ASA 8.0:WebVPN 사용자에 대한 RADIUS 인증 구성

목차

소개

이 문서에서는 WebVPN 사용자 인증을 위해 원격 인증 전화 접속 사용자 서비스(RADIUS) 서버를 사용하도록 Cisco ASA(Adaptive Security Appliance)를 구성하는 방법을 보여 줍니다.이 예에서 RADIUS 서버는 Cisco ACS(Access Control Server) 버전 4.1입니다. 이 구성은 소프트웨어 버전 8.0(2)을 실행하는 ASA에서 ASDM(Adaptive Security Device Manager) 6.0(2)을 사용하여 수행됩니다.

참고: 이 예에서 RADIUS 인증은 WebVPN 사용자에 대해 구성되지만 이 컨피그레이션은 다른 유형의 원격 액세스 VPN에도 사용할 수 있습니다.표시된 대로 원하는 연결 프로파일(터널 그룹)에 AAA 서버 그룹을 할당하기만 하면 됩니다.

사전 요구 사항

- 기본 WebVPN 컨피그레이션이 필요합니다.
- Cisco ACS에는 사용자 인증을 위해 구성된 사용자가 있어야 합니다.자세한 내용은 사용자 관리의 기본 사용자 계정 추가 섹션을 참조하십시오.

ACS 서버 구성

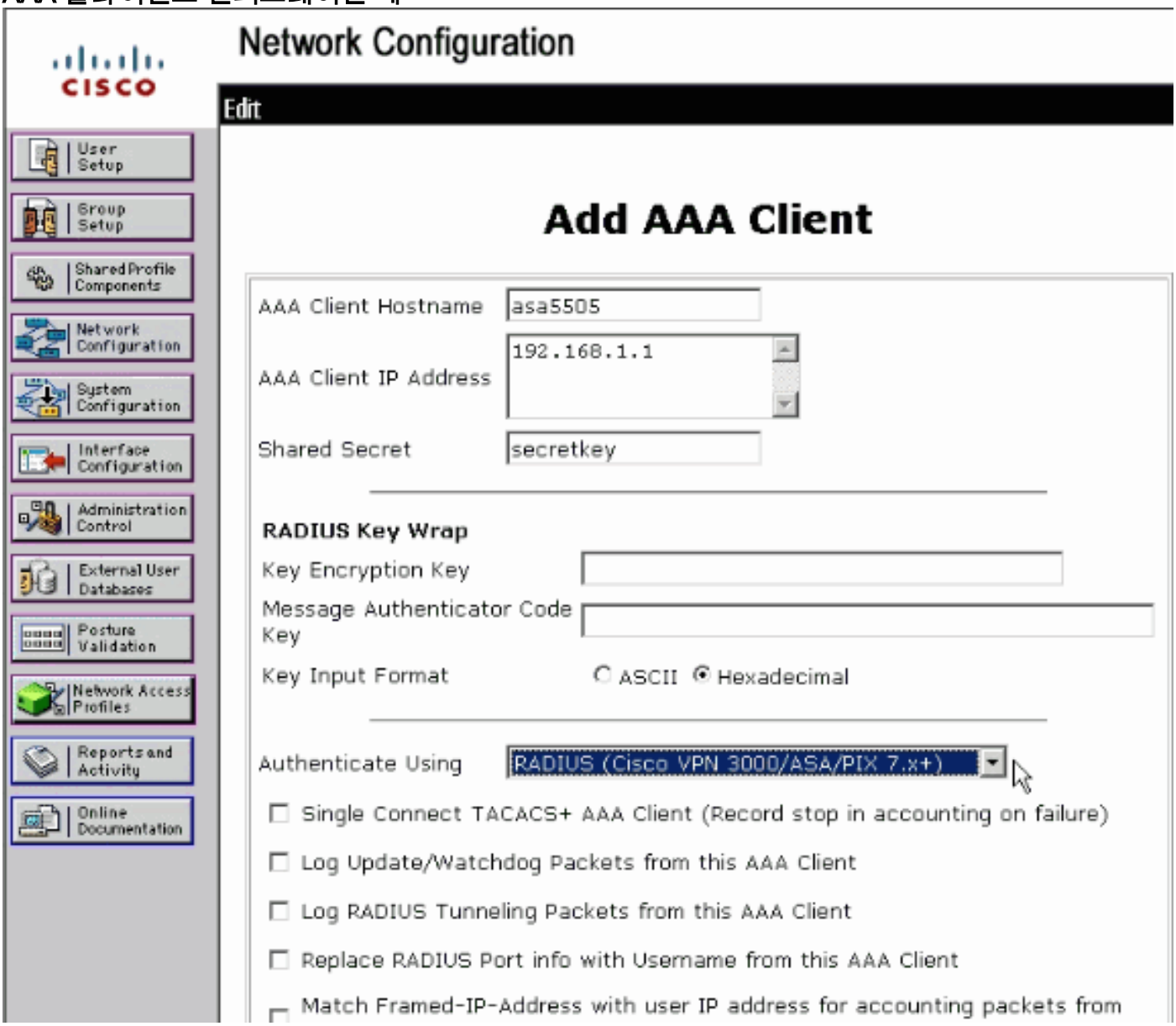
이 섹션에서는 ACS 및 ASA에서 RADIUS 인증을 구성하는 정보를 제공합니다.

ACS 서버가 ASA와 통신하도록 구성하려면 다음 단계를 완료합니다.

1. ACS 화면 왼쪽 메뉴에서 Network Configuration(네트워크 컨피그레이션)을 선택합니다.

2. AAA **Clients(AAA 클라이언트)** 아래에서 Add Entry(항목 추가)**를 선택합니다**.
3. 클라이언트 정보를 제공합니다.**AAA Client Hostname**(AAA 클라이언트 호스트 이름) - 선택한 이름**AAA Client IP Address(AAA 클라이언트 IP 주소)** - 보안 어플라이언스가 ACS에 연결하는 주소**Shared Secret**(공유 암호) - ACS 및 보안 어플라이언스에 구성된 비밀 키
4. Authenticate Using(**인증 사용**) 드롭다운**에서 RADIUS(Cisco VPN 3000/ASA/PIX 7.x+)를 선택합니다**.
5. Submit **+Apply를 클릭합니다**.

AAA 클라이언트 컨피그레이션 예



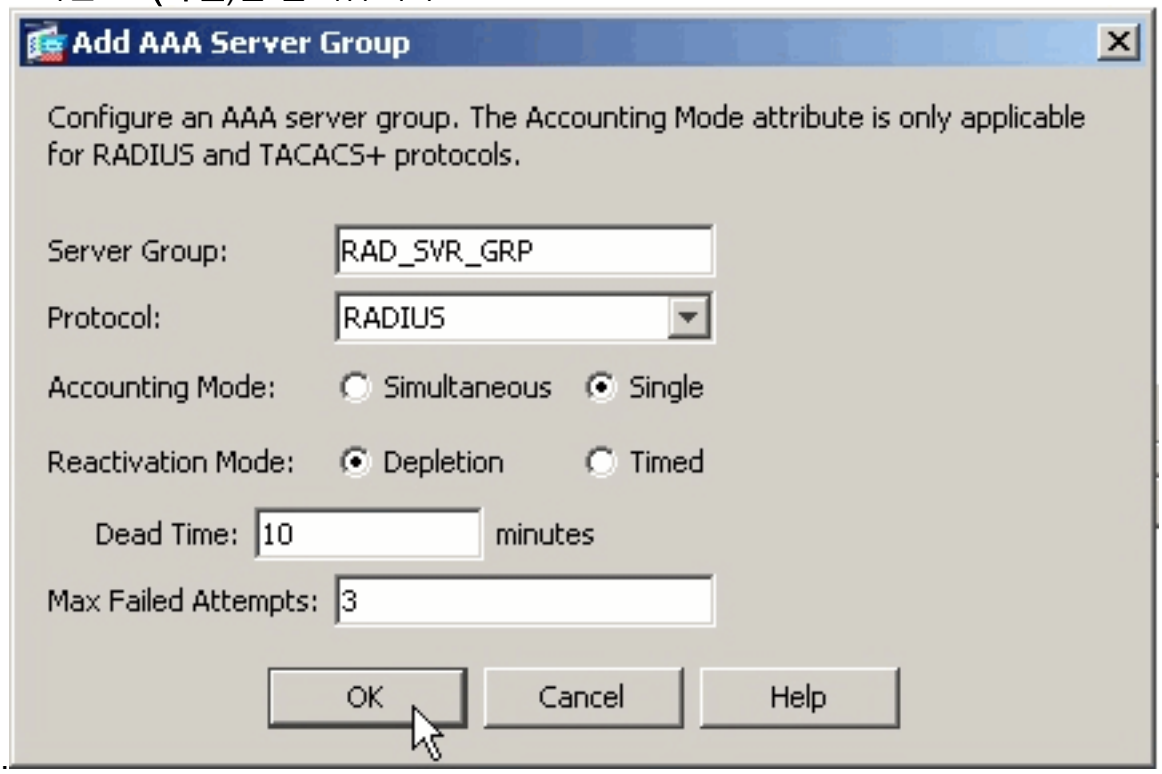
보안 어플라이언스 구성

ASDM

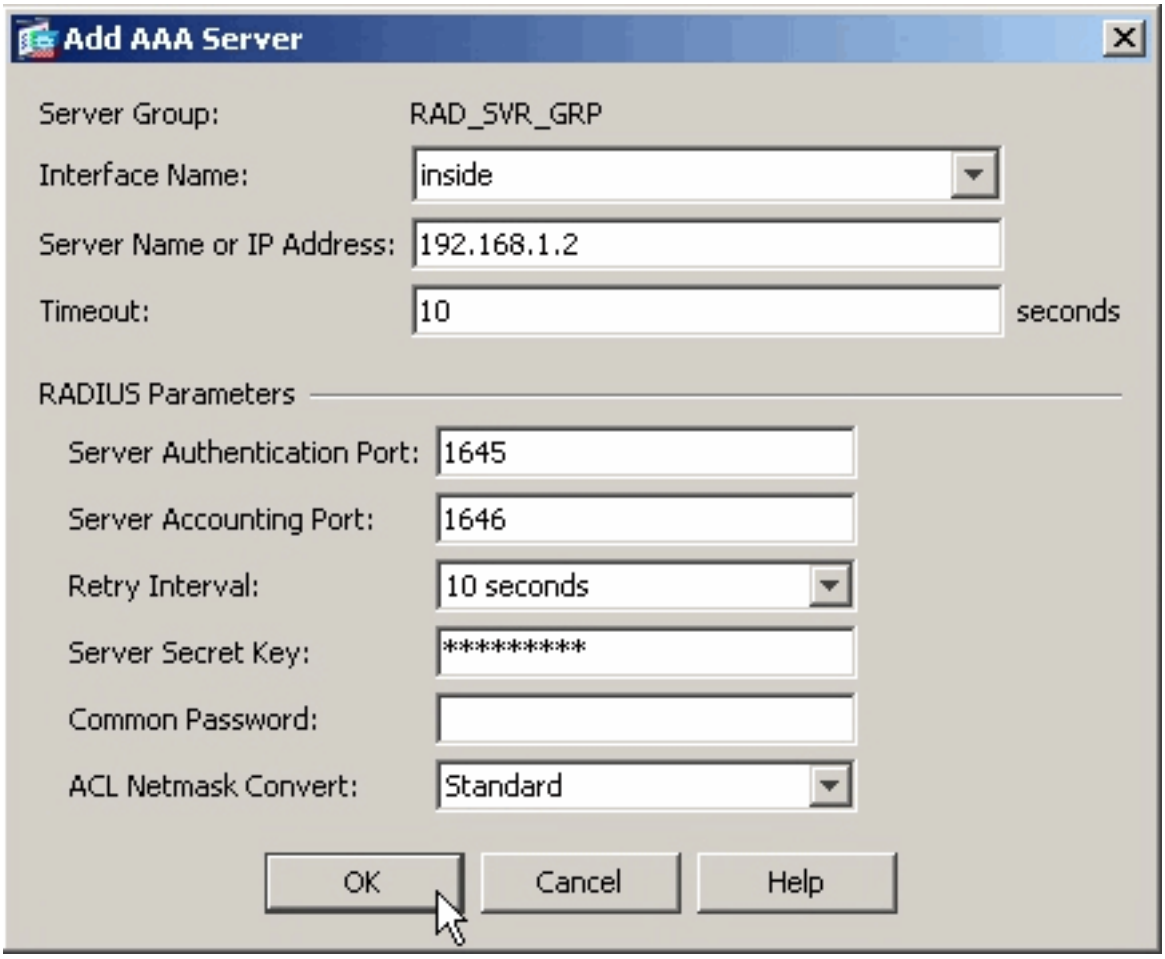
ASA가 ACS 서버와 통신하고 WebVPN 클라이언트를 인증하도록 구성하려면 ASDM에서 다음 단계를 완료합니다.

1. Configuration(**컨피그레이션) > Remote Access VPN(원격 액세스 VPN) > AAA Setup(AAA 설정) > AAA Server Groups(AAA 서버 그룹)를 선택합니다**.

2. AAA Server Groups 옆에 있는 Add를 클릭합니다.
3. 표시되는 창에서 새 AAA 서버 그룹의 이름을 지정하고 **RADIUS**를 프로토콜로 선택합니다.완료되면 **OK(확인)**를 클릭합니다

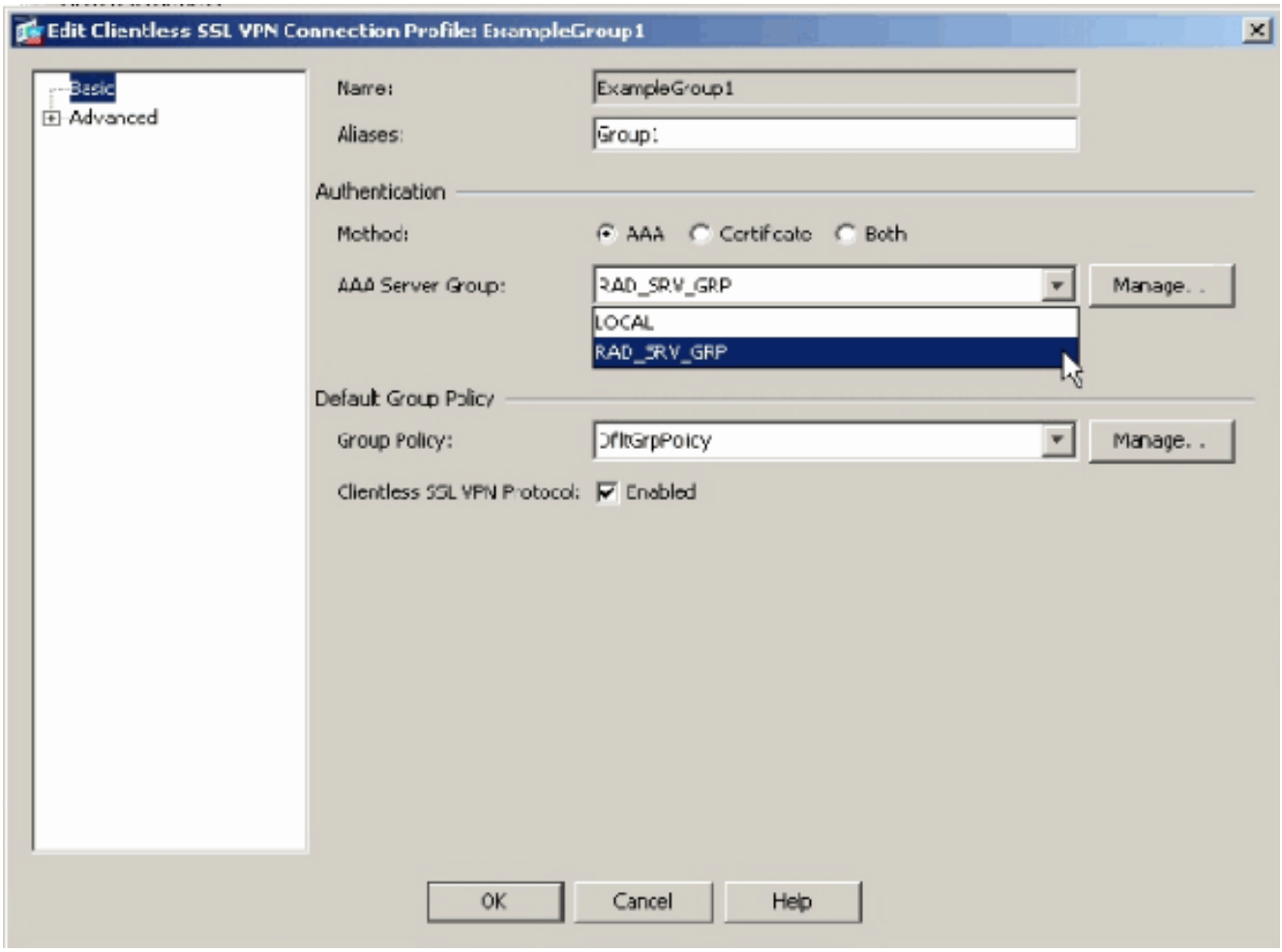


4. 맨 위 창에서 새 그룹이 선택되었는지 확인하고 아래쪽 창 오른쪽에 추가를 클릭합니다.
5. 서버 정보 제공:**Interface Name**(인터페이스 이름) - ASA가 ACS 서버에 연결하기 위해 사용해야 하는 인터페이스**Server Name or IP address(서버 이름 또는 IP 주소)** - ASA가 ACS 서버에 연결하기 위해 사용해야 하는 주소입니다.**Server Secret Key**(서버 비밀 키) - ACS 서버의 ASA에 대해 구성된 공유 비밀 키**ASA의 AAA 서버 컨피그레이션 예**



6. AAA 서버 그룹 및 서버를 구성했으면 Configuration(구성) > Remote Access VPN(원격 액세스 VPN) > Clientless SSL VPN Access(클라이언트리스 SSL VPN 액세스) > Connection Profiles(연결 프로파일)로 이동하여 WebVPN에서 새 AAA 컨피그레이션을 사용하도록 구성합니다.**참고:** 이 예에서는 WebVPN을 사용하지만 이 AAA 설정을 사용하도록 모든 원격 액세스 연결 프로파일(터널 그룹)을 설정할 수 있습니다.
7. AAA를 구성할 프로필을 선택하고 Edit(수정)를 **클릭합니다**.
8. Authentication(**인증**)에서 이전에 생성한 RADIUS 서버 그룹을 선택합니다.완료되면 **OK(확인**)를 클릭합니다

.



명령줄 인터페이스

ASA가 ACS 서버와 통신하고 WebVPN 클라이언트를 인증하도록 구성하려면 CLI(Command Line Interface)에서 다음 단계를 완료합니다.

```
ciscoasa#configure terminal

!--- Configure the AAA Server group. ciscoasa(config)# aaa-server RAD_SRV_GRP protocol RADIUS
ciscoasa(config-aaa-server-group)# exit !--- Configure the AAA Server. ciscoasa(config)# aaa-
server RAD_SRV_GRP (inside) host 192.168.1.2 ciscoasa(config-aaa-server-host)# key secretkey
ciscoasa(config-aaa-server-host)# exit !--- Configure the tunnel group to use the new AAA setup.
ciscoasa(config)# tunnel-group ExampleGroup1 general-attributes ciscoasa(config-tunnel-general)#
authentication-server-group RAD_SRV_GRP
```

다음을 확인합니다.

이 섹션을 사용하여 컨피그레이션이 제대로 작동하는지 확인합니다.

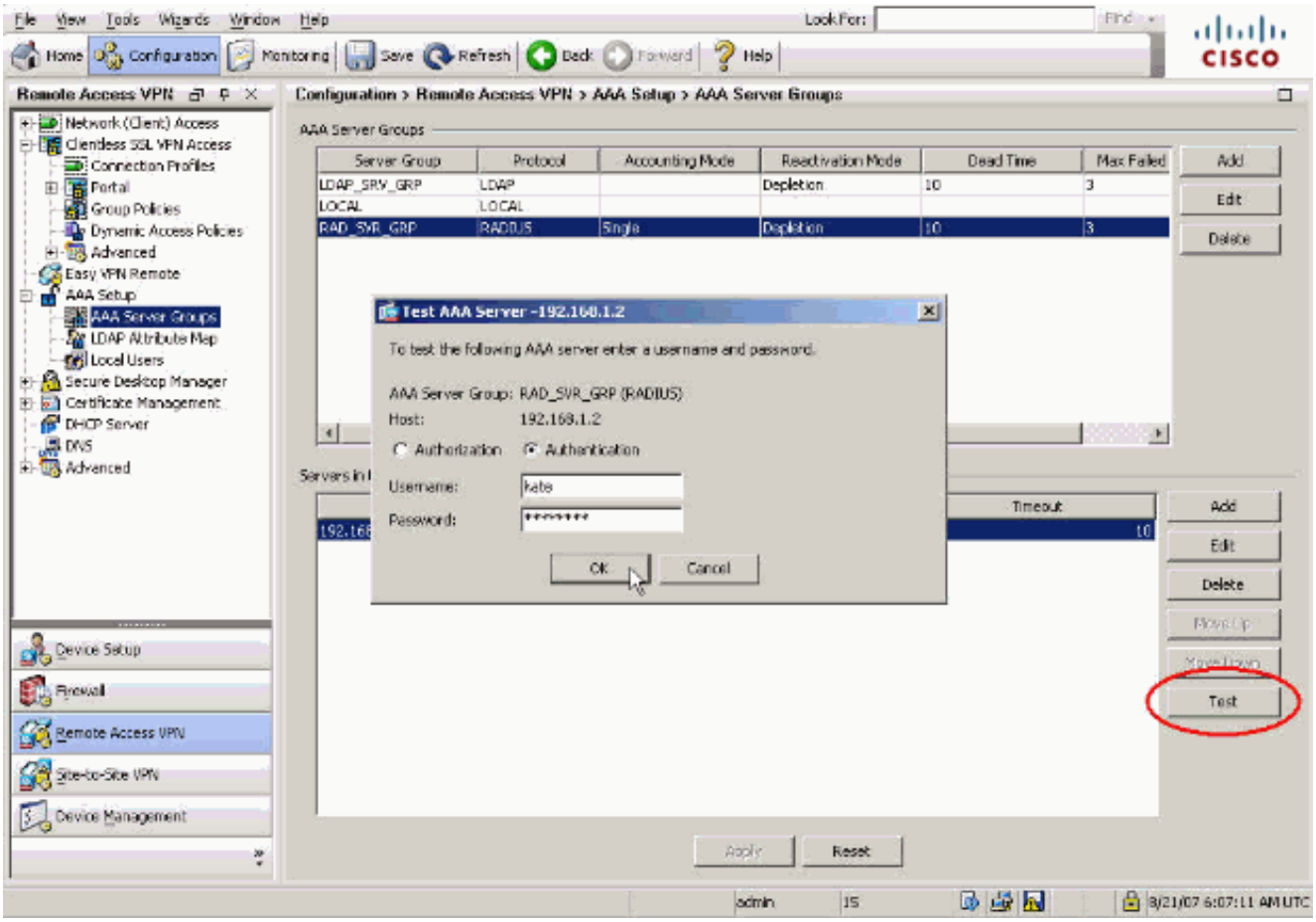
ASDM을 사용한 테스트

AAA Server Groups 컨피그레이션 화면의 **Test** 버튼을 사용하여 RADIUS 컨피그레이션을 확인합니다.사용자 이름과 비밀번호를 입력하면 이 버튼을 사용하여 ACS 서버에 테스트 인증 요청을 보낼 수 있습니다.

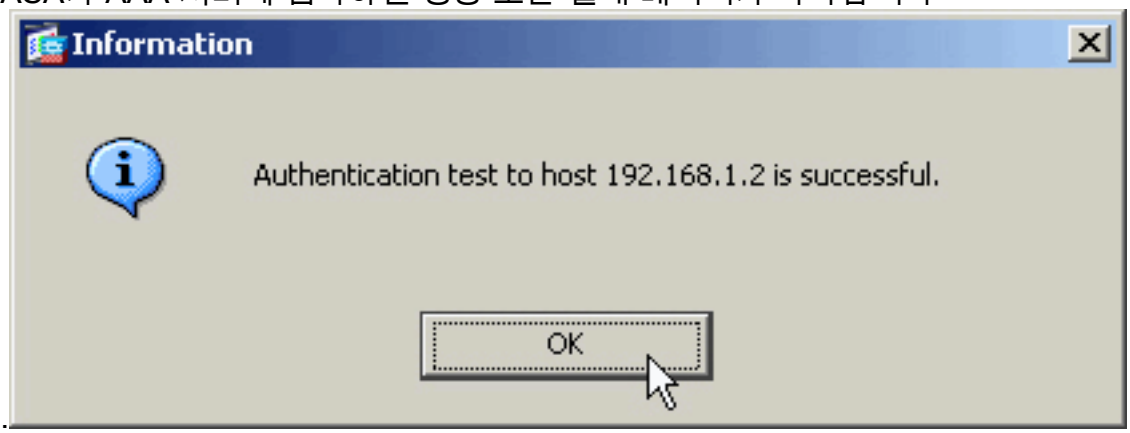
1. Configuration(**컨피그레이션**) > **Remote Access VPN(원격 액세스 VPN)** > **AAA Setup(AAA 설정)** > **AAA Server Groups(AAA 서버 그룹)**를 선택합니다.

2. 상단 창에서 원하는 AAA 서버 그룹을 선택합니다.
3. 하단 창에서 테스트할 AAA 서버를 선택합니다.
4. 아래쪽 창 오른쪽의 **Test** 버튼을 클릭합니다.
5. 표시되는 창에서 **Authentication** 라디오 버튼을 클릭하고 테스트할 자격 증명을 입력합니다.완료되면 **OK(확인)**를 클릭합니다

.



6. ASA가 AAA 서버에 접속하면 성공 또는 실패 메시지가 나타납니다



.

CLI로 테스트

명령행에서 **test** 명령을 사용하여 AAA 설정을 테스트할 수 있습니다.테스트 요청이 AAA 서버로 전송되고 그 결과가 명령줄에 나타납니다.

```
ciscoasa#test aaa-server authentication RAD_SVR_GRP host 192.168.1.2 username kate password
cisco123
INFO: Attempting Authentication test to IP address <192.168.1.2> (timeout: 12 seconds)
```

```
INFO: Authentication Successful
```

문제 해결

debug **radius** 명령을 사용하면 이 시나리오의 인증 문제를 해결할 수 있습니다.이 명령은 RADIUS 세션 디버깅 및 RADIUS 패킷 디코딩을 활성화합니다.표시된 각 디버그 출력에서 디코딩된 첫 번째 패킷은 ASA에서 ACS 서버로 전송된 패킷입니다.두 번째 패킷은 ACS 서버의 응답입니다.

참고: debug **명령**을 사용하기 전에 디버그 명령에 대한 중요 정보를 참조하십시오.

인증에 성공하면 RADIUS 서버는 **access-accept** 메시지를 전송합니다.

```
ciscoasa#debug radius

!--- First Packet. Authentication Request. ciscoassa#radius mkreq: 0x88 alloc_rip 0xd5627ae4 new
request 0x88 --> 52 (0xd5627ae4) got user '' got password add_req 0xd5627ae4 session 0x88 id 52
RADIUS_REQUEST radius.c: rad_mkpkt RADIUS packet decode (authentication request) ---------------
---------------------- Raw packet data (length = 62)..... 01 34 00 3e 18 71 56 d7 c4 ad e2 73
30 a9 2e cf | .4.>.qV....s0... 5c 65 3a eb 01 06 6b 61 74 65 02 12 0e c1 28 b7 |
\e:...kate....(. 87 26 ed be 7b 2c 7a 06 7c a3 73 19 04 06 c0 a8 | .&..{,z.|.s..... 01 01 05 06
00 00 00 34 3d 06 00 00 00 05 | .......4=..... Parsed packet data..... Radius: Code = 1 (0x01)
Radius: Identifier = 52 (0x34) Radius: Length = 62 (0x003E) Radius: Vector:
187156D7C4ADE27330A92ECF5C653AEB Radius: Type = 1 (0x01) User-Name Radius: Length = 6 (0x06)
Radius: Value (String) = 6b 61 74 65 | kate Radius: Type = 2 (0x02) User-Password Radius: Length
= 18 (0x12) Radius: Value (String) = 0e c1 28 b7 87 26 ed be 7b 2c 7a 06 7c a3 73 19 |
..(..&..{,z.|.s. Radius: Type = 4 (0x04) NAS-IP-Address Radius: Length = 6 (0x06) Radius: Value
(IP Address) = 192.168.1.1 (0xC0A80101) Radius: Type = 5 (0x05) NAS-Port Radius: Length = 6
(0x06) Radius: Value (Hex) = 0x34 Radius: Type = 61 (0x3D) NAS-Port-Type Radius: Length = 6
(0x06) Radius: Value (Hex) = 0x5 send pkt 192.168.1.2/1645 rip 0xd5627ae4 state 7 id 52
rad_vrfy() : response message verified rip 0xd544d2e8 : chall_state '' : state 0x7 : timer 0x0 :
reqauth: 18 71 56 d7 c4 ad e2 73 30 a9 2e cf 5c 65 3a eb : info 0x88 session_id 0x88 request_id
0x34 user 'kate' response '***' app 0 reason 0 skey 'secretkey' sip 192.168.1.2 type 1 !---
Second Packet. Authentication Response. RADIUS packet decode (response) ------------------------
-------------- Raw packet data (length = 50)..... 02 34 00 32 35 a1 88 2f 8a bf 2a 14 c5 31 78
59 | .4.25../..*..1xY 60 31 35 89 08 06 ff ff ff ff 19 18 43 41 43 53 | `15.........CACS 3a 30
2f 32 61 36 2f 63 30 61 38 30 31 30 31 2f | :0/2a6/c0a80101/ 35 32 | 52 Parsed packet data.....
Radius: Code = 2 (0x02) Radius: Identifier = 52 (0x34) Radius: Length = 50 (0x0032) Radius:
Vector: 35A1882F8ABF2A14C531785960313589 Radius: Type = 8 (0x08) Framed-IP-Address Radius:
Length = 6 (0x06) Radius: Value (IP Address) = 255.255.255.255 (0xFFFFFFFF) Radius: Type = 25
(0x19) Class Radius: Length = 24 (0x18) Radius: Value (String) = 43 41 43 53 3a 30 2f 32 61 36
2f 63 30 61 38 30 | CACS:0/2a6/c0a80 31 30 31 2f 35 32 | 101/52 rad_procpkt: ACCEPT
RADIUS_ACCESS_ACCEPT: normal termination
RADIUS_DELETE
remove_req 0xd5627ae4 session 0x88 id 52
free_rip 0xd5627ae4
radius: send queue empty
```

인증이 실패하면 ACS 서버가 **액세스 거부** 메시지를 전송합니다.

```
ciscoasa#debug radius

!--- First Packet. Authentication Request. ciscoasa# radius mkreq: 0x85 alloc_rip 0xd5627ae4 new
request 0x85 --> 49 (0xd5627ae4) got user '' got password add_req 0xd5627ae4 session 0x85 id 49
RADIUS_REQUEST radius.c: rad_mkpkt RADIUS packet decode (authentication request) ---------------
---------------------- Raw packet data (length = 62)..... 01 31 00 3e 88 21 46 07 34 5d d2 a3
a0 59 1e ff | .1.>.!F.4]...Y.. cc 15 2a 1b 01 06 6b 61 74 65 02 12 60 eb 05 32 |
..*...kate..`..2 87 69 78 a3 ce d3 80 d8 4b 0d c3 37 04 06 c0 a8 | .ix.....K..7.... 01 01 05 06
```

00 00 00 31 3d 06 00 00 00 05 | ....1=..... Parsed packet data..... Radius: Code = 1 (0x01) Radius: Identifier = 49 (0x31) Radius: Length = 62 (0x003E) Radius: Vector: 88214607345DD2A3A0591EFFCC152A1B Radius: Type = 1 (0x01) User-Name Radius: Length = 6 (0x06) Radius: Value (String) = 6b 61 74 65 | kate Radius: Type = 2 (0x02) User-Password Radius: Length = 18 (0x12) Radius: Value (String) = 60 eb 05 32 87 69 78 a3 ce d3 80 d8 4b 0d c3 37 | `..2.ix.....K..7 Radius: Type = 4 (0x04) NAS-IP-Address Radius: Length = 6 (0x06) Radius: Value (IP Address) = 192.168.1.1 (0xC0A80101) Radius: Type = 5 (0x05) NAS-Port Radius: Length = 6 (0x06) Radius: Value (Hex) = 0x31 Radius: Type = 61 (0x3D) NAS-Port-Type Radius: Length = 6 (0x06) Radius: Value (Hex) = 0x5 send pkt 192.168.1.2/1645 rip 0xd5627ae4 state 7 id 49
rad_vrfy() : response message verified rip 0xd544d2e8 : chall_state '' : state 0x7 : timer 0x0 : reqauth: 88 21 46 07 34 5d d2 a3 a0 59 1e ff cc 15 2a 1b : info 0x85 session_id 0x85 request_id 0x31 user 'kate' response '***' app 0 reason 0 skey 'secretkey' sip 192.168.1.2 type 1 *!--- Second packet. Authentication Response.* RADIUS packet decode (response) ---------- Raw packet data (length = 32)..... 03 31 00 20 70 98 50 af 39 cc b9 ba df a7 bd ff | .1. p.P.9..... 06 af fb 02 12 0c 52 65 6a 65 63 74 65 64 0a 0d | ......Rejected.. Parsed packet data..... Radius: Code = 3 (0x03) Radius: Identifier = 49 (0x31) Radius: Length = 32 (0x0020) Radius: Vector: 709850AF39CCB9BADFA7BDFF06AFFB02 Radius: Type = 18 (0x12) Reply-Message Radius: Length = 12 (0x0C) **Radius: Value (String) =**
52 65 6a 65 63 74 65 64 0a 0d | Rejected..
rad_procpkt: REJECT
RADIUS_DELETE
remove_req 0xd5627ae4 session 0x85 id 49
free_rip 0xd5627ae4
radius: send queue empty

관련 정보

- 원격 인증 전화 접속 사용자 서비스(RADIUS)
- RFC(Request for Comments)
- 기술 지원 및 문서 – Cisco Systems