目次

概要

このドキュメントでは、Cisco Adaptive Security Appliance（ASA）に接続する Cisco Anyconnect セキュア モビリティ クライアント用のユーザ単位属性をサポートするように、カスタム スキーマと OpenLDAP を設定する方法について説明します。 ASA 設定は、すべてのユーザ属性が OpenLDAP サーバから取得されるため、非常に基本的です。 証明書と組み合わせて使用した場合の LDAP 認証と認可の違いも、このドキュメントで説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Linux 設定に関する基本的な知識
- ASA CLI 設定に関する基本的な知識

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くものです。

- Cisco ASA バージョン 8.4 以降
- OpenLDAP バージョン 2.4.30

設定

基本 OpenLDAP 設定

ステップ 1：サーバを設定します。

この例では test-cisco.com ldap ツリーを使用します。

ldap.conf ファイルは、ローカル LDAP クライアントが使用できるシステム レベルのデフォルトを設定するために使用されます。

注 システム レベルのデフォルトを設定する必要はありませんが、ローカル LDAP クライアントを実行するときのサーバのテストとトラブルシューティングに役立ちます。

/etc/openldap/ldap.conf:

slapd.conf ファイルは、OpenLDAP のサーバ設定に使用されます。 デフォルトのスキーマ ファイルには、一般的な LDAP 定義が含まれています。 たとえば、オブジェクト クラス名 personis が core.schema ファイルに定義されています。 この設定では、この共通スキーマを使用し、Cisco 固有属性用の独自のスキーマを定義します。

/etc/openldap/slapd.conf:

```
include         /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/openldap.schema
include /etc/openldap/schema/nis.schema

# Defines backend database type and redirects all # queries with specified suffix to that
database
database hdb
suffix "dc=test-cisco,dc=com"
checkpoint 32 30

# Rootdn will be used to perform all administrative tasks.
rootdn          "cn=Manager,dc=test-cisco,dc=com"

# Cleartext passwords, especially for the rootdn, should be avoid.
rootpw          secret

directory /var/lib/openldap-data
index objectClass eq
```

ステップ 2：LDAP 設定を確認します。

基本的な OpenLDAP が動作していることを確認するには、この設定を実行します。

```
include         /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/openldap.schema
```

```
include /etc/openldap/schema/nis.schema

# Defines backend database type and redirects all # queries with specified suffix to that
database
database hdb
suffix "dc=test-cisco,dc=com"
checkpoint 32 30

# Rootdn will be used to perform all administrative tasks.
rootdn          "cn=Manager,dc=test-cisco,dc=com"

# Cleartext passwords, especially for the rootdn, should be avoid.
rootpw          secret

directory /var/lib/openldap-data
index objectClass eq
```

ステップ 3：データベースにレコードを追加します。

すべてのテストと設定を適切に実施したら、データベースにレコードを追加します。 ユーザとグループの基本的なコンテナを追加するには、この設定を実行します。

```
include         /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/openldap.schema
include /etc/openldap/schema/nis.schema

# Defines backend database type and redirects all # queries with specified suffix to that
database
database hdb
suffix "dc=test-cisco,dc=com"
checkpoint 32 30

# Rootdn will be used to perform all administrative tasks.
rootdn          "cn=Manager,dc=test-cisco,dc=com"

# Cleartext passwords, especially for the rootdn, should be avoid.
rootpw          secret

directory /var/lib/openldap-data
index objectClass eq
```

カスタム OpenLDAP スキーマ

これで基本設定が動作するため、カスタム スキーマを追加できます。 この設定例では、*CiscoPerson* という新しいタイプのオブジェクト クラスが作成され、次の属性がこのオブジェクト クラスで作成および使用されます。

- CiscoBanner
- CiscoACLin
- CiscoDomain
- CiscoDNS
- CiscoIPAddress
- CiscoIPNetmask
- CiscoSplitACL
- CiscoSplitTunnelPolicy
- CiscoGroupPolicy

ステップ 1：cisco.schema で新しいスキーマを作成します。

```
include         /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/openldap.schema
include /etc/openldap/schema/nis.schema

# Defines backend database type and redirects all # queries with specified suffix to that
database
database hdb
suffix "dc=test-cisco,dc=com"
checkpoint 32 30

# Rootdn will be used to perform all administrative tasks.
rootdn          "cn=Manager,dc=test-cisco,dc=com"

# Cleartext passwords, especially for the rootdn, should be avoid.
rootpw          secret

directory /var/lib/openldap-data
index objectClass eq
```

重要事項

- 会社の民間企業 OID を使用します。 どの OID でも使用できますが、ベスト プラクティスは、IANA によって割り当てられた OID を使用することです。 この例に設定されている値は 1.3.6.1.4.1.9 で開始されます（シスコによって予約、http://www.iana.org/assignments/enterprise-numbers ）。
- OID（500.1.1 ～ 500.1.9）の次の部分は Cisco OID のメイン ツリー（「1.3.6.1.4.1.9」）で直接干渉しないために使用されています。
- このデータベースでは、schema/core.ldif で定義された *Person* オブジェクト クラスを使用します。 このオブジェクトは、TOP タイプであり、レコードでは該当する属性を 1 つだけ含むことができます（これが CiscoPersonobject クラスが Auxiliary タイプである理由です）。
- *CiscoPerson* というオブジェクト クラスは SN または CN を含む必要があり、事前に定義した任意のカスタム Cisco 属性を含むことができます。 他のスキーマに定義されている他の任意の属性（*userPassword*、*telephoneNumber* など）も含めることができることに注意してください。
- 各オブジェクトは異なる OID 番号を持つ必要があることに注意してください。
- カスタム属性では、大文字と小文字が区別されず、UTF-8 エンコーディングによる *文字列型* で、最大 128 文字です（SYNTAX で定義）。

ステップ 2：sldap.conf にスキーマを含めます。

```
pluton openldap # cat slapd.conf | grep include
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/inetorgperson.schema
include         /etc/openldap/schema/openldap.schema
include         /etc/openldap/schema/nis.schema
include         /etc/openldap/schema/cisco.schema
```

ステップ 3：サービスを再開します。

```
pluton openldap # cat slapd.conf | grep include
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/inetorgperson.schema
```

```
include         /etc/openldap/schema/openldap.schema
include         /etc/openldap/schema/nis.schema
include         /etc/openldap/schema/cisco.schema
```

ステップ 4：すべてのカスタム属性で新しいユーザを追加します。

この例では、ユーザは複数の objectClass オブジェクトに属し、すべてのオブジェクトから属性を継承します。 このプロセスでは、既存のデータベース レコードを変更せずに追加スキーマまたは属性を追加することが簡単です。

```
pluton # cat users.ldiff
# User account
dn: uid=cisco,ou=people,dc=test-cisco,dc=com
cn: John Smith
givenName: John
sn: cisco
uid: cisco
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/cisco
mail: jsmith@dev.local
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: CiscoPerson
loginShell: /bin/bash
userPassword: {CRYPT}*
CiscoBanner: This is banner 1
CiscoIPAddress: 10.1.1.1
CiscoIPNetmask: 255.255.255.128
CiscoDomain: domain1.com
CiscoDNS: 10.6.6.6
CiscoACLin: ip:inacl#1=permit ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
CiscoSplitACL: ACL1
CiscoSplitTunnelPolicy: 1
CiscoGroupPolicy: POLICY1

pluton # ldapadd -h 192.168.10.1 -D "CN=Manager,DC=test-cisco,DC=com"
-w secret -x -f users.ldiff
adding new entry "uid=cisco,ou=people,dc=test-cisco,dc=com"
```

ステップ 5：ユーザのパスワードを設定します。

```
pluton # cat users.ldiff
# User account
dn: uid=cisco,ou=people,dc=test-cisco,dc=com
cn: John Smith
givenName: John
sn: cisco
uid: cisco
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/cisco
mail: jsmith@dev.local
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: CiscoPerson
```

```
loginShell: /bin/bash
userPassword: {CRYPT}*
CiscoBanner: This is banner 1
CiscoIPAddress: 10.1.1.1
CiscoIPNetmask: 255.255.255.128
CiscoDomain: domain1.com
CiscoDNS: 10.6.6.6
CiscoACLin: ip:inacl#1=permit ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
CiscoSplitACL: ACL1
CiscoSplitTunnelPolicy: 1
CiscoGroupPolicy: POLICY1

pluton # ldapadd -h 192.168.10.1 -D "CN=Manager,DC=test-cisco,DC=com"
-w secret -x -f users.ldiff
adding new entry "uid=cisco,ou=people,dc=test-cisco,dc=com"
```

ステップ 6：設定を検証します。

```
pluton # cat users.ldiff
# User account
dn: uid=cisco,ou=people,dc=test-cisco,dc=com
cn: John Smith
givenName: John
sn: cisco
uid: cisco
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/cisco
mail: jsmith@dev.local
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: CiscoPerson
loginShell: /bin/bash
userPassword: {CRYPT}*
CiscoBanner: This is banner 1
CiscoIPAddress: 10.1.1.1
CiscoIPNetmask: 255.255.255.128
CiscoDomain: domain1.com
CiscoDNS: 10.6.6.6
CiscoACLin: ip:inacl#1=permit ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
CiscoSplitACL: ACL1
CiscoSplitTunnelPolicy: 1
CiscoGroupPolicy: POLICY1

pluton # ldapadd -h 192.168.10.1 -D "CN=Manager,DC=test-cisco,DC=com"
-w secret -x -f users.ldiff
adding new entry "uid=cisco,ou=people,dc=test-cisco,dc=com"
```

ASA の設定

ステップ 1：インターフェイスおよび証明書を設定します。

```
pluton # cat users.ldiff
# User account
dn: uid=cisco,ou=people,dc=test-cisco,dc=com
cn: John Smith
givenName: John
sn: cisco
uid: cisco
```

```
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/cisco
mail: jsmith@dev.local
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: CiscoPerson
loginShell: /bin/bash
userPassword: {CRYPT}*
CiscoBanner: This is banner 1
CiscoIPAddress: 10.1.1.1
CiscoIPNetmask: 255.255.255.128
CiscoDomain: domain1.com
CiscoDNS: 10.6.6.6
CiscoACLin: ip:inacl#1=permit ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
CiscoSplitACL: ACL1
CiscoSplitTunnelPolicy: 1
CiscoGroupPolicy: POLICY1

pluton # ldapadd -h 192.168.10.1 -D "CN=Manager,DC=test-cisco,DC=com"
-w secret -x -f users.ldiff
adding new entry "uid=cisco,ou=people,dc=test-cisco,dc=com"
```

ステップ 2：自己署名証明書を生成します。

```
pluton # cat users.ldiff
# User account
dn: uid=cisco,ou=people,dc=test-cisco,dc=com
cn: John Smith
givenName: John
sn: cisco
uid: cisco
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/cisco
mail: jsmith@dev.local
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: CiscoPerson
loginShell: /bin/bash
userPassword: {CRYPT}*
CiscoBanner: This is banner 1
CiscoIPAddress: 10.1.1.1
CiscoIPNetmask: 255.255.255.128
CiscoDomain: domain1.com
CiscoDNS: 10.6.6.6
CiscoACLin: ip:inacl#1=permit ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
CiscoSplitACL: ACL1
CiscoSplitTunnelPolicy: 1
CiscoGroupPolicy: POLICY1

pluton # ldapadd -h 192.168.10.1 -D "CN=Manager,DC=test-cisco,DC=com"
-w secret -x -f users.ldiff
adding new entry "uid=cisco,ou=people,dc=test-cisco,dc=com"
```

ステップ 3：外部インターフェイスで WebVPN をイネーブルにします。

```
pluton # cat users.ldiff
# User account
dn: uid=cisco,ou=people,dc=test-cisco,dc=com
cn: John Smith
givenName: John
sn: cisco
uid: cisco
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/cisco
mail: jsmith@dev.local
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: CiscoPerson
loginShell: /bin/bash
userPassword: {CRYPT}*
CiscoBanner: This is banner 1
CiscoIPAddress: 10.1.1.1
CiscoIPNetmask: 255.255.255.128
CiscoDomain: domain1.com
CiscoDNS: 10.6.6.6
CiscoACLin: ip:inacl#1=permit ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
CiscoSplitACL: ACL1
CiscoSplitTunnelPolicy: 1
CiscoGroupPolicy: POLICY1

pluton # ldapadd -h 192.168.10.1 -D "CN=Manager,DC=test-cisco,DC=com"
-w secret -x -f users.ldiff
adding new entry "uid=cisco,ou=people,dc=test-cisco,dc=com"
```

ステップ 4：ACL 設定を分割します。

ACL 名は、OpenLDAP によって返されます。

```
pluton # cat users.ldiff
# User account
dn: uid=cisco,ou=people,dc=test-cisco,dc=com
cn: John Smith
givenName: John
sn: cisco
uid: cisco
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/cisco
mail: jsmith@dev.local
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: CiscoPerson
loginShell: /bin/bash
userPassword: {CRYPT}*
CiscoBanner: This is banner 1
CiscoIPAddress: 10.1.1.1
CiscoIPNetmask: 255.255.255.128
CiscoDomain: domain1.com
CiscoDNS: 10.6.6.6
CiscoACLin: ip:inacl#1=permit ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
```

```
CiscoSplitACL: ACL1
CiscoSplitTunnelPolicy: 1
CiscoGroupPolicy: POLICY1

pluton # ldapadd -h 192.168.10.1 -D "CN=Manager,DC=test-cisco,DC=com"
-w secret -x -f users.ldiff
adding new entry "uid=cisco,ou=people,dc=test-cisco,dc=com"
```

ステップ 5：デフォルトのグループ ポリシー（DfltAccessPolicy）を使用するトンネル グループ名を作成します。

特定の LDAP 属性（*CiscoGroupPolicy*）を持つユーザは別のポリシーにマッピングされます。POLICY1

```
pluton # cat users.ldiff
# User account
dn: uid=cisco,ou=people,dc=test-cisco,dc=com
cn: John Smith
givenName: John
sn: cisco
uid: cisco
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/cisco
mail: jsmith@dev.local
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: CiscoPerson
loginShell: /bin/bash
userPassword: {CRYPT}*
CiscoBanner: This is banner 1
CiscoIPAddress: 10.1.1.1
CiscoIPNetmask: 255.255.255.128
CiscoDomain: domain1.com
CiscoDNS: 10.6.6.6
CiscoACLin: ip:inacl#1=permit ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
CiscoSplitACL: ACL1
CiscoSplitTunnelPolicy: 1
CiscoGroupPolicy: POLICY1

pluton # ldapadd -h 192.168.10.1 -D "CN=Manager,DC=test-cisco,DC=com"
-w secret -x -f users.ldiff
adding new entry "uid=cisco,ou=people,dc=test-cisco,dc=com"
```

ASA aaa-server 設定では、OpenLDAP によって返された属性から Anyconnect ユーザ用に ASA で解釈できる属性へのマッピングのために LDAP 属性マップを使用します。

```
pluton # cat users.ldiff
# User account
dn: uid=cisco,ou=people,dc=test-cisco,dc=com
cn: John Smith
givenName: John
sn: cisco
uid: cisco
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/cisco
mail: jsmith@dev.local
objectClass: top
objectClass: posixAccount
```

```
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: CiscoPerson
loginShell: /bin/bash
userPassword: {CRYPT}*
CiscoBanner: This is banner 1
CiscoIPAddress: 10.1.1.1
CiscoIPNetmask: 255.255.255.128
CiscoDomain: domain1.com
CiscoDNS: 10.6.6.6
CiscoACLin: ip:inacl#1=permit ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
CiscoSplitACL: ACL1
CiscoSplitTunnelPolicy: 1
CiscoGroupPolicy: POLICY1

pluton # ldapadd -h 192.168.10.1 -D "CN=Manager,DC=test-cisco,DC=com"
-w secret -x -f users.ldiff
adding new entry "uid=cisco,ou=people,dc=test-cisco,dc=com"
```

ステップ 6：LDAP サーバで指定したトンネル グループを認証できるようにします。

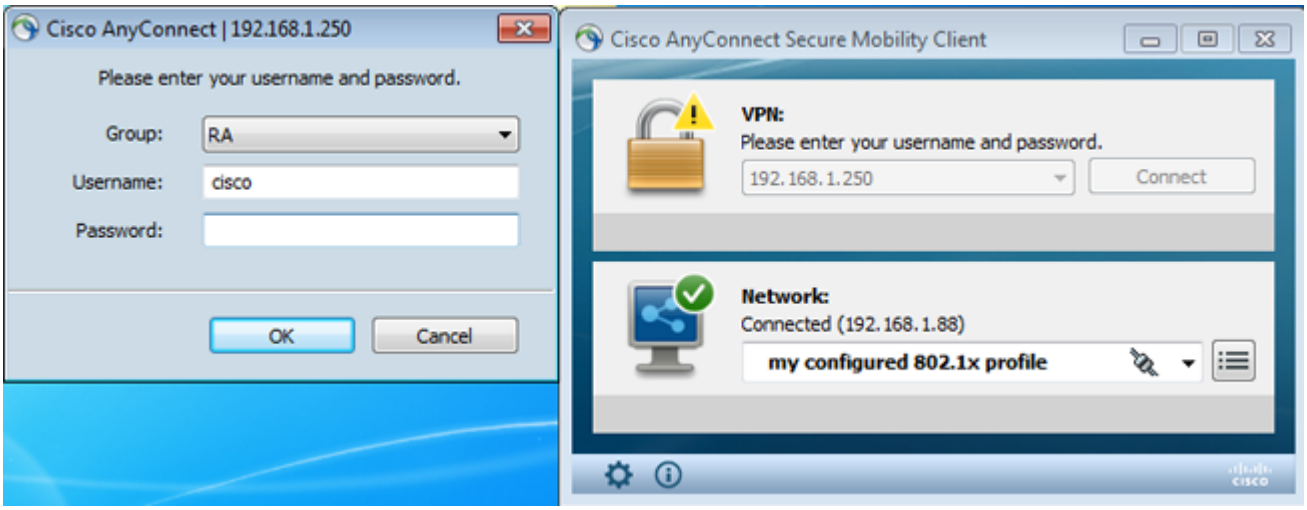
```
pluton # cat users.ldiff
# User account
dn: uid=cisco,ou=people,dc=test-cisco,dc=com
cn: John Smith
givenName: John
sn: cisco
uid: cisco
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/cisco
mail: jsmith@dev.local
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: CiscoPerson
loginShell: /bin/bash
userPassword: {CRYPT}*
CiscoBanner: This is banner 1
CiscoIPAddress: 10.1.1.1
CiscoIPNetmask: 255.255.255.128
CiscoDomain: domain1.com
CiscoDNS: 10.6.6.6
CiscoACLin: ip:inacl#1=permit ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
CiscoSplitACL: ACL1
CiscoSplitTunnelPolicy: 1
CiscoGroupPolicy: POLICY1

pluton # ldapadd -h 192.168.10.1 -D "CN=Manager,DC=test-cisco,DC=com"
-w secret -x -f users.ldiff
adding new entry "uid=cisco,ou=people,dc=test-cisco,dc=com"
```

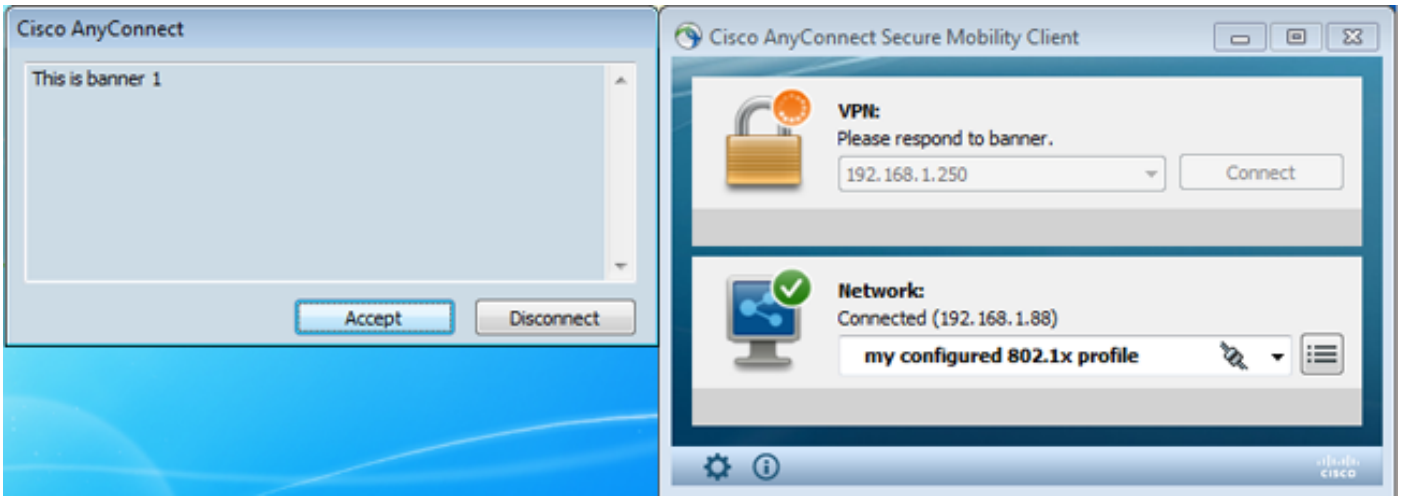
確認

VPN アクセスのテスト

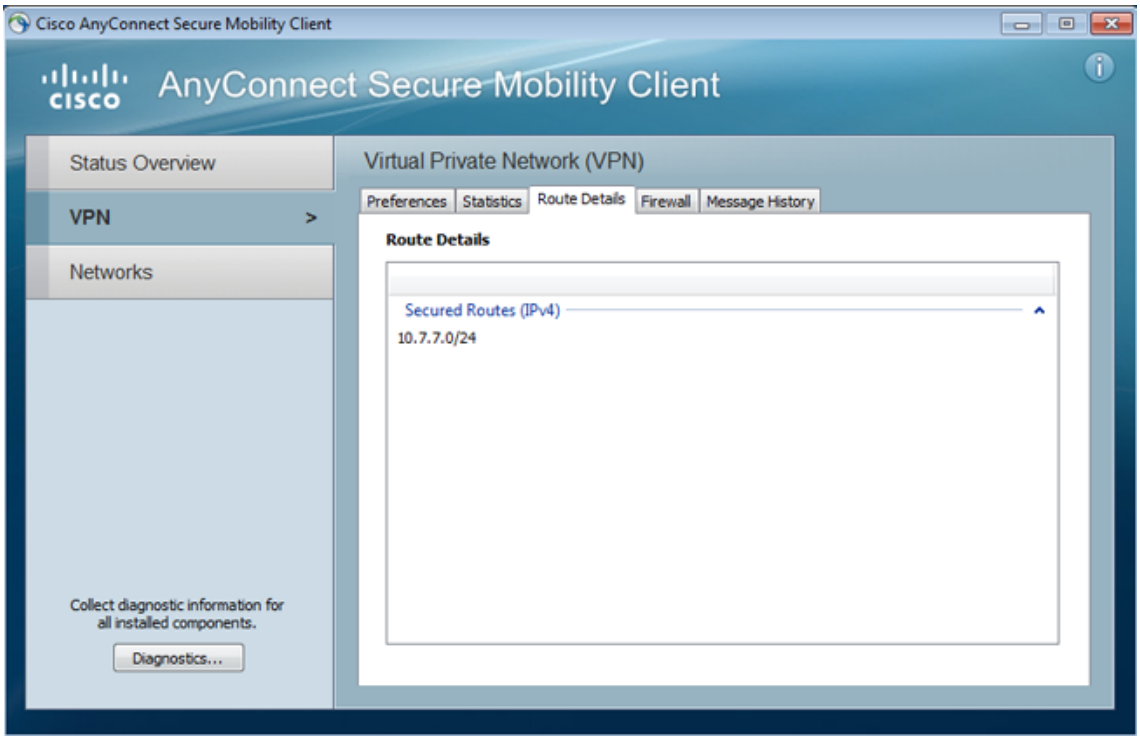
Anyconnect は 192.168.1.250 に接続するように設定されています。 ログインはユーザ名 *cisco*、パスワード *pass1* です。



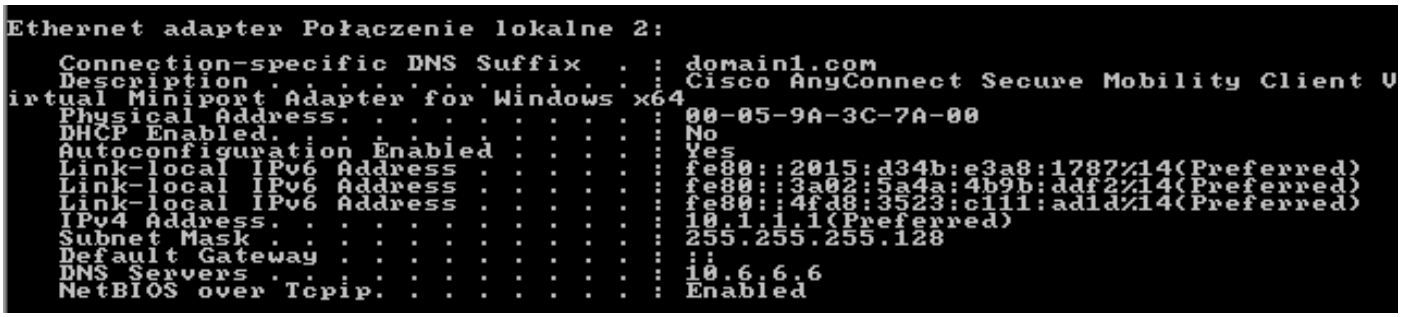
認証後は正しいバナーが使用されます。



正しい分割 ACL が送信されます（ASA で定義されている ACL1）。



Anyconnect のインターフェイスは、IP 10.1.1.1、ネットマスク 255.255.255.128 で設定されます。 ドメインは domain1.com であり、DNS サーバは 10.6.6.6 です。



ASA で、ユーザ *cisco* は IP 10.1.1.1 を受信し、グループ ポリシー *POLICY1* に割り当てられています。

```
ASA# show vpn-sessiondb detail anyconnect

Session Type: AnyConnect Detailed

Username     : cisco                  Index        : 29
Assigned IP  : 10.1.1.1               Public IP    : 192.168.1.88
Protocol     : AnyConnect-Parent SSL-Tunnel
License      : AnyConnect Premium
Encryption   : RC4                    Hashing      : none SHA1
Bytes Tx     : 10212                  Bytes Rx     : 856
Pkts Tx      : 8                      Pkts Rx      : 2
Pkts Tx Drop : 0                      Pkts Rx Drop : 0
Group Policy : POLICY1                Tunnel Group : RA
Login Time   : 10:18:25 UTC Thu Apr 4 2013
Duration     : 0h:00m:17s
Inactivity   : 0h:00m:00s
NAC Result   : Unknown
VLAN Mapping : N/A                    VLAN         : none

AnyConnect-Parent Tunnels: 1
SSL-Tunnel Tunnels: 1

AnyConnect-Parent:
 Tunnel ID    : 29.1
 Public IP    : 192.168.1.88
 Encryption   : none                   TCP Src Port : 49262
 TCP Dst Port : 443                    Auth Mode    : userPassword
 Idle Time Out: 30 Minutes             Idle TO Left : 29 Minutes
 Client Type  : AnyConnect
 Client Ver   : 3.1.01065
 Bytes Tx     : 5106                   Bytes Rx     : 788
 Pkts Tx      : 4                      Pkts Rx      : 1
 Pkts Tx Drop : 0                      Pkts Rx Drop : 0

SSL-Tunnel:
 Tunnel ID    : 29.2
 Assigned IP  : 10.1.1.1               Public IP    : 192.168.1.88
 Encryption   : RC4                    Hashing      : SHA1
 Encapsulation: TLSv1.0                TCP Src Port : 49265
 TCP Dst Port : 443                    Auth Mode    : userPassword
 Idle Time Out: 30 Minutes             Idle TO Left : 29 Minutes
 Client Type  : SSL VPN Client
 Client Ver   : Cisco AnyConnect VPN Agent for Windows 3.1.01065
 Bytes Tx     : 5106                   Bytes Rx     : 68
 Pkts Tx      : 4                      Pkts Rx      : 1
 Pkts Tx Drop : 0                      Pkts Rx Drop : 0
```

```
 Filter Name  : AAA-user-cisco-E0CF3C05

NAC:
 Reval Int (T): 0 Seconds              Reval Left(T): 0 Seconds
 SQ Int (T)   : 0 Seconds              EoU Age(T)   : 17 Seconds
 Hold Left (T): 0 Seconds              Posture Token:
```

このユーザのダイナミック アクセス リストもインストールされています。

```
ASA# show access-list AAA-user-cisco-E0CF3C05
access-list AAA-user-cisco-E0CF3C05; 1 elements; name hash: 0xf9b6b75c (dynamic)
access-list AAA-user-cisco-E0CF3C05 line 1 extended permit
ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
(hitcnt=0) 0xf8010475
```

デバッグ

デバッグを有効にしたら、WebVPN セッションの各ステップを追跡できます。

この例では、属性の取得とともに LDAP 認証を示しています。

```
ASA# show debug
debug ldap  enabled at level 255
debug webvpn anyconnect enabled at level 254
ASA#
[63] Session Start
[63] New request Session, context 0xbbe10120, reqType = Authentication
[63] Fiber started
[63] Creating LDAP context with uri=ldap://192.168.11.10:389
[63] Connect to LDAP server: ldap://192.168.11.10:389, status = Successful
[63] supportedLDAPVersion: value = 3
[63] Binding as Manager
[63] Performing Simple authentication for Manager to 192.168.11.10
[63] LDAP Search:
       Base DN = [DC=test-cisco,DC=com]
       Filter  = [uid=cisco]
       Scope   = [SUBTREE]
[63] User DN = [uid=cisco,ou=People,dc=test-cisco,dc=com]
[63] Server type for 192.168.11.10 unknown - no password policy
[63] Binding as cisco
[63] Performing Simple authentication for cisco to 192.168.11.10
[63] Processing LDAP response for user cisco
[63] Authentication successful for cisco to 192.168.11.10
[63] Retrieved User Attributes:
[63]    cn: value = John Smith
[63]    givenName: value = John
[63]    sn: value = cisco
[63]    uid: value = cisco
[63]    uidNumber: value = 10000
[63]    gidNumber: value = 10000
[63]    homeDirectory: value = /home/cisco
[63]    mail: value = jsmith@dev.local
[63]    objectClass: value = top
[63]    objectClass: value = posixAccount
[63]    objectClass: value = shadowAccount
[63]    objectClass: value = inetOrgPerson
[63]    objectClass: value = organizationalPerson
[63]    objectClass: value = person
[63]    objectClass: value = CiscoPerson
[63]    loginShell: value = /bin/bash
```

重要： カスタム LDAP 属性は、LDAP 属性マップの定義に従って、ASA 属性にマッピングされます。

```
[63]    CiscoBanner: value = This is banner 1
```

```
[63]            mapped to Banner1: value = This is banner 1
[63]    CiscoIPAddress: value = 10.1.1.1
[63]            mapped to IETF-Radius-Framed-IP-Address: value = 10.1.1.1
[63]    CiscoIPNetmask: value = 255.255.255.128
[63]            mapped to IETF-Radius-Framed-IP-Netmask: value = 255.255.255.128
[63]    CiscoDomain: value = domain1.com
[63]            mapped to IPSec-Default-Domain: value = domain1.com
[63]    CiscoDNS: value = 10.6.6.6
[63]            mapped to Primary-DNS: value = 10.6.6.6
[63]    CiscoACLin: value = ip:inacl#1=permit
ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
[63]            mapped to Cisco-AV-Pair: value = ip:inacl#1=permit
ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0
[63]    CiscoSplitACL: value = ACL1
[63]            mapped to IPSec-Split-Tunnel-List: value = ACL1
[63]    CiscoSplitTunnelPolicy: value = 1
[63]            mapped to IPSec-Split-Tunneling-Policy: value = 1
[63]    CiscoGroupPolicy: value = POLICY1
[63]            mapped to IETF-Radius-Class: value = POLICY1
[63]            mapped to LDAP-Class: value = POLICY1
[63]    userPassword: value = {SSHA}5s81Fmi/9aG/WfPSy3lGmw1ORI4lywWC
[63] ATTR_CISCO_AV_PAIR attribute contains 68 bytes
[63] Fiber exit Tx=315 bytes Rx=907 bytes, status=1
[63] Session End
```

LDAP セッションが終了します。 ここで、ASA はこれらの属性を処理し、適用します。

ダイナミック ACL が作成されます（Cisco-AV-Pair 内のエントリである ACE に基づく）。

```
webvpn_svc_parse_acl: processing ACL: name: 'AAA-user-cisco-E0CF3C05',
list: YES, id -1
webvpn_svc_parse_acl: before add: acl_id: -1, acl_name: AAA-user-cisco-E0CF3C05
webvpn_svc_parse_acl: after add: acl_id: 5, acl_name: AAA-user-cisco-E0CF3C05,
refcnt: 1
```

WebVPN セッションが進みます。

```
webvpn_rx_data_tunnel_connect
CSTP state = HEADER_PROCESSING
http_parse_cstp_method()
...input: 'CONNECT /CSCOSSLC/tunnel HTTP/1.1'
webvpn_cstp_parse_request_field()
...input: 'Host: 192.168.1.250'
Processing CSTP header line: 'Host: 192.168.1.250'
webvpn_cstp_parse_request_field()
...input: 'User-Agent: Cisco AnyConnect VPN Agent for Windows 3.1.01065'
Processing CSTP header line: 'User-Agent: Cisco AnyConnect VPN Agent
for Windows 3.1.01065'
Setting user-agent to: 'Cisco AnyConnect VPN Agent for Windows 3.1.01065'
webvpn_cstp_parse_request_field()
...input: 'Cookie: webvpn=1476503744@122880@
1365070898@908F356D1C1F4CDF1138088854AF0E480FDCB1BD'
Processing CSTP header line: 'Cookie: webvpn=1476503744@122880@
1365070898@908F356D1C1F4CDF1138088854AF0E480FDCB1BD'
Found WebVPN cookie: 'webvpn=1476503744@122880@
1365070898@908F356D1C1F4CDF1138088854AF0E480FDCB1BD'
WebVPN Cookie: 'webvpn=1476503744@122880@1365070898@
908F356D1C1F4CDF1138088854AF0E480FDCB1BD'
IPADDR: '1476503744', INDEX: '122880', LOGIN: '1365070898'
webvpn_cstp_parse_request_field()
...input: 'X-CSTP-Version: 1'
Processing CSTP header line: 'X-CSTP-Version: 1'
Setting version to '1'
webvpn_cstp_parse_request_field()
```

```
...input: 'X-CSTP-Hostname: admin-Komputer'
Processing CSTP header line: 'X-CSTP-Hostname: admin-Komputer'
Setting hostname to: 'admin-Komputer'
webvpn_cstp_parse_request_field()
...input: 'X-CSTP-MTU: 1367'
Processing CSTP header line: 'X-CSTP-MTU: 1367'
webvpn_cstp_parse_request_field()
...input: 'X-CSTP-Address-Type: IPv6,IPv4'
Processing CSTP header line: 'X-CSTP-Address-Type: IPv6,IPv4'
webvpn_cstp_parse_request_field()
...input: 'X-CSTP-Local-Address-IP4: 192.168.1.88'
webvpn_cstp_parse_request_field()
...input: 'X-CSTP-Base-MTU: 1468'
webvpn_cstp_parse_request_field()
...input: 'X-CSTP-Remote-Address-IP4: 192.168.1.250'
webvpn_cstp_parse_request_field()
...input: 'X-CSTP-Full-IPv6-Capability: true'
webvpn_cstp_parse_request_field()
...input: 'X-DTLS-Master-Secret: F5ADDD0151261404504FC3B165C3B68A90E51
A1C8EB7EA9B2FE70F1EB8E10929FFD79650B07E218EC8774678CDE1FB5E'
Processing CSTP header line: 'X-DTLS-Master-Secret: F5ADDD015126140450
4FC3B165C3B68A90E51A1C8EB7EA9B2FE70F1EB8E10929FFD79650B07E2
18EC8774678CDE1FB5E'
webvpn_cstp_parse_request_field()
...input: 'X-DTLS-CipherSuite: AES256-SHA:AES128-SHA:DES-CBC3-SHA:DES-CBC-SHA'
Processing CSTP header line: 'X-DTLS-CipherSuite: AES256-SHA:AES128-SHA:
DES-CBC3-SHA:DES-CBC-SHA'
webvpn_cstp_parse_request_field()
...input: 'X-DTLS-Accept-Encoding: lzs'
Processing CSTL header line: 'X-DTLS-Accept-Encoding: lzs'
webvpn_cstp_parse_request_field()
...input: 'X-DTLS-Header-Pad-Length: 0'
webvpn_cstp_parse_request_field()
...input: 'X-CSTP-Accept-Encoding: lzs,deflate'
Processing CSTP header line: 'X-CSTP-Accept-Encoding: lzs,deflate'
webvpn_cstp_parse_request_field()
...input: 'X-CSTP-Protocol: Copyright (c) 2004 Cisco Systems, Inc.'
Processing CSTP header line: 'X-CSTP-Protocol:
Copyright (c) 2004 Cisco Systems, Inc.'
```

次に、アドレス割り当てが発生します。 ASA には IP プールは定義されていません。 LDAP が *CiscoIPAddress* 属性を返さない場合（*IETF-Radius-Framed-IP-Address* にマッピングされ、IP アドレスの割り当てに使用される）、設定はこの段階で失敗します。

```
Validating address: 10.1.1.1
CSTP state = WAIT_FOR_ADDRESS
webvpn_cstp_accept_address: 10.1.1.1/255.255.255.128
webvpn_cstp_accept_ipv6_address: No IPv6 Address
CSTP state = HAVE_ADDRESS
```

WebVPN のセッションが完了します。

```
SVC: NP setup
np_svc_create_session(0x1E000, 0xb5eafa80, TRUE)
webvpn_svc_np_setup
SVC ACL Name: AAA-user-cisco-E0CF3C05
SVC ACL ID: 5
SVC ACL ID: 5
vpn_put_uauth success!
SVC IPv6 ACL Name: NULL
SVC IPv6 ACL ID: -1
SVC: adding to sessmgmt
SVC: Sending response
Sending X-CSTP-FW-RULE msgs: Start
```

```
Sending X-CSTP-FW-RULE msgs: Done
Sending X-CSTP-Quarantine: false
Sending X-CSTP-Disable-Always-On-VPN: false
Unable to initiate NAC, NAC might not be enabled or invalid policy
CSTP state = CONNECTED
```

ASA の認証と認可の分離

認証と認可のプロセスを分離することが推奨される場合があります。 たとえば、ローカルに定義されたユーザにパスワード認証を使用します。 次に、ローカル認証の成功後、LDAP サーバからすべてのユーザ属性を取得します。

```
SVC: NP setup
np_svc_create_session(0x1E000, 0xb5eafa80, TRUE)
webvpn_svc_np_setup
SVC ACL Name: AAA-user-cisco-E0CF3C05
SVC ACL ID: 5
SVC ACL ID: 5
vpn_put_uauth success!
SVC IPv6 ACL Name: NULL
SVC IPv6 ACL ID: -1
SVC: adding to sessmgmt
SVC: Sending response
Sending X-CSTP-FW-RULE msgs: Start
Sending X-CSTP-FW-RULE msgs: Done
Sending X-CSTP-Quarantine: false
Sending X-CSTP-Disable-Always-On-VPN: false
Unable to initiate NAC, NAC might not be enabled or invalid policy
CSTP state = CONNECTED
```

違いは、LDAP セッションにあります。 前の例の ASA の場合

- マネージャのクレデンシャルで OpenLDAP にバインド
- ユーザ *cisco* の検索を実行
- Cisco のクレデンシャルで OpenLDAP にバインド（簡単な認証）

現在は、LDAP 認可を使用する場合、ユーザはローカル データベースによってすでに認証されているため、3 番目のステップは不要になります。

より一般的なシナリオには、認証プロセス用の RSA トークンの使用と、認可用の LDAP/AD 属性の使用が含まれます。

LDAP とローカル グループからの ASA 属性

LDAP 属性と RADIUS 属性の違いを理解することが重要です。

LDAP を使用する場合、ASA では、*RADIUS 属性*にマッピングできません。 たとえば、RADIUS を使用するときに、*cisco-av-pair* 属性 217（Address-Pools）を返すことが可能です。 その属性は IP アドレスを割り当てるために使用されるローカルに設定された IP アドレスのプールを定義します。

LDAP マッピングを使用すると、この特定の *cisco-av-pair* 属性を使用できます。 *LDAP マッピングによる cisco-av-pair* 属性は、さまざまなタイプの ACL を指定するためにのみ使用できます。

LDAP には、このような制限があるため RADIUS ほど柔軟ではありません。 LDAP からマッピン

グできない属性（アドレス プールなど）を付けて、ローカル的に定義されたグループ ポリシーを ASA で作成することにより、このための対応策とすることができます。 LDAP ユーザが認証されると、このグループ ポリシー（この例では POLICY1）に割り当てられ、ユーザ固有でない属性がグループ ポリシーから取得されます。

LDAP マッピングでサポートされる属性の完全なリストは、次のドキュメントで確認できます。 『CLI を使用した Cisco ASA 5500 シリーズ設定ガイド、8.4 および 8.6』

ASA でサポートされている RADIUS VPN3000 の属性の詳細なリストと比較できます。 次のドキュメントを参照してください。 『CLI を使用した Cisco ASA 5500 シリーズ設定ガイド、8.4 および 8.6』

ASA でサポートされている RADIUS IETF 属性の完全なリストについては、次のドキュメントを参照してください。 『CLI を使用した Cisco ASA 5500 シリーズ設定ガイド、8.4 および 8.6』

ASA と証明書認証を行う LDAP

ASA では、Anyconnect で提供される証明書を使用した LDAP 証明書属性の検索およびバイナリ比較をサポートしていません。 VPN 認証はネットワーク アクセス デバイス（NAD）で終了されるため、この機能は、Cisco ACS または ISE 用（および 802.1X サプリカント専用）に予約されています。

別の解決策があります。 ユーザ認証で証明書を使用する場合は、ASA で証明書の検証が実行され、証明書からの特定のフィールド（CN など）に基づいて LDAP 属性を取得できます。

```
SVC: NP setup
np_svc_create_session(0x1E000, 0xb5eafa80, TRUE)
webvpn_svc_np_setup
SVC ACL Name: AAA-user-cisco-E0CF3C05
SVC ACL ID: 5
SVC ACL ID: 5
vpn_put_uauth success!
SVC IPv6 ACL Name: NULL
SVC IPv6 ACL ID: -1
SVC: adding to sessmgmt
SVC: Sending response
Sending X-CSTP-FW-RULE msgs: Start
Sending X-CSTP-FW-RULE msgs: Done
Sending X-CSTP-Quarantine: false
Sending X-CSTP-Disable-Always-On-VPN: false
Unable to initiate NAC, NAC might not be enabled or invalid policy
CSTP state = CONNECTED
```

ユーザ証明書が ASA によって検証された後、LDAP 認可が実行され、ユーザ属性（CN フィールドから）が取得されて適用されます。

デバッグ

ユーザ証明書が使用されました。 cn=test1,ou=Security,o=Cisco,l=Krakow,st=PL,c=PL

証明書マッピングは、RA トンネル グループにその証明書をマッピングするように設定されています。

```
SVC: NP setup
```

```
np_svc_create_session(0x1E000, 0xb5eafa80, TRUE)
webvpn_svc_np_setup
SVC ACL Name: AAA-user-cisco-E0CF3C05
SVC ACL ID: 5
SVC ACL ID: 5
vpn_put_uauth success!
SVC IPv6 ACL Name: NULL
SVC IPv6 ACL ID: -1
SVC: adding to sessmgmt
SVC: Sending response
Sending X-CSTP-FW-RULE msgs: Start
Sending X-CSTP-FW-RULE msgs: Done
Sending X-CSTP-Quarantine: false
Sending X-CSTP-Disable-Always-On-VPN: false
Unable to initiate NAC, NAC might not be enabled or invalid policy
CSTP state = CONNECTED
```

証明書の検証とマッピング

```
ASA# show debug
debug ldap  enabled at level 255
debug webvpn anyconnect enabled at level 254
debug crypto ca enabled at level 3
debug crypto ca messages enabled at level 3
debug crypto ca transactions enabled at level 3Apr 09 2013 17:31:32: %ASA-7-717025: Validating
certificate chain containing 1 certificate(s).Apr 09 2013 17:31:32: %ASA-7-717029: Identified
client certificate within certificate chain. serial number: 00FE9C3D61E131CDB1, subject name:
cn=test1,ou=Security,o=Cisco,l=Krakow,st=PL,c=PL.Apr 09 2013 17:31:32: %ASA-6-717022:
Certificate was successfully validated. Certificate is resident and trusted, serial number:
00FE9C3D61E131CDB1, subject name: cn=test1,ou=Security,o=Cisco,l=Krakow,st=PL,c=PL.Apr 09 2013
17:31:32: %ASA-6-717028: Certificate chain was successfully validated with revocation status
check.Apr 09 2013 17:31:32: %ASA-6-717028: Certificate chain was successfully validated with
revocation status check.Apr 09 2013 17:31:32: %ASA-7-717036: Looking for a tunnel group match
based on certificate maps for peer certificate with serial number: 00FE9C3D61E131CDB1, subject
name: cn=test1,ou=Security,o=Cisco,l=Krakow,st=PL,c=PL, issuer_name:
cn=TAC,ou=RAC,o=TAC,l=Warsaw,st=Maz,c=PL.Apr 09 2013 17:31:32: %ASA-7-717038: Tunnel group match
found. Tunnel Group: RA, Peer certificate: serial number: 00FE9C3D61E131CDB1, subject name:
cn=test1,ou=Security,o=Cisco,l=Krakow,st=PL,c=PL, issuer_name:
cn=TAC,ou=RAC,o=TAC,l=Warsaw,st=Maz,c=PL.
```

証明書からのユーザ名の抽出と LDAP を使用した認可

```
Apr 09 2013 17:31:32: %ASA-7-113028: Extraction of username from VPN client certificate has been
requested. [Request 53]Apr 09 2013 17:31:32: %ASA-7-113028: Extraction of username from VPN
client certificate has been requested. [Request 53]Apr 09 2013 17:31:32: %ASA-7-113028:
Extraction of username from VPN client certificate has been requested. [Request 53]Apr 09 2013
17:31:32: %ASA-7-113028: Extraction of username from VPN client certificate has been requested.
[Request 53]Apr 09 2013 17:31:32: %ASA-7-113028: Extraction of username from VPN client
certificate has been requested. [Request 53]Apr 09 2013 17:31:32: %ASA-6-113004: AAA user
authorization Successful : server =  192.168.11.10 : user = test1Apr 09 2013 17:31:32: %ASA-6-
113004: AAA user authorization Successful : server =  192.168.11.10 : user = test1Apr 09 2013
17:31:32: %ASA-6-113004: AAA user authorization Successful : server =  192.168.11.10 : user =
test1Apr 09 2013 17:31:32: %ASA-6-113004: AAA user authorization Successful : server =
192.168.11.10 : user = test1Apr 09 2013 17:31:32: %ASA-6-113004: AAA user authorization
Successful : server =  192.168.11.10 : user = test1
```

LDAP からの属性の取得

```
Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute
aaa.ldap.cn = John SmithApr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88:
Session Attribute aaa.ldap.givenName = JohnApr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1,
Addr 192.168.1.88: Session Attribute aaa.ldap.sn = test1Apr 09 2013 17:31:32: %ASA-7-734003:
DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.uid = test1Apr 09 2013 17:31:32:
%ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.uidNumber =
10000Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute
```

aaa.ldap.**gidNumber = 10000**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**homeDirectory = /home/cisco**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**mail = jsmith@dev.local**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**objectClass.1 = top**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**objectClass.2 = posixAccount**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**objectClass.3 = shadowAccount**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**objectClass.4 = inetOrgPerson**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**objectClass.5 = organizationalPerson**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**objectClass.6 = person**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**objectClass.7 = CiscoPerson**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**loginShell = /bin/bash**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**userPassword = {CRYPT}***Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**CiscoBanner = This is banner 1**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**CiscoIPAddress = 10.1.1.1**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**CiscoIPNetmask = 255.255.255.128**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**CiscoDomain = domain1.com**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**CiscoDNS = 10.6.6.6**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**CiscoACLin = ip:inacl#1=permit ip 10.1.1.0 255.255.255.128 10.11.11.0 255.255.255.0**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**CiscoSplitACL = ACL1**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**CiscoSplitTunnelPolicy = 1**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.ldap.**CiscoGroupPolicy = POLICY1**

Cisco による属性マッピング

Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.**cisco.grouppolicy = POLICY1**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.**cisco.ipaddress = 10.1.1.1**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.**cisco.username = test1**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.**cisco.username1 = test1**Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.**cisco.username2 =** Apr 09 2013 17:31:32: %ASA-7-734003: DAP: User test1, Addr 192.168.1.88: Session Attribute aaa.**cisco.tunnelgroup = RA**Apr 09 2013 17:31:32: %ASA-6-734001: DAP: User test1, Addr 192.168.1.88, Connection AnyConnect: The following **DAP records** were selected for this connection: **DfltAccessPolicy**Apr 09 2013 17:31:32: %ASA-6-113039: **Group <POLICY1> User <test1> IP <192.168.1.88> AnyConnect parent session started**.Apr 09 2013 17:31:32: %ASA-6-113039: **Group <POLICY1> User <test1> IP <192.168.1.88> AnyConnect parent session started**.

第 2 の認証

2 要素認証が必要な場合は、LDAP 認証および認可とともにトークン パスワードを使用することが可能です。

Apr 09 2013 17:31:32: %ASA-6-113039: **Group <POLICY1> User <test1> IP <192.168.1.88> AnyConnect parent session started**.

次に、ユーザは RSA からのユーザ名とパスワード（ユーザの持ち物、トークン）、と LDAP ユーザ名/パスワード（ユーザの知識）を提示する必要があります。 第 2 の認証として、証明書からのユーザ名を使用することも可能です。 二重認証の詳細については、『CLI を使用した Cisco ASA 5500 シリーズ設定ガイド、8.4 および 8.6』を参照してください。

関連情報

- 『CLI を使用した Cisco ASA 5500 シリーズ設定ガイド、8.4 および 8.6』

- OpenLDAP ソフトウェア 2.4 管理者ガイド
- 民間企業番号
- テクニカルサポートとドキュメント - Cisco Systems