SMENC=ISO-8859-1&SMLOCALE=US-EN&target=https%3A%2F%2Fwww.example.com%2
Femco%2Fmyemco%2F&smauthreason=0
```

次の図は、HTTP アナライザの出力例におけるアクション URI、非表示、ユーザ名、パスワードの各種パラメータを強調して示しています。これは一例にすぎません。出力は Web サイトに応じて大きく異なります。

図 *2:* アクション *URI*、非表示、ユーザ名、パスワードの各種パラメータ



1	action URI パラメータ
2	非表示パラメータ
3	ユーザ名パラメータとパスワード パラメータ

ステップ 6 Web サーバへのログオンに成功したら、HTTP ヘッダー アナライザを使用してサーバの応答を検証し、サーバによってブラウザに設定されたセッション クッキーの名前を探します。これは、**auth-cookie-name** パラメータです。

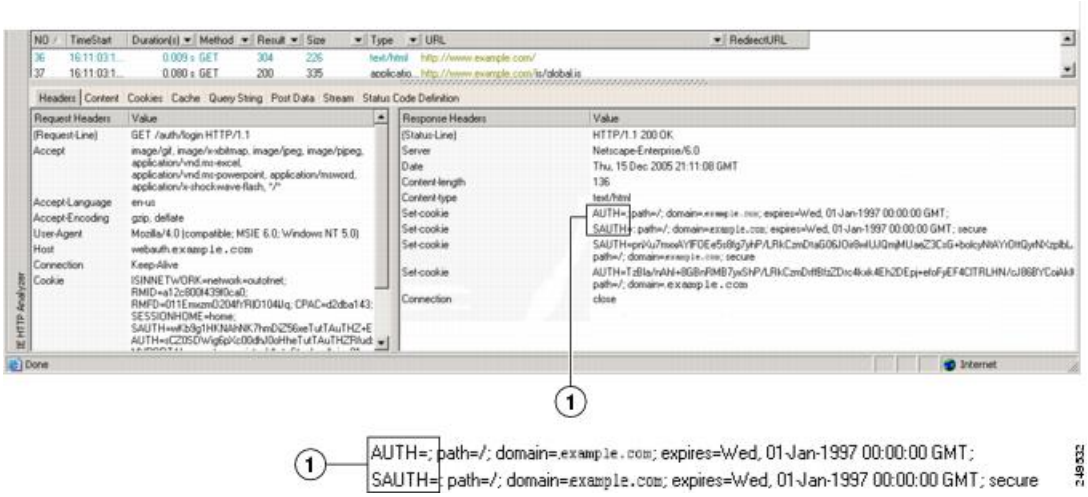
次のサーバ応答ヘッダーでは、SMSESSION がセッションのクッキーの名前です。必要なのはこの名前だけです。値は不要です。

```
Set-Cookie:
SMSESSION=yN4Yp5hHVNDgs4FT8dn7+Rwev41hsE49XlKc+1twie0gqnjbhkTkUnR8XWP3hvDH6PZ
PbHIHtWLDKTa8ngDB/lbYTjIxrbDx8WPWwaG3CxVa3adOxHFR8yjD55GevK3ZF4ujgU1lhO6fta0d
SSOSepWvnsCb7IFxCw+MGiw0o88uHa2t4l+SillqfJvcpuXfiIAO06D/gtDF40Ow5YKHEl2KhDEvv
+yQzxwfEz2cl7Ef5iMr8LgGcDK7qvMcvrgUqx68JQOK2+RSwtHQ15bCZmsDU5vQVCvSQWC8OMHNGw
pS253XwRLvd/h6S/tM0k98QMv+i3N8oOdj1V7flBqecH7+kVrU01F6oFzr0zM1kMyLr5HhlVDh7B0
k9wp0dUFZiAzaf43jupD5f6CEkuLeudYW1xgNzsR8eqtPK6t1gFJyOn0s7QdNQ7q9knsPJsekRAH9
hrLBhWBLTU/3B1QS94wEGD2YTuiW36TiP14hYwOlCAYRj2/bY3+lYzVu7EmzMQ+UefYxh4cF2gYD8
RZL2RwmP9JV5148I3XBFPNUw/3V5jf7nRuLr/CdfK3OO8+Pa3V6/nNhokErSgyxjzMd88DVzM41Lx
xaUDhbcmkoHT9ImzBvKzJX0J+o7FoUDFOxEdIqlAN4GNqk49cpi2sXDbIarALp6Bl3+tbB4MlHGH+
0CPscZXqoi/kon9YmGauHyRs+0m6wthdlAmCnvlJCDfDoXtn8DpabgiW6VDTrvl3SGPyQtUv7Wdah
uq5SxbUzjY2JxQnrUtwB977NCzYu2sOtN+dsEReWJ6ueyJBbMzKyzUB4L3i5uSYN50B4PCv1w5KdR
Ka5p3N0Nfq6RM6dfipMEJw0Ny1sZ7ohz3fbvQ/YZ7lw/k7ods/8VbaR15ivkE8dSCzuf/AInHtCzu
```

```
Q6wApzEp9CUoG8/dapWriHjNoi4llJOgCst33wEhxFxcWy2UWxs4EZSjsI5GyBnefSQTPVfma5dc/
emWor9vWr0HnTQaHP5rg5dTNqunkDEdMIHfbeP3F90cZejVzihM6igiS6P/CEJAjE;Domain=.exa
mple.com;Path=/
```

次の図は、HTTP アナライザの出力における許可クッキーの例を示しています。これは一例にすぎません。出力は Web サイトに応じて大きく異なります。

図 *3: HTTP* アナライザの出力例における認可クッキー



1	認可クッキー

ステップ 7 場合によっては、認証の成否にかかわらず同じクッキーがサーバによって設定される可能性があり、このようなクッキーは、SSO の目的上、認められません。クッキーが異なっていることを確認するには、無効なログインクレデンシャルを使用してステップ 1 ～ 6 を繰り返し、「失敗した」クッキーと「成功した」クッキーを比較します。これで、HTTP Form プロトコルによる SSO を ASA に設定するために必要なパラメータ データを用意できました。

プラグインの SSO の設定

プラグインは、シングルサインオン（SSO）をサポートします。プラグインは、クライアントレス SSL VPN セッションを認証するときに入力したクレデンシャルと同じクレデンシャル（ユーザ名とパスワード）を使用します。プラグインはマクロ置換をサポートしないため、内部ドメイン パスワードなどのさまざまなフィールドや、RADIUS または LDAP サーバの属性で SSO を実行するオプションはありません。

プラグインに対して SSO サポートを設定するには、プラグインをインストールし、サーバへのリンクを表示するためのブックマーク エントリを追加します。また、csco_sso=1 パラメータを使用して SSO サポートを指定します。次に、SSO 用にイネーブルにするプラグインのブックマークの例を示します。

```
ssh://ssh-server/?cisco_sso=1
rdp://rdp-server/?Parameter1=value&Parameter2=value&csco_sso=1
```

マクロ置換による SSO の設定

ここでは、SSO のマクロ置換の使用について説明します。マクロ置換を使用して SSO を設定することで、ブックマークに特定の変数を挿入して動的な値に置換できます。

（注） スマート トンネル ブックマークでは、自動サインオンはサポートされていますが変数置換はサポートされていません。たとえば、スマート トンネル向けに設定された SharePoint ブックマークは、アプリケーションにログオンするために、クライアントレス SSL VPN にログオンするために使用するクレデンシャルと同じユーザ名とパスワードを使用します。（この SSO 機能は、クライアントレス VPN にのみ適用され、AnyConnect には適用されません。）変数置換および自動サインオンは同時に、または別々に使用できます。

一部の Web ページでの自動サインオンに、マクロ置換を含むブックマークを使用できるようになりました。以前の POST プラグイン アプローチは、管理者がサインオン マクロを含む POST ブックマークを指定し、POST 要求のポストの前にロードするキックオフ ページを受信できるようにするために作成されました。このPOSTプラグインアプローチでは、クッキーまたはその他のヘッダー項目の存在を必要とする要求は排除されました。現在は、管理者は事前ロード ページおよび URL を決定し、これによってポスト ログイン要求の送信場所が指定されます。事前ロードページによって、エンドポイントブラウザは、クレデンシャルを含むPOST要求を使用するのではなく、Web サーバまたは Web アプリケーションに送信される特定の情報を取得できます。

次に、ブックマーク内の置換およびフォームベースの HTTP POST 操作が可能な変数（またはマクロ）を示します。

- CSCO_WEBVPN_USERNAME：ユーザのログイン ID
- CSCO_WEBVPN_PASSWORD：ユーザのログイン パスワード
- CSCO_WEBVPN_INTERNAL_PASSWORD：ユーザの内部（または、ドメイン）パスワードこのキャッシュ済みクレデンシャルは、AAA サーバに対して認証されません。この値を入力すると、セキュリティ アプライアンスは、パスワードまたはプライマリ パスワードの値ではなく、この値を自動サインオンのパスワードとして使用します。

  （注） 上記の 3 つの変数は、GET ベースの HTTP（S）ブックマークでは使用できません。これらの値を使用できるのは、POST ベースの HTTP（S）および CIFS ブックマークだけです。

- CSCO_WEBVPN_CONNECTION_PROFILE：ユーザのログイン グループ ドロップダウン（接続プロファイル エイリアス）
- CSCO_WEBVPN_MACRO1：RADIUS-LDAP ベンダー固有属性（VSA）によって設定。LDAP から ldap-attribute-map コマンドをマッピングしている場合、このマクロの Cisco 属性である WebVPN-Macro-Substitution-Value1 を使用します。Active Directory での LDAP 属

性マッピングの例については、次の URL を参照してください。
http://www.cisco.com/en/US/docs/security/asa/asa83/configuration/guide/ref_extserver.html#wp1572118

RADIUS による CSCO_WEBVPN_MACRO1 のマクロ置換は、VSA#223 によって行われます。

表 1 : VSA#223

WebVPN-Macro-Value1	Y	223	文字列	シングル	無制限
WebVPN-Macro-Value2	Y	224	文字列	シングル	無制限

特定の DAP またはグループ ポリシーについて、https://CSCO_WEBVPN_MACRO1 や https://CSCO_WEBVPN_MACRO2 のようにすると、www.cisco.com/email などの値が、クライアントレス SSL VPN ポータルのブックマークに動的に読み込まれます。

- CSCO_WEBVPN_MACRO2：RADIUS-LDAP のベンダー固有属性（VSA）によって設定されます。LDAP から ldap-attribute-map コマンドをマッピングしている場合、このマクロの Cisco 属性である WebVPN-Macro-Substitution-Value2 を使用します。Active Directory での LDAP 属性マッピングの例については、次の URL を参照してください。
http://www.cisco.com/en/US/docs/security/asa/asa83/configuration/guide/ref_extserver.html#wp1572118

  RADIUS による CSCO_WEBVPN_MACRO2 のマクロ置換は、VSA#224 によって行われます。

クライアントレス SSL VPN が（ブックマークの形式または POST 形式の）エンドユーザの要求内にあるこれらの 6 つの文字列のいずれかを認識するたびに、文字列がユーザ指定の値に置き換えられ、この要求がリモート サーバに渡されます。

ユーザ名とパスワードのルックアップが ASA で失敗した場合は、空の文字列で置き換えられ、動作は自動サインインが不可の場合の状態に戻されます。

ユーザ名とパスワードの要件

ネットワークによっては、リモートセッション中にユーザが、コンピュータ、インターネット サービス プロバイダー、クライアントレス SSL VPN、メール サーバ、ファイル サーバ、企業アプリケーションの一部またはすべてにログインする必要が生じることがあります。ユーザはさまざまなコンテキストで認証を行うために、固有のユーザ名、パスワード、PIN などさまざまな情報が要求される場合があります。次の表に、クライアントレス SSL VPN ユーザが理解しておく必要のあるユーザ名とパスワードのタイプを示します。

ログイン ユーザ名/ パスワードのタイプ		入力するタイミング
コンピュータ	コンピュータへのアクセス	コンピュータの起動

ログイン ユーザ名/ パスワードのタイプ		入力するタイミング
Internet Service Provider：インターネット サービス プロバイダー	インターネットへのアクセス	インターネット サービス プロバイダーへの接続
クライアントレス SSL VPN	リモート ネットワークへのアクセス	クライアントレス SSL VPN の起動
ファイル サーバ	リモート ファイル サーバへのアクセス	クライアントレス SSL VPN ファイル ブラウジング機能を使用して、リモート ファイル サーバにアクセスするとき
企業アプリケーションへのログイン	ファイアウォールで保護された内部サーバへのアクセス	クライアントレス SSL VPN Web ブラウジング機能を使用して、保護されている内部 Web サイトにアクセスするとき
メール サーバ	クライアントレス SSL VPN 経由によるリモート メール サーバへのアクセス	電子メール メッセージの送受信

セキュリティ ヒントの通知

ユーザはいつでもツールバーの [Logout] アイコンをクリックして、クライアントレス SSL VPN セッションを閉じることができます（ブラウザ ウィンドウを閉じてもセッションは閉じません）。

クライアントレス SSL VPN は、企業ネットワーク上のリモート PC やワークステーションと ASA との間のデータ転送のセキュリティを保証するものです。クライアントレス SSL VPN を使用してもすべてのサイトとの通信がセキュアであるとは限らないことを、ユーザに通知してください。したがって、ユーザが HTTPS 以外の Web リソース（インターネット上や内部ネットワーク上にあるリソース）にアクセスする場合、企業の ASA から目的の Web サーバまでの通信は暗号化されていないため、プライベートではありません。

クライアントレスSSLVPNの機能を使用するためのリモート システムの設定

この項では、クライアントレス SSL VPN を使用するようにリモート システムを設定する方法について説明します。

- クライアントレス SSL VPN について （22 ページ）
- クライアントレス SSL VPN の前提条件 （22 ページ）

ユーザ アカウントを別々に設定でき、各ユーザは異なるクライアントレス SSL VPN の機能を使用できます。

クライアントレス SSL VPN について

次のようなサポートされている接続を使用して、インターネットに接続できます。

- 家庭の DSL、ケーブル、ダイヤルアップ。
- 公共のキオスク。
- ホテルのホットスポット。
- 空港の無線ノード。
- インターネット カフェ。

（注） クライアントレス SSL VPN がサポートしている Web ブラウザのリストについては、『サポート対象の VPN プラットフォーム、Cisco ASA 5500 シリーズ』を参照してください。

クライアントレス SSL VPN の前提条件

- ポート転送を介してアプリケーションにアクセスするために、ブラウザでクッキーをイネーブルにする必要があります。
- クライアントレス SSL VPN の URL が必要です。URL は、https:*//address* 形式の https アドレスでなければなりません。*address* は、SSL VPN がイネーブルになっている ASA（またはロード バランシング クラスタ）のインターフェイスの IP アドレスまたは DNS ホスト名です。たとえば、https://cisco.example.com などです。
- クライアントレス SSL VPN のユーザ名とパスワードが必要です。

（注） クライアントレス SSL VPN ではローカル印刷がサポートされていますが、VPN 経由による企業ネットワーク上のプリンタへの印刷はサポートされていません。

クライアントレス SSL VPN フローティング ツールバーの使用

フローティング ツールバーを使用すると、クライアントレス SSL VPN を簡単に使用できます。ツールバーを使用して、メインのブラウザ ウィンドウに影響を与えずに、URL の入力、ファイルの場所のブラウズ、設定済み Web 接続の選択ができます。

フローティング ツールバーは、現在のクライアントレス SSL VPN セッションを表します。[**Close**] ボタンをクリックすると、クライアントレス SSL VPN セッションの終了を求めるメッセージが ASA によって表示されます。

ヒント テキスト フィールドにテキストを貼り付けるには、Ctrl+V を使用します（クライアントレス SSL VPN セッション中は、表示されるツールバー上での右クリックはオフになっています）。

（注） ポップアップをブロックするようにブラウザが設定されている場合、フローティング ツールバーは表示できません。

Web のブラウズ

クライアントレス SSL VPN を使用しても、すべてのサイトとの通信がセキュアになるわけではありません。セキュリティ ヒントの通知 （21 ページ）を参照してください。

クライアントレス SSL VPN での Web ブラウジングのルックアンドフィールは、ユーザが使い慣れたものと異なる場合があります。次に例を示します。

- クライアントレス SSL VPN のタイトル バーが各 Web ページの上部に表示される。
- Web サイトへのアクセス方法：
  - クライアントレス SSL VPN ホーム ページ上の [Enter Web Address] フィールドに URL を入力する
  - クライアントレス SSL VPN ホーム ページ上にある設定済みの Web サイト リンクをクリックする
  - 上記 2 つのどちらかの方法でアクセスした Web ページ上のリンクをクリックする
  - 保護されている Web サイトのユーザ名とパスワードが必要です。

特定のアカウントの設定によっては、次のようになる場合もあります。

- 一部の Web サイトがブロックされている
- 使用可能な Web サイトが、クライアントレス SSL VPN ホーム ページ上にリンクとして表示されるものに限られる

また、特定のアカウントの設定によっては、次のようになる場合もあります。

- 一部の Web サイトがブロックされている
- 使用可能な Web サイトが、クライアントレス SSL VPN ホーム ページ上にリンクとして表示されるものに限られる

ネットワークのブラウズ（ファイル管理）

ユーザは、組織ネットワークを介してファイルを見つける方法に慣れていない場合があります。

（注） コピー処理の進行中は、**Copy File to Server** コマンドを中断したり、別の画面に移動したりしないでください。コピー処理を中断すると、不完全なファイルがサーバに保存される可能性があります。

重要なポイントは次のとおりです。

- 共有リモート アクセス用にファイル アクセス権を設定する必要があります。
- 保護されているファイル サーバのサーバ名とパスワードが必要です。
- フォルダとファイルが存在するドメイン、ワークグループ、およびサーバの名前が必要です。

（注） クライアントレス SSL VPN を介してアクセスできるのは、共有フォルダと共有ファイルに限られます。

Remote File Explorer の使用

ユーザは、Remote File Explorer を使用して、Web ブラウザから企業ネットワークをブラウズできます。ユーザが Cisco SSL VPN ポータル ページの [Remote File System] アイコンをクリックすると、ユーザのシステムでアプレットが起動し、ツリーおよびフォルダ ビューにリモート ファイル システムが表示されます。

（注） この機能を使用するには、ユーザのマシンに Oracle Java ランタイム環境（JRE）がインストールされ、Web ブラウザで Java がイネーブルになっている必要があります。リモート ファイルの起動には、JRE 1.6 以降が必要です。

ユーザはブラウザで次を実行できます。

- リモート ファイル システムのブラウズ。
- ファイルの名前の変更。
- リモートファイルシステム内、およびリモートとローカルのファイルシステム間でのファイルの移動またはコピー。
- ファイルのバルク アップロードおよびダウンロードの実行。

  ファイルをダウンロードするには、ブラウザでファイルをクリックして、[Operations] > [Download] を選択し、[Save] ダイアログで場所と名前を指定してファイルを保存します。

  ファイルをアップロードするには、宛先フォルダをクリックして、[Operations] > [Upload] を選択し、[Open] ダイアログでファイルの場所と名前を指定します。

  この機能には次の制限があります。

  - ユーザは、アクセスを許可されていないサブフォルダを表示できません。
  - ユーザがアクセスを許可されていないファイルは、ブラウザに表示されても移動またはコピーできません。
  - ネストされたフォルダの最大の深さは 32 です。
  - ツリー ビューでは、ドラッグ アンド ドロップのコピーがサポートされていません。
  - Remote File Explorer の複数のインスタンスの間でファイルを移動するときは、すべてのインスタンスが同じサーバを探索する必要があります（ルート共有）。
  - Remote File Explorer は、1 つのフォルダに最大 1500 のファイルおよびフォルダを表示できます。フォルダがこの制限を超えた場合、フォルダは表示されません。

ポート転送の使用

ポート フォワーディングを使用するには、ローカルにマッピングされたサーバの IP アドレスとポート番号を使用してクライアント アプリケーションを設定する必要があります。

- アプリケーションを使用した後、ユーザは [**Close**] アイコンをクリックして必ず [Application Access] ウィンドウを閉じる必要があります。このウィンドウを正しく閉じないと、Application Access またはアプリケーション自体がオフに切り替わる可能性があります。

始める前に

- Mac OS X の場合、この機能をサポートしているのは Safari ブラウザだけです。
- クライアント アプリケーションがインストールされている必要があります。
- ブラウザでクッキーをイネーブルにする必要があります。
- DNS 名を使用してサーバを指定する場合、ホスト ファイルの変更に必要になるため、PC に対する管理者アクセス権が必要です。
- Oracle Java Runtime Environment（JRE）をインストールしておく必要があります。

  JRE がインストールされていない場合は、ポップアップ ウィンドウが表示され、ユーザに対して使用可能なサイトが示されます。まれに、Java 例外エラーで、ポート フォワーディング アプレットが失敗することがあります。このような状況が発生した場合は、次の手順を実行します。

  1. ブラウザのキャッシュをクリアして、ブラウザを閉じます。
  2. Java アイコンがコンピュータのタスク バーに表示されていないことを確認します。
  3. Java のインスタンスをすべて閉じます。
  4. クライアントレス SSL VPN セッションを確立し、ポート フォワーディング Java アプレットを起動します。

- ブラウザで javascript をイネーブルにする必要があります。デフォルトでは有効に設定されています。
- 必要に応じて、クライアント アプリケーションを設定する必要があります。

（注） Microsoft Outlook クライアントの場合、この設定手順は不要です。Windows 以外のすべてのクライアント アプリケーションでは、設定が必要です。Windows アプリケーションの設定が必要かどうかを確認するには、[Remote Server] フィールドの値をチェックします。[Remote Server] フィールドにサーバ ホスト名が含まれている場合、クライアント アプリケーションの設定は不要です。[Remote Server] フィールドに IP アドレスが含まれている場合、クライアント アプリケーションを設定する必要があります。

手順

ステップ 1 クライアントレス SSL VPN セッションを開始して、[Home] ページの [Application Access] リンクをクリックします。[Application Access] ウィンドウが表示されます。

ステップ 2 [Name] カラムで、使用するサーバ名を確認し、このサーバに対応するクライアント IP アドレスとポート番号を [Local] カラムで確認します。

ステップ 3 この IP アドレスとポート番号を使用して、クライアント アプリケーションを設定します。設定手順は、クライアント アプリケーションによって異なります。

（注） クライアントレス SSL VPN セッション上で実行しているアプリケーションで URL（電子メール メッセージ内のものなど）をクリックしても、サイトがそのセッションで開くわけではありません。サイトをセッション上で開くには、その URL を [Enter Clientless SSL VPN (URL) Address] フィールドに貼り付けます。

ポート転送を介した電子メールの使用

電子メールを使用するには、クライアントレス SSL VPN のホームページから Application Access を起動します。これにより、メール クライアントが使用できるようになります。

（注） IMAP クライアントの使用中にメール サーバとの接続が中断したり、新しく接続を確立できない場合は、IMAP アプリケーションを終了してクライアントレス SSL VPN を再起動します。

アプリケーション アクセスおよびその他のメール クライアントの要件を満たしている必要があります。

Microsoft Outlook Express バージョン 5.5 および 6.0 はテスト済みです。

クライアントレス SSL VPN は、Lotus Notes および Eudora などの、ポート転送を介したその他の SMTPS、POP3S、または IMAP4S 電子メール プログラムをサポートしますが、動作確認は行っていません。

Web アクセスを介した電子メールの使用

次の電子メール アプリケーションがサポートされています。

- Microsoft Outlook Web App to Exchange Server 2010

  OWA には、Internet Explorer 7 以降、または Firefox 3.01 以降が必要です。

- Microsoft Outlook Web Access to Exchange Server 2007、2003、および 2000

  最適な結果を得るために、Internet Explorer 8.x 以降、または Firefox 8.x で OWA を使用してください。

- Louts iNotes

（注） Web ベースの電子メール製品がインストールされており、その他の Web ベースの電子メール アプリケーションも動作する必要がありますが、検証されていません。

電子メール プロキシを介した電子メールの使用

次のレガシー電子メール アプリケーションがサポートされています。

- Microsoft Outlook 2000 および 2002
- Microsoft Outlook Express 5.5 および 6.0

メール アプリケーションの使用方法と例については、「クライアントレス SSL VPN を介した電子メールの使用」を参照してください。

はじめる前に

SSL 対応メール アプリケーションがインストールされている必要があります。

ASA SSL バージョンを TLSv1 Only に設定しないでください。Outlook および Outlook Express では TLS はサポートされません。

メール アプリケーションが正しく設定されている必要があります。

その他の SSL 対応クライアントも動作しますが、動作確認は行っていません。

スマート トンネルの使用

スマート トンネルの使用に管理権限は必要ありません。

（注） ポート フォワーダの場合と異なり、Java は自動的にダウンロードされません。

- スマート トンネルを使用する場合、Windows では ActiveX または JRE、Mac OS X では Java Web Start が必要です。
- ブラウザでクッキーをイネーブルにする必要があります。
- ブラウザで javascript をイネーブルにする必要があります。
- Mac OS X では、フロントサイド プロキシはサポートされていません。
- サポートされているオペレーティング システムとブラウザだけを使用してください。
- TCP ソケットベースのアプリケーションだけがサポートされています。