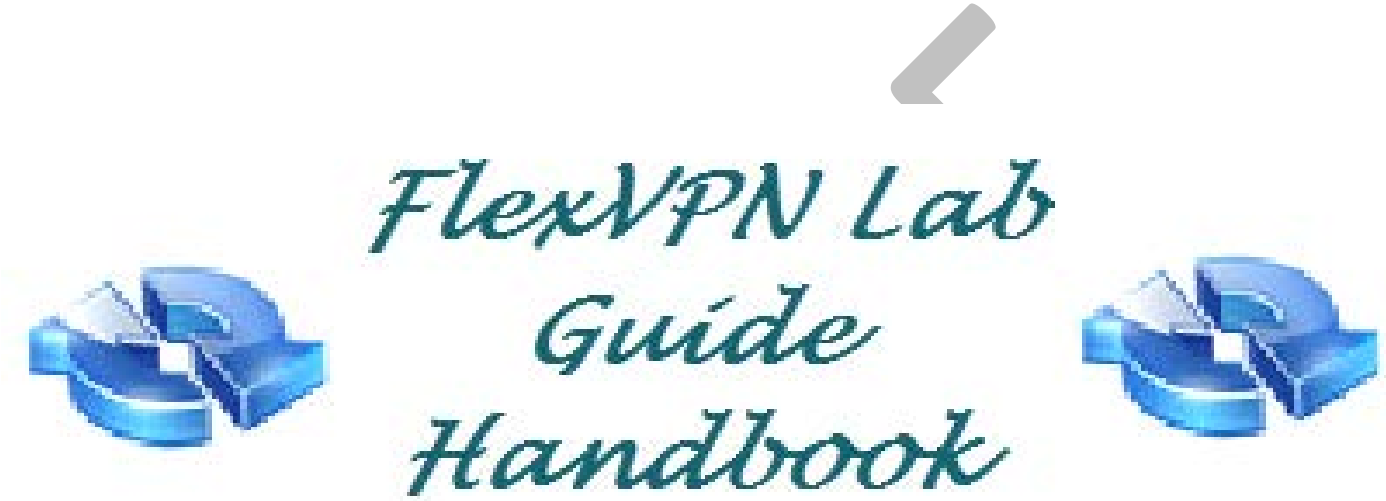
FlexVPN Lab Guide/Handbook



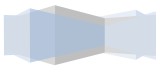
Authored by :

Tariq Ahmad
CCIEx2 # 26141
(Voice,Security)

- **Scenarios/Table of Contents :**

- Advanced HUB & SPOKE (IKEv2 Routing with RADIUS Authorization)
  - Certificate on HUB, PSK on SPOKES
  - Virtual-Access Interface : HUB
  - Static Tunnel Interface : SPOKE1/SPOKE2
  - Config Exchange / IKEv2 Routing
- Basic HUB & SPOKE (IKEv2 Routing with LOCAL Authorization)
  - PSK on HUB, PSK on SPOKES
  - Virtual-Access Interface : HUB
  - Static Tunnel Interface : SPOKE1/SPOKE2
  - Config Exchange / IKEv2 Routing
- Advanced HUB & SPOKE (IKEv2 Routing with LOCAL Authorization)
  - CERT on HUB, CERT on SPOKES
  - Virtual-Access Interface : HUB
  - Static Tunnel Interface : SPOKE1/SPOKE2
  - Config Exchange / IKEv2 Routing
- Advanced HUB & SPOKE (Direct Spoke to Spoke Tunnel)
  - CERT on HUB, CERT on SPOKES
  - Virtual-Access Interface : HUB
  - Virtual-Access Interface/Static Tunnel Interface : SPOKE1/SPOKE2
  - Config Exchange / IKEv2 Routing
- FLEX VPN RESILIENCY
- DUAL-HUB (FLEX VPN RESILIENCY)
  - Primary HUB
  - Secondary HUB
- IKEv2 BACKUP PEERS
- DUAL-HUB (FLEX VPN RESILIENCY)
  - BACKUP PEERS
- Extensible Authentication Protocol (EAP)
- FlexVPN Remote Access VPN using EAP Authentication via Cisco Secure ACS 5X
  - EAP-MD5 : Authentication Protocol
  - Cisco AnyConnect Secure Mobility Client v3

- FlexVPN Remote Access VPN using EAP Authentication via Cisco Identity Services Engine (ISE) 1.1.X
  - EAP-MD5 : Authentication Protocol
  - Cisco AnyConnect Secure Mobility Client v3

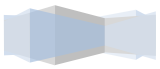


FlexVPN – a Unified Overlay VPN Introduction

Introduction :

FlexVPN integrates different configuration parameters into one style hence termed as Unified Overlay VPN.
It supports following features :

- Interoperability
- Dynamic Routing
- IPSec Routing
- Spoke to Spoke direct (shortcut switching)
- Remote Access VPN
- Simple Failover
- Source based Failover
- Configuration Push
- Per-Peer Config
- Per-Peer QoS
- Full AAA Managment



Understanding IKEv2

IKEv2 Benefits :

IKEv2 offers myriad of benefits as compared to traditional IKEv1 protocol. Here are some of these listed :

- **Dead Peer Detection and Network Address Translation-Traversal**

Internet Key Exchange Version 2 (IKEv2) provides built-in support for Dead Peer Detection (DPD) and Network Address Translation-Traversal (NAT-T).

- **Certificate URLs**

Certificates can be referenced through a URL and hash, instead of being sent within IKEv2 packets, to avoid fragmentation.

- **Denial of Service Attack Resilience (Anti-DoS)**

IKEv2 does not process a request until it determines the requester, which addresses to some extent the Denial of Service (DoS) problems in IKEv1, which can be spoofed into performing substantial cryptographic (expensive) processing from false locations.

- **EAP Support**

IKEv2 allows the use of Extensible Authentication Protocol (EAP) for authentication.

- **Multiple Crypto Engines**

If your network has both IPv4 and IPv6 traffic and you have multiple crypto engines, choose one of the following configuration options:

- One engine handles IPv4 traffic and the other engine handles IPv6 traffic.
- One engine handles both IPv4 and IPv6 traffic.

- **Reliability and State Management (Windowing)**

IKEv2 uses sequence numbers and acknowledgments to provide reliability, and mandates some error-processing logistics and shared state management.

IKEv2 Suite-B Support

Suite-B is a set of cryptographic algorithms promulgated by the National Security Agency as part of its Cryptographic Modernization Program. Suite-B for Internet Key Exchange (IKE) and IPsec is defined in RFC 4869. The Suite-B components are as follows:

- **Advanced Encryption Standard (AES)** 128- and 256-bit keys configured in the IKEv2 proposal. For data traffic, AES should be used in Galois Counter Mode (GCM) that is configured in the IPsec transform set.
- **Elliptic Curve Digital Signature Algorithm (ECDSA)** configured in the IKEv2 profile.
- **Secure Hashing Algorithm 2** (SHA-256 and SHA-384) configured in the IKEv2 proposal and IPsec transform set.

Suite-B requirements comprise four user-interface suites of cryptographic algorithms for use with IKE and IPsec. Each suite consists of

- Encryption Algorithm
- Digital-Signature Algorithm
- Key-Agreement Algorithm
- Hash- or Message-Digest Algorithm.

Introducing Smart Defaults

Introduction :

IKEv2 Smart Defaults feature minimizes the FlexVPN configuration by covering most of the use cases. IKEv2 smart defaults can be customized for specific use cases, though this is not recommended.The following rules apply to the IKEv2 Smart Defaults feature:

- A default configuration is displayed in the corresponding show command with **default** as a keyword and with no argument. For example, the **show crypto ikev2 proposal default** command displays the default IKEv2 proposal and the **show crypto ikev2 proposal** command displays the default IKEv2 proposal, along with any user-configured proposals.

- A default configuration is displayed in the **show running-config all** command; it is not displayed in the **show running-config** command.

- You can modify the default configuration, which is displayed in the **show running-config all** command.

- A default configuration can be disabled using the **no** form of the command; for example, **no crypto ikev2 proposal default**. A disabled default configuration is not used in negotiation but the configuration is displayed in the **show running-config** command. A disabled default configuration loses any user modification and restores system-configured values.

- A default configuration can be reenabled using the default form of the command, which restores system-configured values; for example, **default crypto ikev2 proposal**.

- The default mode for the default transform set is **transport**; the default mode for all other transform sets is **tunnel**.

Here is the list of commands that are enabled with the IKEv2 Smart Defaults feature, along with the default values.

IKEv2 default Authorization Policy

```
LAB#show crypto ikev2 authorization policy default
 IKEv2 Authorization Policy : default
  route set interface
  route accept any tag : 1 distance : 1
```

IKEv2 default Proposal

```
LAB#show crypto ikev2 proposal default
 IKEv2 proposal: default
      Encryption : AES-CBC-256 AES-CBC-192 AES-CBC-128
      Integrity  : SHA512 SHA384 SHA256 SHA96 MD596
      PRF        : SHA512 SHA384 SHA256 SHA1 MD5
      DH Group   : DH_GROUP_1536_MODP/Group 5 DH_GROUP_1024_MODP/Group 2
```

IKEv2 default Policy

```
LAB#show crypto ikev2 policy default
 IKEv2 policy : default
       Match fvrf : any
       Match address local : any
       Proposal    : default
```

IPSEC default Transform Set

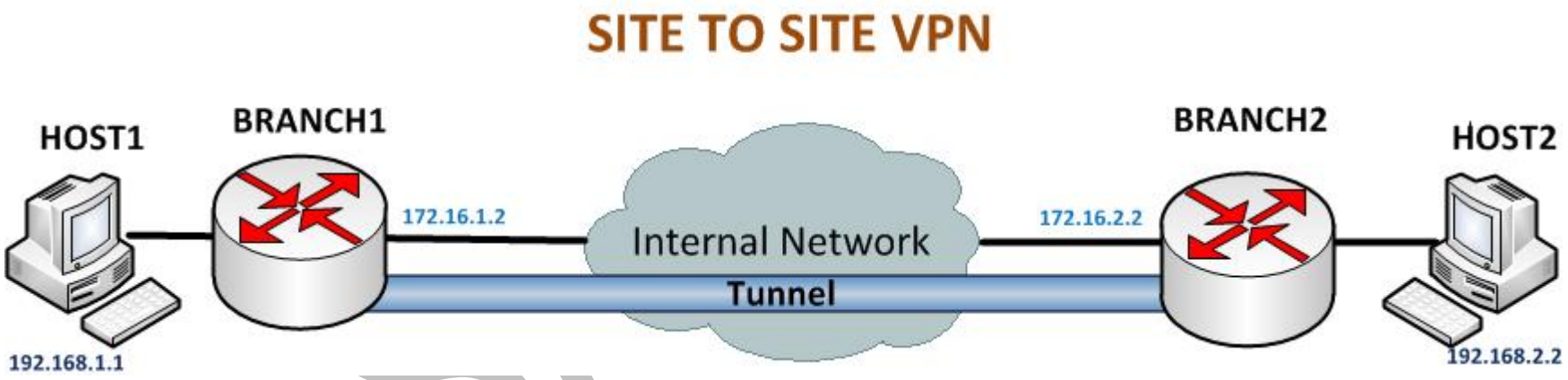
```
LAB#show crypto ipsec transform-set default
{ esp-aes esp-sha-hmac  }
   will negotiate = { Transport,  },
```

1. Site to Site VPN using Smart Defaults

Introduction :

In this scenario, we would setup a Site to Site VPN between 2 Branch Routers i.e. Branch1 & Branch2 using IKEv2 Smart Defaults.**IKEv2 Smart Defaults** feature minimizes the FlexVPN configuration by covering most of the use cases. IKEv2 smart defaults can be customized for specific use cases, though this is not recommended.

Scenario Diagram :



Verification :

In order to verify that tunnel is established, ping from **HOST1 (192.168.1.1)** in LAN1 to **HOST2 (192.168.2.2)** in LAN2. Ping should be successful

Verify using following commands:

```
Branch1#ping 192.168.2.2 so lo0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 132/146/168 ms
Branch1#
```

Verify that IKEv2 session is established between the peers using following command. Pay attention to **highlighted** items. Both peers are using PSK authentication

```
Branch1#show crypto ikev2 session detailed
 IPv4 Crypto IKEv2 Session

Session-id:2, Status:UP-ACTIVE, IKE count:1, CHILD count:1

Tunnel-id Local                 Remote                fvrf/ivrf            Status
2         172.16.1.2/500        172.16.2.2/500        none/none            READY
      Encr: AES-CBC, keysize: 256, Hash: SHA512, DH Grp:5, Auth sign: PSK, Auth verify: PSK
      Life/Active Time: 86400/55 sec
      CE id: 1001, Session-id: 2
      Status Description: Negotiation done
      Local spi: AA948962D756AC67       Remote spi: B2F33FFE802CCFC5
      Local id: 172.16.1.2
      Remote id: 172.16.2.2
      Local req msg id:  0              Remote req msg id:  2
      Local next msg id: 0              Remote next msg id: 2
```

FINAL CONFIGURATION :

Here's the final configuration on Branch1 & Branch2. Make sure you specify **correct peer IP address** under specific Branch configuration

BRANCH1

```
hostname Branch1
!
crypto ikev2 keyring KR
 peer Branch2
  address 172.16.2.2
  pre-shared-key local cisco123
  pre-shared-key remote 123cisco
 !
!
crypto ikev2 profile default
 match identity remote address 172.16.2.2 255.255.255.255
 authentication remote pre-share
 authentication local pre-share
 keyring local KR
!
interface Loopback0
 ip address 192.168.1.1 255.255.255.0
!
interface Tunnel0
 ip address 10.10.0.1 255.255.255.252
 tunnel source FastEthernet0/0
 tunnel destination 172.16.2.2
 tunnel protection ipsec profile default
!
interface FastEthernet0/0
 ip address 172.16.1.2 255.255.255.0
!
```

IKEv2 Constructs :

Let's first discuss IKEv2 constructs before jumping into configuration tasks. Here are the FOUR IKEv2 constructs:

- **IKEv2 Proposal**
- **IKEv2 Policy**
- **IKEv2 Profile**
- **IKEv2 Key Ring**

Let's dig them in detail:

- **IKEv2 Proposal:**

An **Internet Key Exchange Version 2 (IKEv2)** proposal is a collection of transforms used in the negotiation of Internet Key Exchange (IKE) security associations (SAs) as part of the **IKE_SA_INIT** exchange. The transform types used in the negotiation are as follows:

- Encryption algorithm
- Integrity algorithm
- Pseudo-Random Function (PRF) algorithm
- Diffie-Hellman (DH) group

- **IKEv2 Policy:**

An **IKEv2 policy** contains **IKEv2 proposals** that are used to negotiate the encryption, integrity, PRF algorithms, and DH group in the **IKE_SA_INIT** exchange. It can have match statements, which are used as selection criteria to select a policy during negotiation.

- **IKEv2 Profile**

An **IKEv2 profile** is a repository of nonnegotiable parameters of the IKE SA, such as **local or remote identities and authentication methods** and services that are available to authenticated peers that match the profile. ***An IKEv2 profile must be attached to either a crypto map or an IPSec profile on the initiator***. An IKEv2 profile is not mandatory on the responder.

Configuration :

Here is how we will configure the devices :

- **Lets first configure IKEv2 Proposal**

First IKEv2 construct is IKEv2 Proposal . Specify one or more Encryption Method, Integrity Algorithm & Diffie-Hellman group. Method specified on one peer **MUST** match the remote peer for IKEv2 session to be established.

```
Branch1(config)#crypto ikev2 proposal IKEv2-PROP
IKEv2 proposal MUST have atleast an encryption algorithm, an integrity algorithm and a dh group configured
Branch1(config-IKEv2-PROPosal)#encryption aes-cbc-256 aes-cbc-128 3des
Branch1(config-IKEv2-PROPosal)#integrity sha384 sha256 sha1
Branch1(config-IKEv2-PROPosal)#group 14 5
```

- **Next configure IKEv2 Policy**

Specify the IKEv2 Proposal under IKEv2 Policy

```
Branch1(config)#crypto ikev2 policy IKEv2-POL
IKEv2 policy MUST have atleast one complete proposal attached
Branch1(config-IKEv2-POL)#proposal IKEv2-PROP
Branch1(config-IKEv2-POL)#!
```

- **Configure IKEv2 Keyring**

Specify local & remote authentication keys for remote peer.

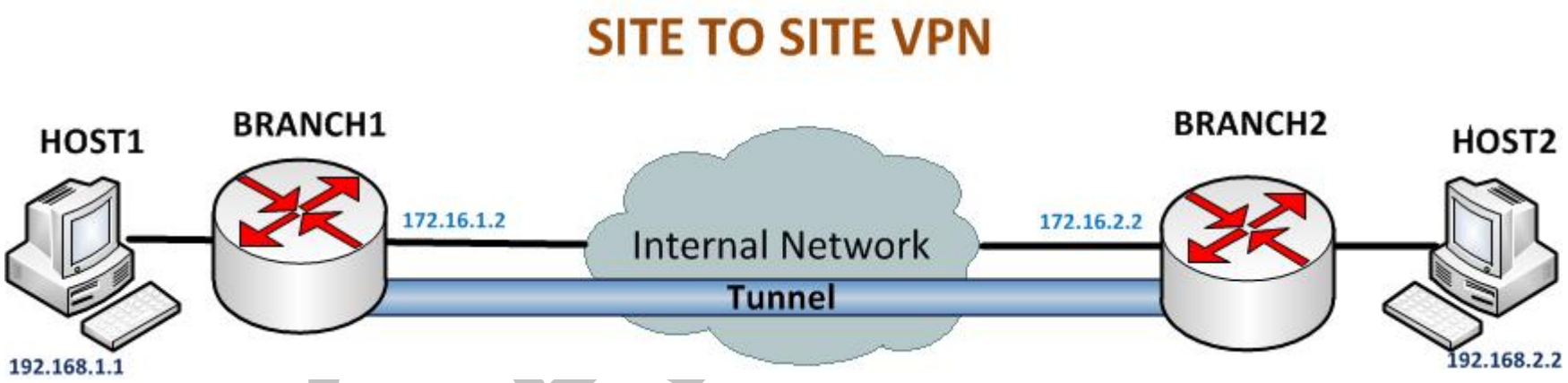
```
Branch1(config)#crypto ikev2 keyring KR
Branch1(config-ikev2-keyring)#peer Branch2
Branch1(config-ikev2-keyring-peer)#address 172.16.2.2
Branch1(config-ikev2-keyring-peer)#pre-shared-key local cisco123
```

3. Site to Site VPN using IPv6 routing

Introduction :

In this scenario, we would setup a Site to Site VPN between 2 Branch Routers i.e. Branch1 & Branch2 using IKEv2 Custom CLI constructs.

Scenario Diagram :



Verification :

In order to verify that tunnel is established, ping from **BRANCH1** (2001:DB8:CAFE::1/64) to **BRANCH2** (2001:DB8:BEEF::1) to ensure IPv6 (OSPFv3) routes are being exchanged. Ping should be successful

Verify IPv6 routes

```
Branch1#show ipv6 route
IPv6 Routing Table - default - 4 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
       NDr - Redirect, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, l - LISP
O   2001:DB8:BEEF::/64 [110/1001]
     via FE80::2, Tunnel0
C   2001:DB8:CAFE::/64 [0/0]
     via FastEthernet0/0, directly connected
L   2001:DB8:CAFE::1/128 [0/0]
     via FastEthernet0/0, receive
L   FF00::/8 [0/0]
     via Null0, receive
```

Ping Link-Local addresses

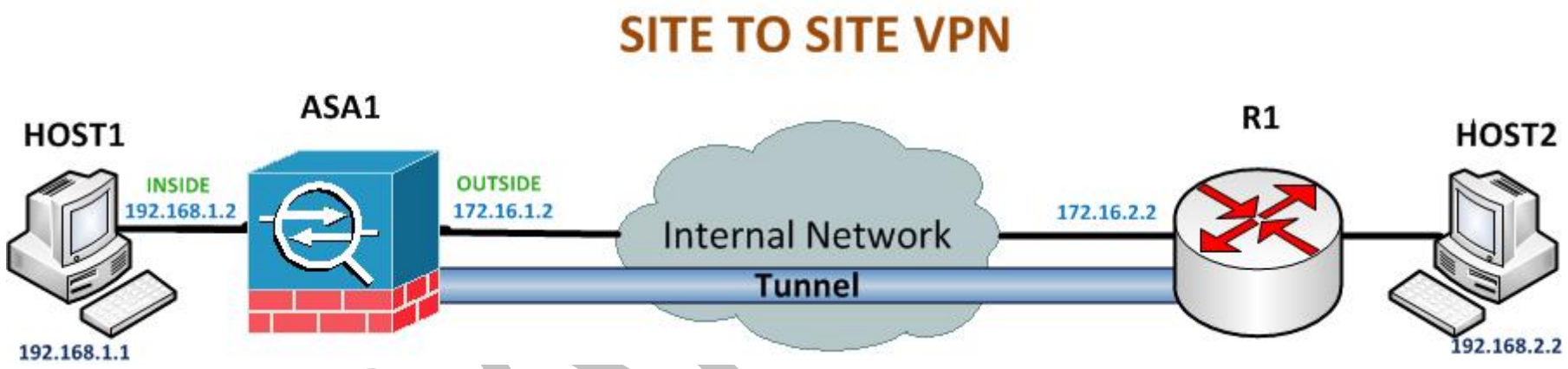
```
Branch1#ping FE80::2
Output Interface: Tunnel0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::2, timeout is 2 seconds:
Packet sent with a source address of FE80::1%Tunnel0
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 108/119/148 ms
```

5. Site to Site VPN (ASA – IOS) using Static VTI

Introduction :

In this scenario, we would setup a Site to Site VPN between 2 different devices i.e. an ASA1 firewall appliance & R1 (Cisco Router) using IKEv2 Constructs & Static Crypto Maps on ASA1 & Static Virtual Tunnel Interface on R1

Scenario Diagram :



❖ **Specify Connection Profile (Tunnel Group)**

IPSec Session will land on the specified tunnel group. Make sure you specify type as **'ipsec-l2l'**

```
ASA1(config)#tunnel-group 172.16.2.2 type ipsec-l2l
ASA1(config)#tunnel-group 172.16.2.2 ipsec-attributes
ASA1(config-tunnel-ipsec)#ikev2 remote-authentication pre-shared-key cisco123
INFO: You must configure ikev2 local-authentication pre-shared-key
      or certificate to complete authentication.
ASA1(config-tunnel-ipsec)#ikev2 local-authentication pre-shared-key cisco123
ASA1(config-tunnel-ipsec)#!
```

❖ **Specify & Enable Crypto Map on ASA1 Outside Interface**

Crypto MAPs will reference interesting ACL traffic, Peer & IPSEC Proposal

```
ASA1(config)#crypto map CMAP 10 match address 100
ASA1(config)#crypto map CMAP 10 set peer 172.16.2.2
ASA1(config)#crypto map CMAP 10 set ikev2 ipsec-proposal ike_prop1
ASA1(config)#crypto map CMAP interface outside
```

Here is the ASA1 interface IP addressing & configuration

```
ASA1# show interface ip brief
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0           172.16.1.2      YES manual up                    up
GigabitEthernet1           192.168.1.2     YES manual up                    up


ASA1#show run interface gigabitEthernet 0
interface GigabitEthernet0
 nameif outside
 security-level 0
 ip address 172.16.1.2 255.255.255.0

ASA1#show run interface gigabitEthernet 1
```

❖ **Configure IKEv2 Policy**

Specify the IKEv2 Proposal under IKEv2 Policy

```
R1(config)#crypto ikev2 policy IKEv2-POL
IKEv2 policy MUST have atleast one complete proposal attached
R1(config-IKEv2-POL)#proposal IKEv2-PROP
R1(config-IKEv2-POL)#!
```

❖ **Configure IKEv2 Keyring**

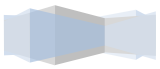
Specify local & remote authentication keys for remote peer.

```
R1(config)#crypto ikev2 keyring KR
R1(config-ikev2-keyring)#peer ASA1
R1(config-ikev2-keyring-peer)#address 172.16.1.2
R1(config-ikev2-keyring-peer)#pre-shared-key local cisco123
R1(config-ikev2-keyring-peer)#pre-shared-key remote cisco123
R1(config-ikev2-keyring-peer)#!
```

❖ **Configure IKEv2 Profile**

Specify authentication methods & keyring under IKEv2 Profile.

```
R1(config)#crypto ikev2 profile IKEv2-PROF-ASA1
IKEv2 profile MUST have:
   1. A local and a remote authentication method.
   2. A match identity or a match certificate statement.
R1(config-ikev2-profile)#match identity remote address 172.16.1.2 255.255.255.255
R1(config-ikev2-profile)#authentication remote pre-share
R1(config-ikev2-profile)#authentication local pre-share
R1(config-ikev2-profile)#keyring local KR
R1(config-ikev2-profile)#!
```

```
IOS-CA(cs-server)#
*Dec 23 13:44:01.783: %PKI-6-CS_ENABLED: Certificate server now enabled.
```

Now, once CA is setup, verify its status is enabled.

```
IOS-CA#show crypto pki server IOS-CA
Certificate Server IOS-CA:
    Status: enabled
    State: enabled
    Server's configuration is locked  (enter "shut" to unlock it)
    Issuer name: CN=IOS-CA.cisco.com,OU=VPN,O=CISCO
    CA cert fingerprint: EB0616D1 1571166F 9525CD34 FDA0CA39
    Granting mode is: auto
    Last certificate issued serial number (hex): 1
    CA certificate expiration timer: 13:44:01 UTC Dec 23 2015
    CRL NextUpdate timer: 19:44:01 UTC Dec 23 2012
    Current primary storage dir: nvram:
    Database Level: Minimum - no cert data written to storage
```

Next, Configure Branch routers with trustpoint configuration

```
Branch1(config)#crypto pki trustpoint IOS-CA
Branch1(ca-trustpoint)#enrollment url http://3.3.3.3:80
Branch1(ca-trustpoint)#subject-name CN=Branch1.cisco.com,OU=VPN,O=CISCO
Branch1(ca-trustpoint)#revocation-check none
```

Authenticate & Enroll to CA Server

```
Branch1(config)#crypto pki authenticate IOS-CA
Certificate has the following attributes:
       Fingerprint MD5: EB0616D1 1571166F 9525CD34 FDA0CA39
      Fingerprint SHA1: 0EA1F879 4CDE55FB DB1193EA 644152A7 00111855
```

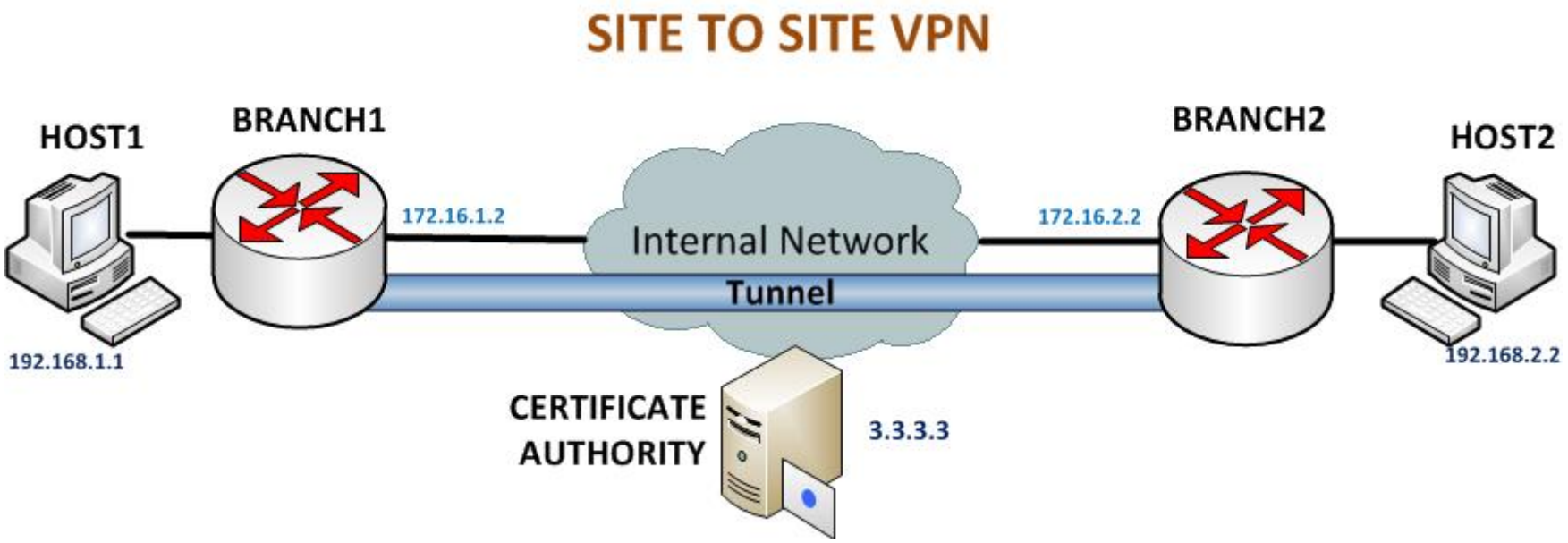
```
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.

Branch1(config)#crypto key generate rsa general-keys label Branch1.cisco.com
The name for the keys will be: Branch1.cisco.com
Choose the size of the key modulus in the range of 360 to 4096 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 768
% Generating 768 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 0 seconds)

Branch1(config)#
00:17:19: %SSH-5-ENABLED: SSH 1.99 has been enabled

Branch1(config)#crypto pki enroll IOS-CA
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
   password to the CA Administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the configuration.
   Please make a note of it.

Password: cisco123
Re-enter password: cisco123

% The subject name in the certificate will include: CN=Branch1.cisco.com,OU=VPN,O=CISCO
% The subject name in the certificate will include: Branch1.cisco.com
% Include the router serial number in the subject name? [yes/no]: no
% Include an IP address in the subject name? [no]: no
Request certificate from CA? [yes/no]: yes
% Certificate request sent to Certificate Authority
% The 'show crypto pki certificate verbose IOS-CA' command will show the fingerprint.
```

8. Site to Site VPN using RSA-Signatures

Introduction :

In this scenario, we would setup a Site to Site VPN between 2 Branch Routers i.e. Branch1 & Branch2 using custom IKEv2 Configuration & Certificate-based authentication. We will be configuring **IKEv2 constructs** as well as utilizing **IKEv2 Smart Defaults.** Additionally, we would be configuring **Branch1 using Virtual-Access** & **Branch2 using Static Tunnel Interface**. Normally, this method is used in HUB/SPOKE VPN setup but here it can be used for Site to Site VPN as well.

Scenario Diagram :



Verification :

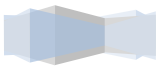
In order to verify that tunnel is established, ping from **HOST1 (192.168.1.1)** in LAN1 to **HOST2 (192.168.2.2)** in LAN2. Ping should be successful

In this case, we are using **Smart Defaults for IKEv2 Proposal & IPSec Transform Set**

```
Branch1#show crypto ikev2 proposal
 IKEv2 proposal: default
      Encryption : AES-CBC-256 AES-CBC-192 AES-CBC-128
      Integrity  : SHA512 SHA384 SHA256 SHA96 MD596
      PRF        : SHA512 SHA384 SHA256 SHA1 MD5
      DH Group   : DH_GROUP_1536_MODP/Group 5 DH_GROUP_1024_MODP/Group 2

Branch1#show crypto ipsec transform-set default
{ esp-aes esp-sha-hmac  }
   will negotiate = { Transport,  },
```

Verify Tunnel is established. This can be seen by Virtual-Access interface up on Branch1.

```
Branch1#show ip int b
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        172.16.1.2      YES NVRAM  up                    up
FastEthernet0/1        unassigned      YES NVRAM  administratively down down
Loopback0              192.168.1.1     YES NVRAM  up                    up
Loopback1              10.10.0.1       YES NVRAM  up                    up
Virtual-Access1        10.10.0.1       YES unset  up                    up
Virtual-Template1      10.10.0.1       YES unset  up                    down
Branch1#
```

12.Advanced HUB & SPOKE (IKEv2 Routing with RADIUS Authorization)

- Certificate on HUB, PSK on SPOKES
- Virtual-Access Interface : HUB
- Static Tunnel Interface : SPOKE1/SPOKE2
- Config Exchange / IKEv2 Routing

Introduction :

In this scenario, we would setup a HUB & SPOKE VPN i.e. Branch1 & Branch2 will be SPOKES establishing tunnel with with the HUB1. We will be using Asymmetric authentication here i.e. HUB1 will be using Certificate for LOCAL authentication while both SPOKES will be using Pre-Shared Key. Here, we will introduce concepts of Configuration Exchange & **IKEv2 Routing with RADIUS Authorization** . We will use **Cisco Secure ACS 5.4** as RADIUS Server for this demonstration.

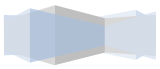
Scenario Diagram :

Please refer to next page

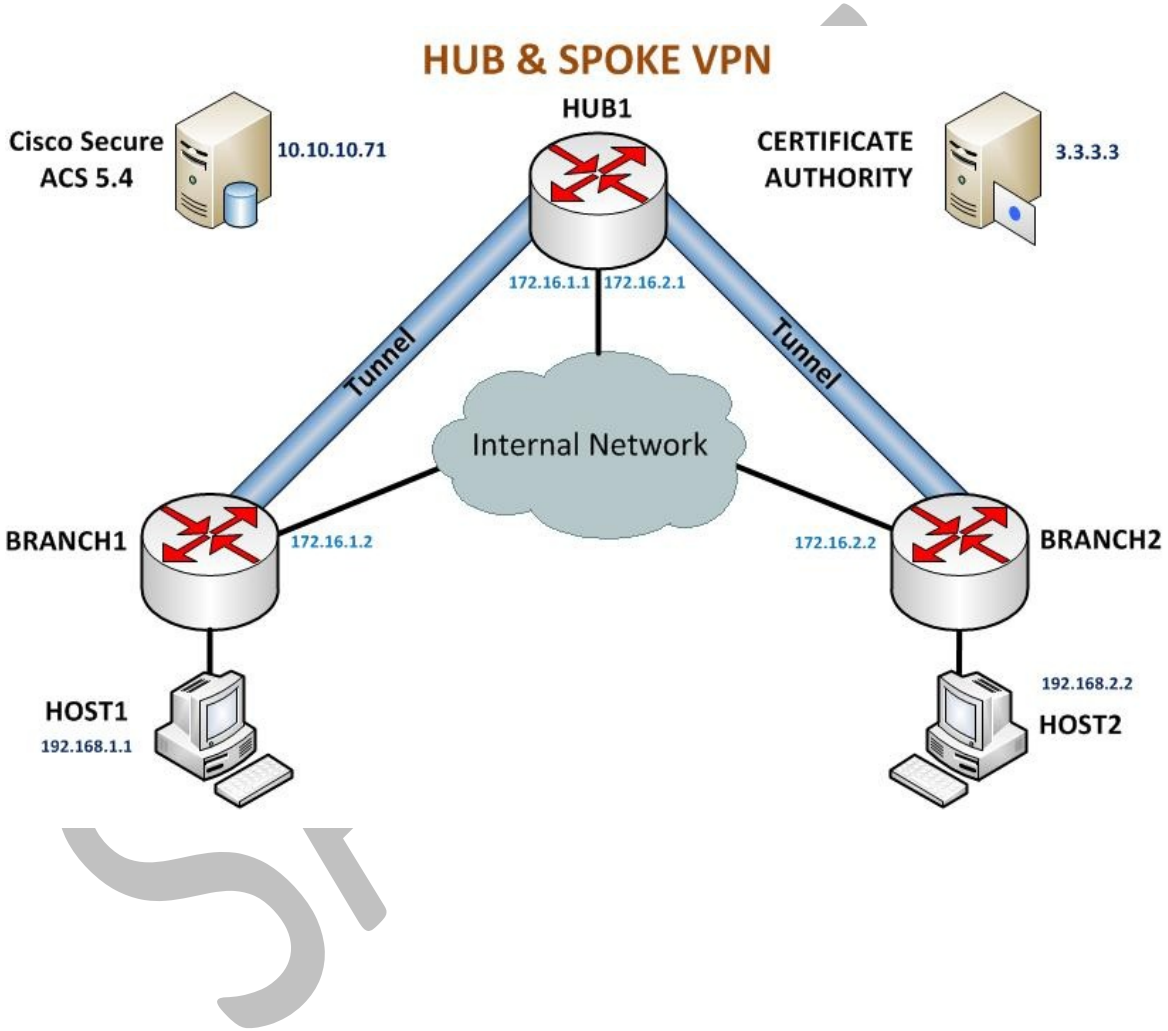
Components :

Here is the list of components used in this LAB scenario

- **Cisco Secure ACS 5.4 : 10.10.10.71**
- **IOS Certificate Authority : 3.3.3.3**



Scenario Diagram :



HUB1 Configuration :

Here is how we will configure the devices :

- **Configure AAA for RADIUS Authorization**

Here,we are specifying authorization via **Cisco Secure ACS5X Server** (using RADIUS as an authentication protocol). We could very well use **Cisco Identity Services Engine (ISE)** for authorization as well. Here, we will be performing AAA-based Pre-shared-key & RADIUS authorization

```
HUB1(config)#aaa new-model
HUB1(config)#aaa group server radius ACS54
HUB1(config-sg-radius)#server-private 10.10.10.71 key cisco123
HUB1(config-aaa)#!
HUB1(config)#aaa authorization network ACS group ACS54
HUB1(config)#!
```

- **Configure IKEv2 Name-Mangler**

```
HUB1(config)#crypto ikev2 name-mangler EXTRACT-HOSTNAME
HUB1(config-ikev2-name-mangler)#fqdn hostname
HUB1(config)#crypto ikev2 name-mangler EXTRACT-DOMAIN-NAME
HUB1(config-ikev2-name-mangler)#fqdn domain
```

- **Configure Additional Paramters**

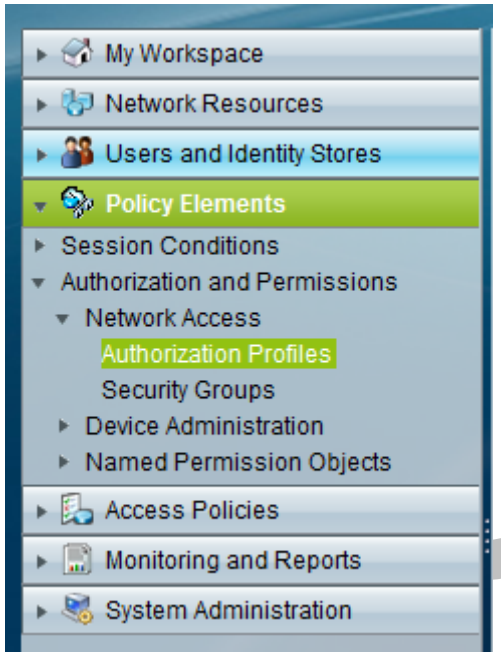
Specify IP pool from which Spokes will be assigned ip addresses dynamically & access-list will specify the subnets to which spokes will have access to. In this case, SPOKES will be able to connect to subnets behind HUB1 **(192.168.10.0)**

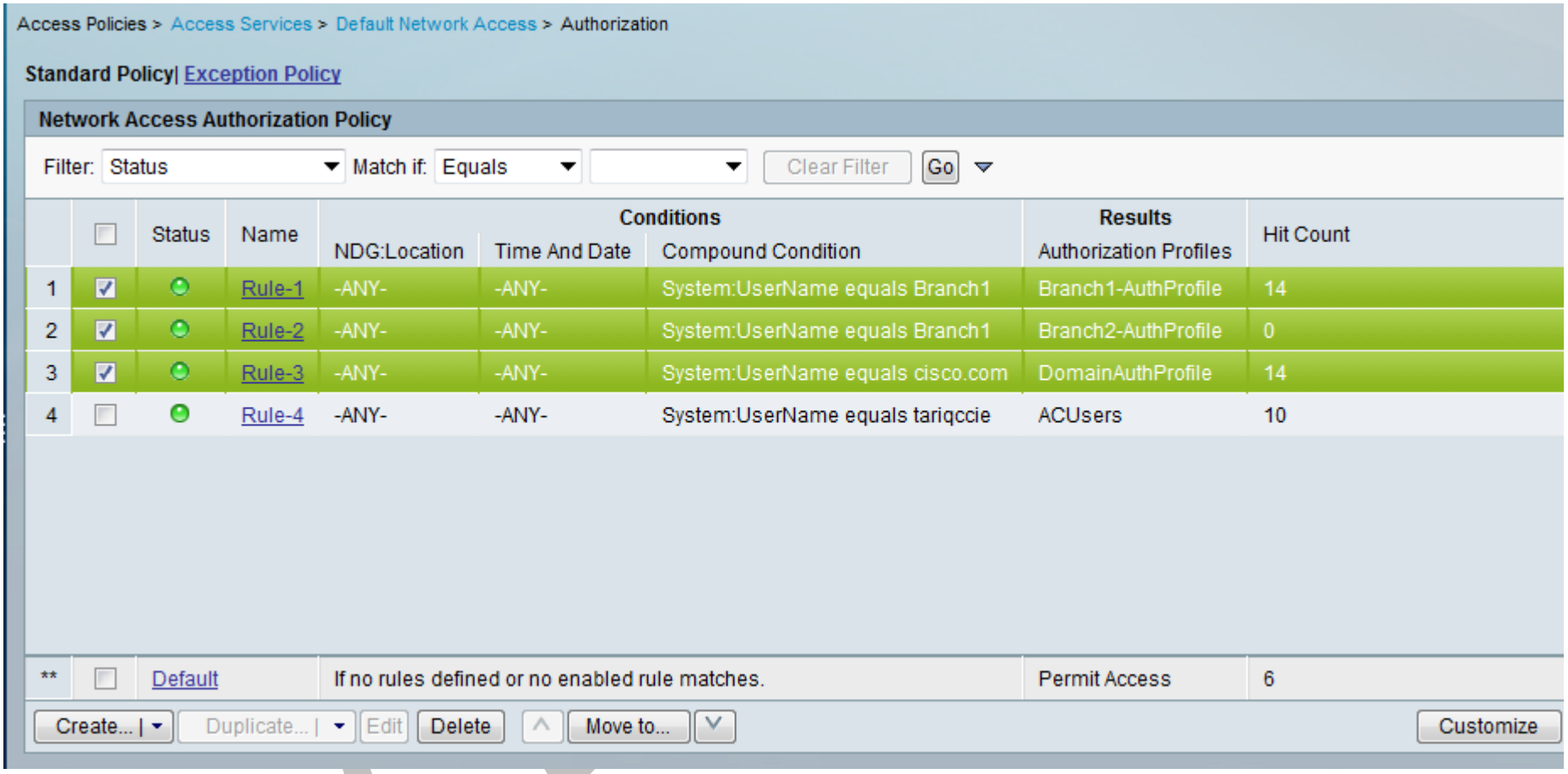
```
HUB1(config)#access-list 99 permit 192.168.10.0 0.0.0.255
HUB1(config)#ip local pool mypool 10.10.0.100 10.10.0.110
HUB1(config)#!
```

❖ **Create Authorization Profile(s) in ACS**

```
You will need to create THREE (3) Authorization Profiles which will be applied to respective users
     1st : Branch1AuthProfile      [Cisco-AV-Pair: ikev2-password-remote=cisco123]
     2nd : Branch2AuthProfile      [Cisco-AV-Pair: ikev2-password-remote=cisco123]
     3rd : DomainAuthProfile       [Cisco-AV-Pair: ipsec:addr-pool=mypool]
                                   [Cisco-AV-Pair: ipsec:route-accept=any]
                                   [Cisco-AV-Pair: ipsec:route-set=interface]
                                   [Cisco-AV-Pair: ipsec:route-set=access-list99]
```

Navigate to *Policy Elements -> Authorization and Permissions -> Network Access -> Authorization Profiles*





Since we are using AAA based authorization, ensure that domain-name extracted via name-mangler is **'Branch1'** & default password for this configured on RADIUS SERVER has to be **'cisco'** which is hard-coded into IOS for radius network authorization. All parameters retrieved from ACS Server will be applied to the remote clients.

```
HUB1#test aaa group ACS54 cisco.com cisco new-code
User successfully authenticated

USER ATTRIBUTES

username                0   "cisco.com"
addr-pool               0   "mypool"
route-accept            0   "any"
route-set               0   "interface"
route-set               0   "access-list99"
```

'Shut/No Shut' Tunnel0 interface on Branch1 & let's see what output we get on HUB1. You will see that RADIUS Server will send above listed attributes to HUB1 in case of successful authentication/authorization.

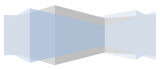
Pay attention to the 'debug' output listed below & especially to the **highlighted** items.

```
HUB1#debug radius
Radius protocol debugging is on
Radius protocol brief debugging is off
Radius protocol verbose debugging is off
Radius packet hex dump debugging is off
Radius packet protocol debugging is on
Radius elog debugging debugging is off
Radius packet retransmission debugging is off
Radius server fail-over debugging is off
Radius elog debugging debugging is off
HUB1#
*Dec 25 12:13:08.359: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state to down
HUB1#
RADIUS/ENCODE(0000001E):Orig. component type = VPN IPSEC
```

```
RADIUS:  Calling-Station-Id  [31]  12  "172.16.1.2"
RADIUS:  Service-Type        [6]   6   Outbound                 [5]
RADIUS:  NAS-IP-Address      [4]   6   10.10.10.99
RADIUS(0000001F): Started 5 sec timeout
RADIUS: Received from id 1645/13 10.10.10.71:1645, Access-Accept, len 185
RADIUS:  authenticator 49 26 C3 D2 F2 EF 32 CE - FB 7A 6B C8 2E 93 7B 09
RADIUS:  User-Name           [1]   11  "cisco.com"
RADIUS:  Class               [25]  24
RADIUS:   43 41 43 53 3A 41 43 53 35 2F 31 34 35 37 39 31  [CACS:ACS5/145791]
RADIUS:   37 35 39 2F 31 33                [ 759/13]
RADIUS:  Vendor, Cisco       [26]  30
RADIUS:   Cisco AVpair       [1]   24  "ipsec:addr-pool=mypool"
RADIUS:  Vendor, Cisco       [26]  30
RADIUS:   Cisco AVpair       [1]   24  "ipsec:route-accept=any"
RADIUS:  Vendor, Cisco       [26]  33
RADIUS:   Cisco AVpair       [1]   27  "ipsec:route-set=interface"
RADIUS:  Vendor, Cisco       [26]  37
RADIUS:   Cisco AVpair       [1]   31  "ipsec:route-set=access-list99"
RADIUS(0000001F): Received from id 1645/13
*Dec 25 12:13:12.967: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state to down
*Dec 25 12:13:13.491: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state to up
HUB1#
```

Check IP routing table on HUB1 and ensure that routes got installed via IKEv2 routing for Branch1(Spoke1)

```
HUB1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
```

Branch1 Configuration :

- **Configure AAA for Local Authorization**

Here,we are specifying authorization via LOCAL database. We could very well use RADIUS Server for authorization i.e. **Cisco Secure ACS** or **Cisco Identity Services Engine (ISE)**

```
Branch1(config)#aaa new-model
Branch1(config)#aaa authorization network default local
```

- **Configure IKEv2 Key Ring**

Specify authentication keys (PSK) for peers (Spokes).

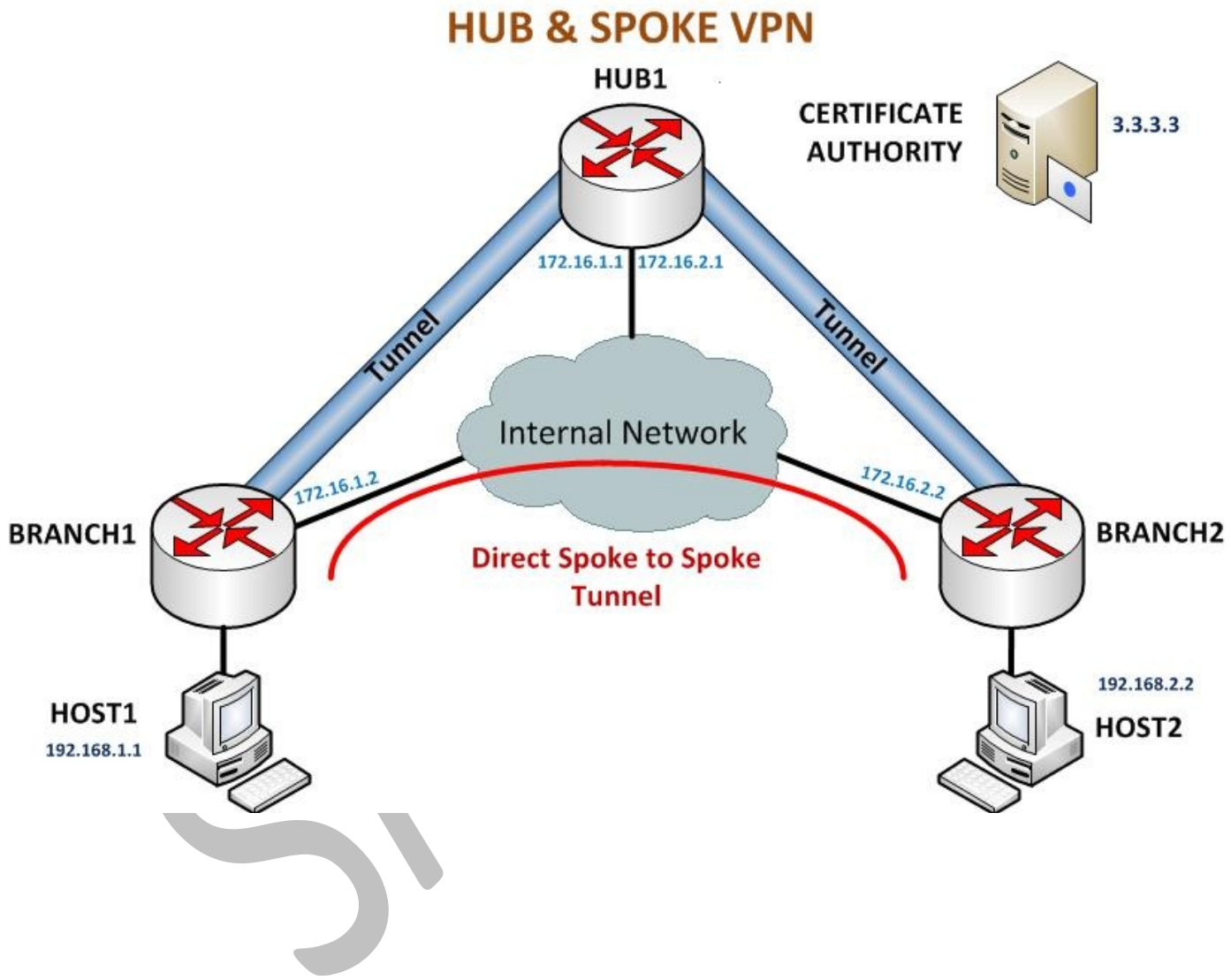
```
Branch1(config)#crypto ikev2 keyring ALL
Branch1(config-ikev2-keyring)#peer all
Branch1(config-ikev2-keyring-peer)#address 0.0.0.0 0.0.0.0
Branch1(config-ikev2-keyring-peer)#pre-shared-key cisco123
```

- **Configure IKEv2 Authorization Policy**

Specify access-list which will specify the subnets to which other peers or HUB will have access to. In this case, PEERS will be able to connect to subnets behind SPOKE1 **(192.168.1.0)**

```
Branch1(config)#access-list 99 permit 192.168.1.0 0.0.0.255
Branch1(config)#crypto ikev2 authorization policy default
%Warning: This will Modify Default IKEv2 Authorization Policy. Exit if you don't want
Branch1(config-ikev2-author-policy)#route set access-list 99
Branch1(config-ikev2-author-policy)#!
```

- **Configure IKEv2 Profile**



- **Configure a Virtual-Template interface for NHRP redirect**

Here we need to configure **"ip nhrp redirect"** which informs to the spoke that it can communicate to other intended spoke directly.

```
HUB1(config)#interface Loopback1
HUB1(config-if)#ip address 10.10.0.1 255.255.255.0

HUB1(config)#interface Virtual-Template1 type tunnel
HUB1(config-if)#ip unnumbered Loopback1
HUB1(config-if)#ip nhrp network-id 1
HUB1(config-if)#ip nhrp redirect
HUB1(config-if)#tunnel protection ipsec profile default
```

Branch1 Configuration :

- **Configure AAA for Local Authorization**

Here,we are specifying authorization via LOCAL database.

```
Branch1(config)#aaa new-model
Branch1(config)#aaa authorization network default local
```

- **Configure Certificate MAP (PKI)**

Specify to match Organizational Unit (OU)

```
Branch1(config)#crypto pki certificate map CERTMAP 10
Branch1(ca-certificate-map)#subject-name co ou = VPN
```

- **Configure IKEv2 Authorization Policy**
- **Configure a Virtual-Template interface for NHRP shortcut**

Here, we need to configure "**ip nhrp shortcut**" which is responsible to rewrite the CEF entry after getting the redirect message from HUB1.

```
Branch1(config)#interface Virtual-Template1 type tunnel
Branch1(config-if)#ip unnumbered Tunnel0
Branch1(config-if)#ip nhrp network-id 1
Branch1(config-if)#ip nhrp shortcut virtual-template 1
Branch1(config-if)#ip nhrp holdtime 300
Branch1(config-if)#tunnel protection ipsec profile default
00:56:20: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Template1, changed state to down!
00:56:22: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
```

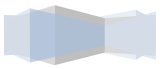
- **Configure Tunnel Interface**

Specify tunnel source/destination& ipsec profile under Tunnel0 interface.Notice we are using 'default' IPSec profile in here.

NOTICE : Tunnel0 is being assigned IP address dynamically (via Config Exchange).

Configure NHRP shortcut under Tunnel0 & bounce the interface

```
Branch1(config)#interface Tunnel0
Branch1(config-if)#ip address negotiated
Branch1(config-if)#ip nhrp network-id 1
Branch1(config-if)#ip nhrp holdtime 300
Branch1(config-if)#ip nhrp shortcut virtual-template 1
Branch1(config-if)#tunnel source FastEthernet0/0
Branch1(config-if)#tunnel destination 172.16.1.1
Branch1(config-if)#tunnel protection ipsec profile default
Branch1(config-if)#shut
Branch1(config-if)#no shut
01:02:00: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to up
01:02:00: %LINK-3-UPDOWN: Interface Tunnel0, changed state to up
```

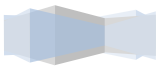
FLEX VPN RESILIENCY

Dual FlexVPN Support:

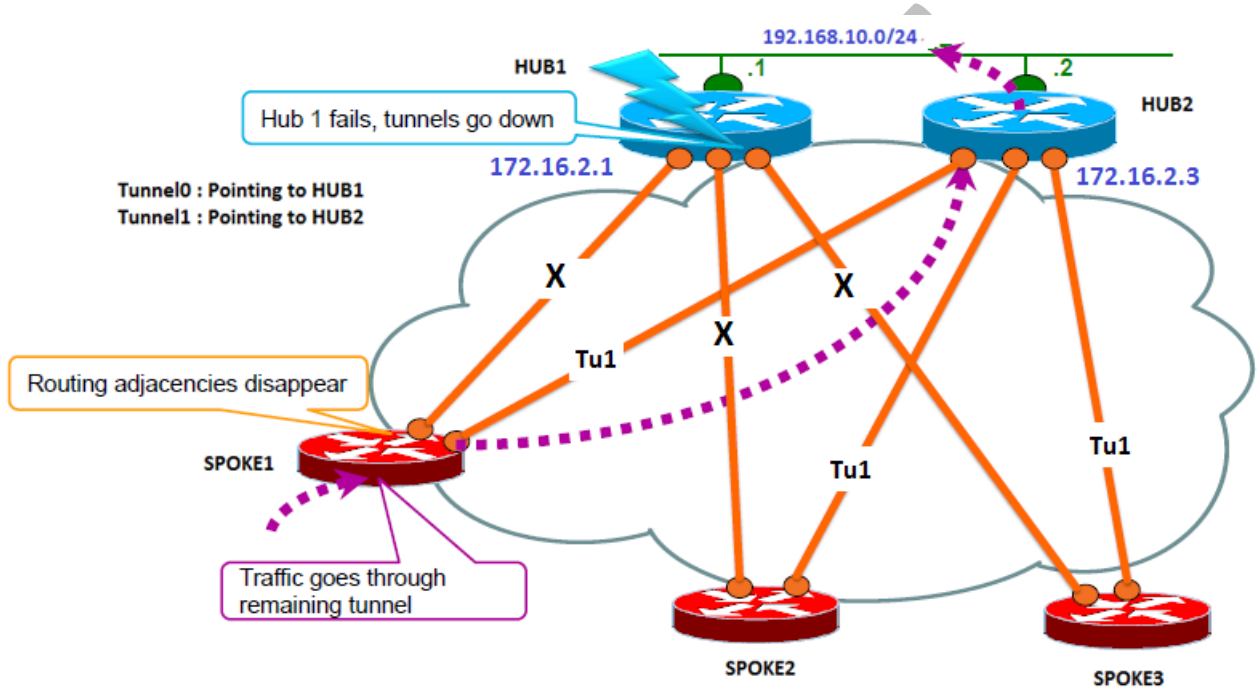
- The Dual FlexVPN Support feature provides the ability to configure two FlexVPN tunnels that share the same inside and outside interfaces.
- The two FlexVPN tunnels use route injections to direct appropriate traffic through the corresponding tunnel interface.
- When the tunnel is up, the tunnel "learns" the network list from the server.
- If the server forwards a network list, FlexVPN installs specific routes to the destination networks in its routing table, directing the traffic to these networks out of the tunnel interface.

Example (on a SPOKE/BRANCH site):

```
interface Tunnel0
 ip address negotiated
 tunnel source FastEthernet0/0
 tunnel destination 172.16.1.2
 tunnel protection ipsec profile default
end
!
interface Tunnel1
 ip address negotiated
 tunnel source FastEthernet0/0
 tunnel destination 172.16.1.3
 tunnel protection ipsec profile default
!
```

Primary HUB goes down, traffic flowing via Secondary HUB



IKEv2 BACKUP PEERS

Backup Features :

A FlexVPN client/spoke can connect to various peers or servers in a predetermined order. The list of peers is called the **gateway list** or **backup gateway list** and is built using the following lists:

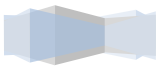
- ***Static backup gateway list or static list***
- Downloaded backup gateway list or downloaded list

In this scenario,we are using a **Static backup gateway**.

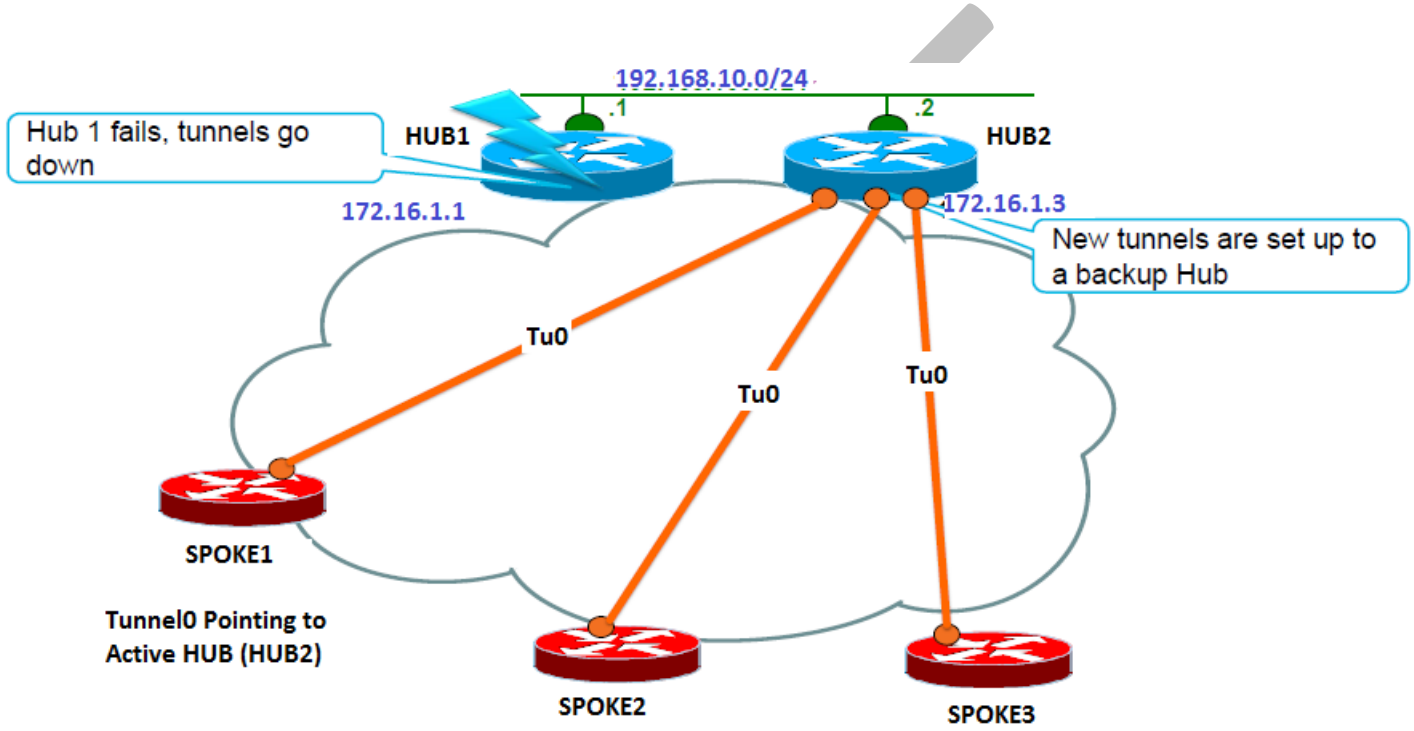
Here are important points to remember :

- The **static backup gateway list** is configured in the FlexVPN profile by providing a list of peers with a sequence number.
- If an existing connection with a peer from the gateway list goes down, the client tries to establish a connection with the next peer in the gateway list.
- Use the **peer** command to add a peer to the backup gateway list.
- Peers are ordered by preference; the lower the sequence number, the higher the preference.
- You can configure a static peer and attach it to a track object. A peer is a **"possible peer"** if the track object of the peer is in the UP state.
- Refer to the following configuration snippet for illustration:

```
crypto ikev2 client flexvpn default
 peer 1 172.16.1.1
 peer 2 172.16.1.3
 client connect Tunnel0
!
```

Tunnel Established to Backup HUB (when PRIMARY HUB fails)



Extensible Authentication Protocol (EAP)

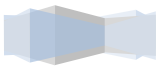
Authenticating Peers using EAP :

The FlexVPN server supports Peer authentication using **Extensible Authentication protocol (EAP)** and acts as a pass-through authenticator relaying EAP messages between the client and the backend EAP server.The backend EAP server is typically a RADIUS server **(Cisco Secure ACS 5X or Cisco Identity Services Engine or FreeRADIUS Server)** that supports EAP authentication.

While a FlexVPN client authenticates the FlexVPN client using EAP, the FlexVPN server must authenticate the FlexVPN server by using certificates.

EAP Features List:

- No X-AUTH in IKEv2; EAP instead
- EAP – authentication framework that provides common functions for various methods:
  - **Tunneling** - EAP-TLS, EAP/PSK, EAP-PEAP etc
  - **Non-tunneling** – EAP-MSCHAPv2, EAP-GTC, EAP-MD5 etc
- Implemented as additional IKE_AUTH exchanges
- Only used to authenticate the initiator to responder (**EAP as initiator**)
- Responder MUST use Certificate



Cisco IKEv2 AnyConnect Client

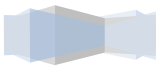
For certificate-based authentication, the FlexVPN server and the AnyConnect client certificates must have an Extended Key Usage (EKU) field as follows:

- ***For the client certificate, EKU field = client authentication certificate***
- ***For the server certificate, EKU field = server authentication certificate***

If the FlexVPN server authenticates to AnyConnect using certificates, a SubjectAltName extension is required in the FlexVPN server certificate that contains the server's IP address or FQDN. Additionally, HTTP certified URLs must be disabled on the FlexVPN server using the no crypto ikev2 http-url cert command.

XML tags specific to IKEv2 sessions in the AnyConnect client profile (example for EAP-MD5 authentication):

```
- <PrimaryProtocol>
    IPsec
  - <StandardAuthenticationOnly>
      true
      <AuthMethodDuringIKENegotiation>EAP-MD5</AuthMethodDuringIKENegotiation>
      <IKEIdentity>MyAnyConnect</IKEIdentity>
    </StandardAuthenticationOnly>
  </PrimaryProtocol>
```

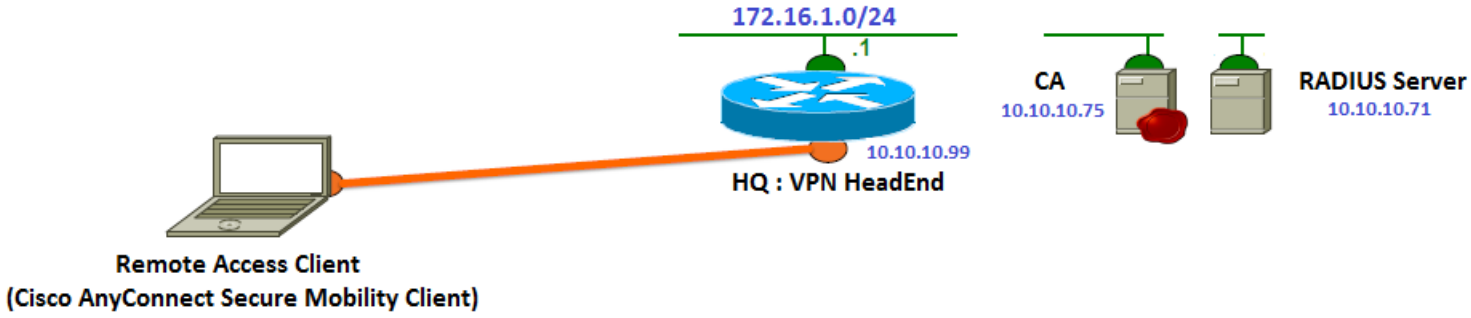
18. FlexVPN Remote Access VPN using EAP Authentication via Cisco Secure ACS 5X

- ❖ EAP-MD5 : Authentication Protocol
- ❖ Cisco AnyConnect Secure Mobility Client v3

Introduction :

In this scenario, we will setup an AnyConnect Client connected to an IOS device using IKEv2 with EAP as an authentication method for Client. Responder or IOS device must use Certificate for authentication. We will perform User Authentication using EAP. AnyConnect Client user will be configured on RADIUS Server (**Cisco Secure ACS 5X**) in this case & authentication and authorization will be performed accordingly. We will use a Windows XP host with **AnyConnect Secure Mobility Client v3** Installed to perform this scenario.

Scenario Diagram :



Certificate Authority :

Certificate Authority setup for this task MUST support EKU (Extended Key Usage). It means you must use relatively new IOS version for this. Here I will be using IOS version 15.2T , however you can also setup OpenSSL or Windows **Certificate Authority (CA)** for this purpose.

Here is how we will configure the IOS CA Server. Pay attention to the **highlighted** items. Our VPN headend must have certificate issed with Server-Auth EKU support.

IOS-CA

```
crypto pki server IOS-CA
 database level complete
 database archive pkcs12 password 7 00071A1507545A545C
 issuer-name cn=IOS-CA.cisco.com,ou=VPN,o=cisco
 grant auto rollover ca-cert
 grant auto
 eku server-auth client-auth
!
    Authority Info Access:
    Extended Key Usage:
        Client Auth
        Server Auth
  Associated Trustpoints: IOS-CA
  Storage: nvram:IOS-CAciscoc#2.cer
  Key Label: HQ.cisco.com

CA Certificate
  Status: Available
  Version: 3
  Certificate Serial Number (hex): 01
  Certificate Usage: Signature
  Issuer:
    cn=IOS-CA.cisco.com
```

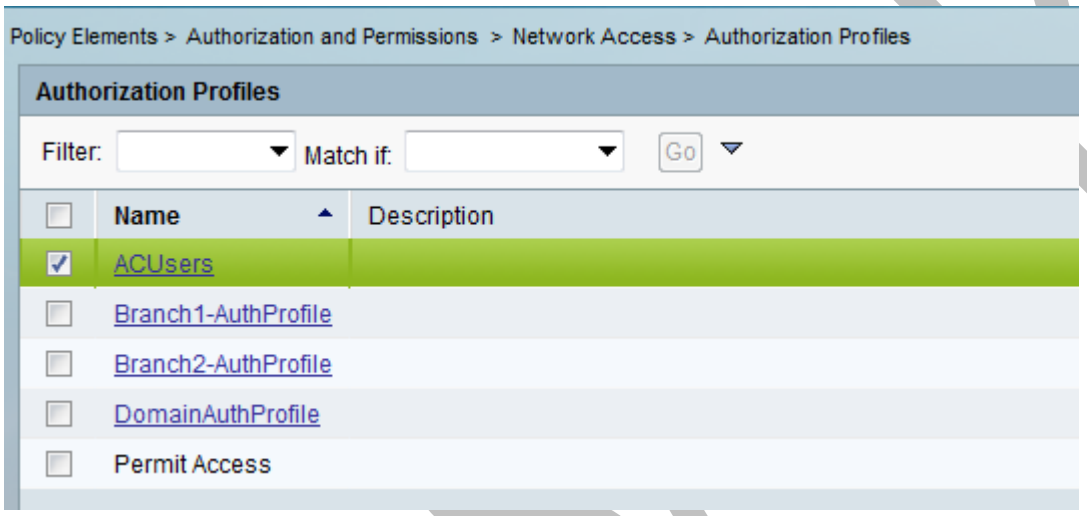
Cisco Secure ACS 5X CONFIGURATION :

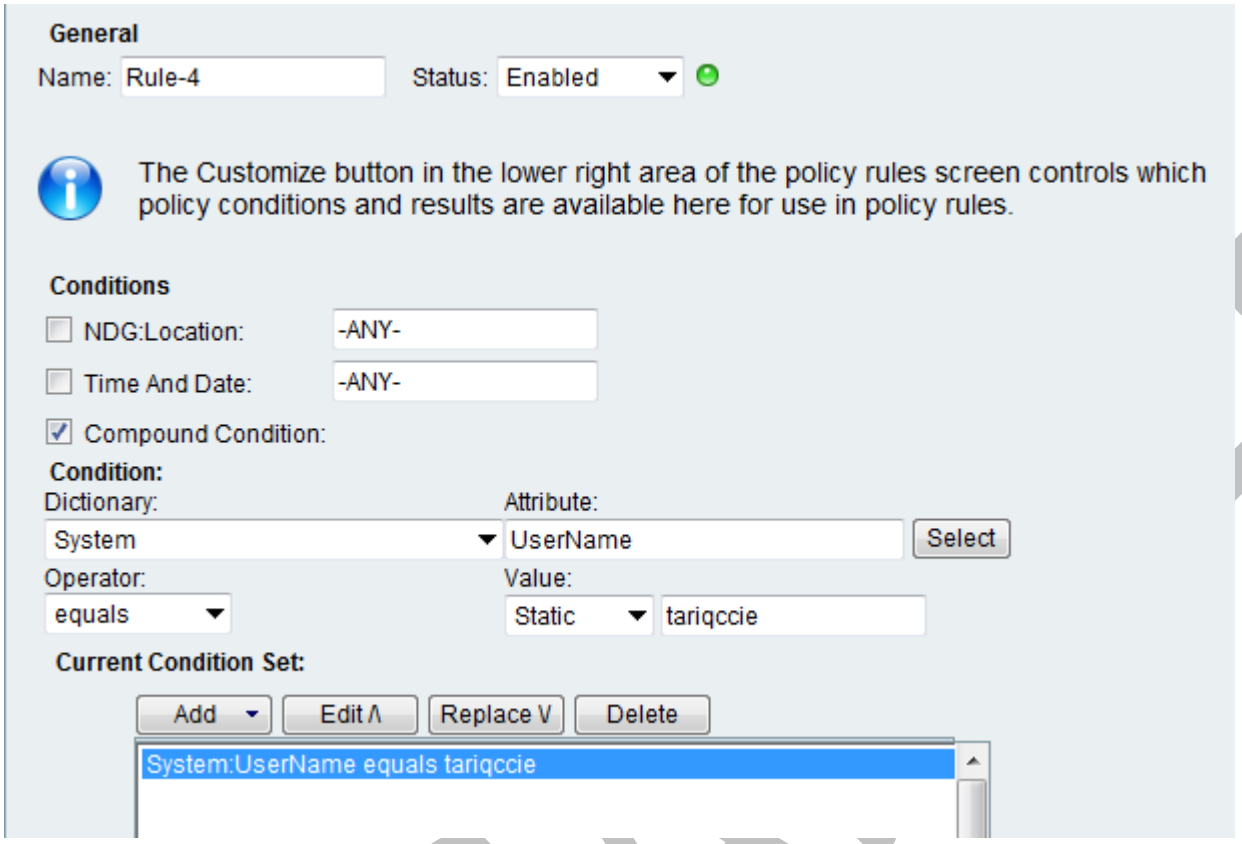
- **Create Authorization Profile(s) in ACS**

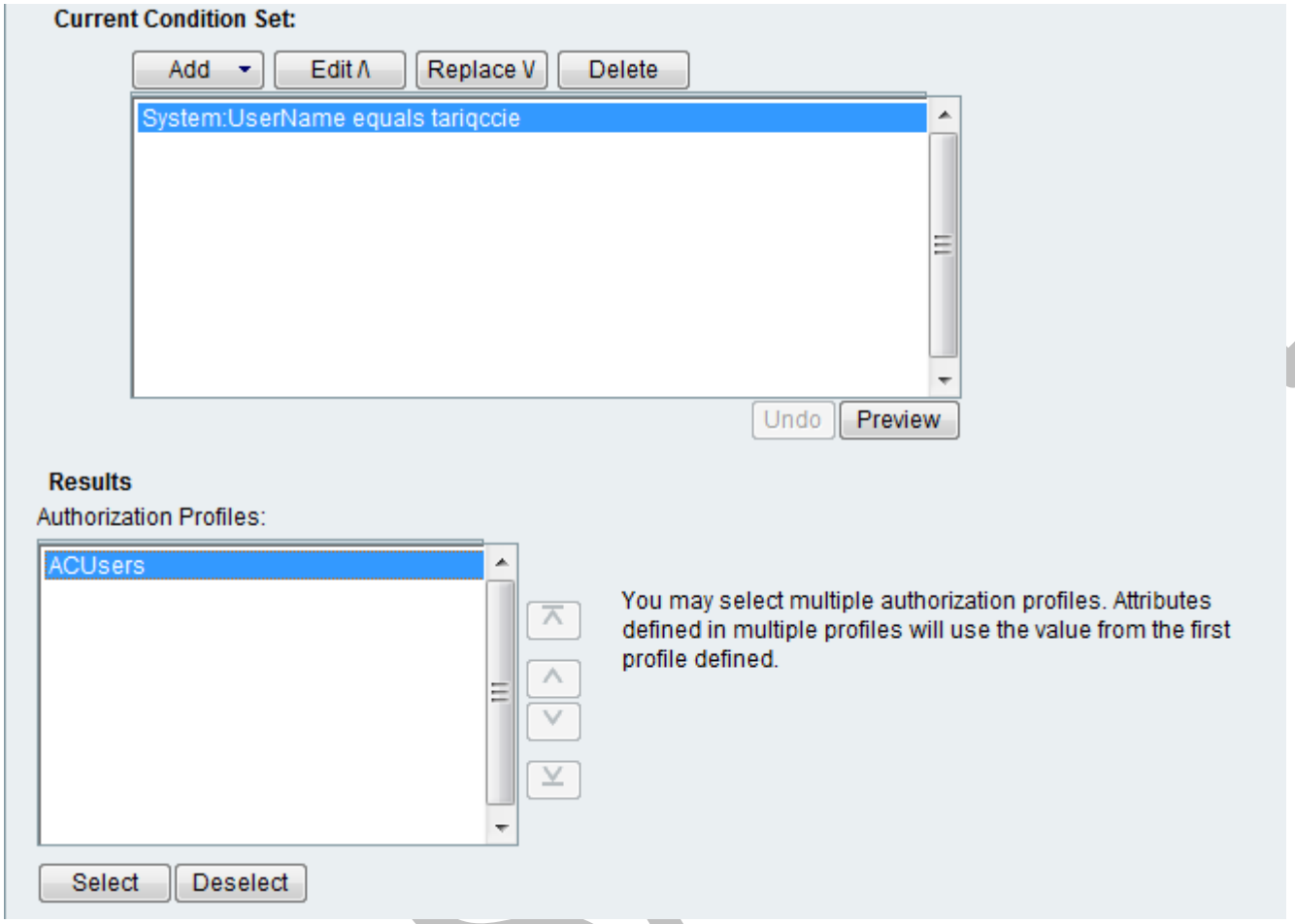
You will need to create Authorization Profile which will be applied to the user created above

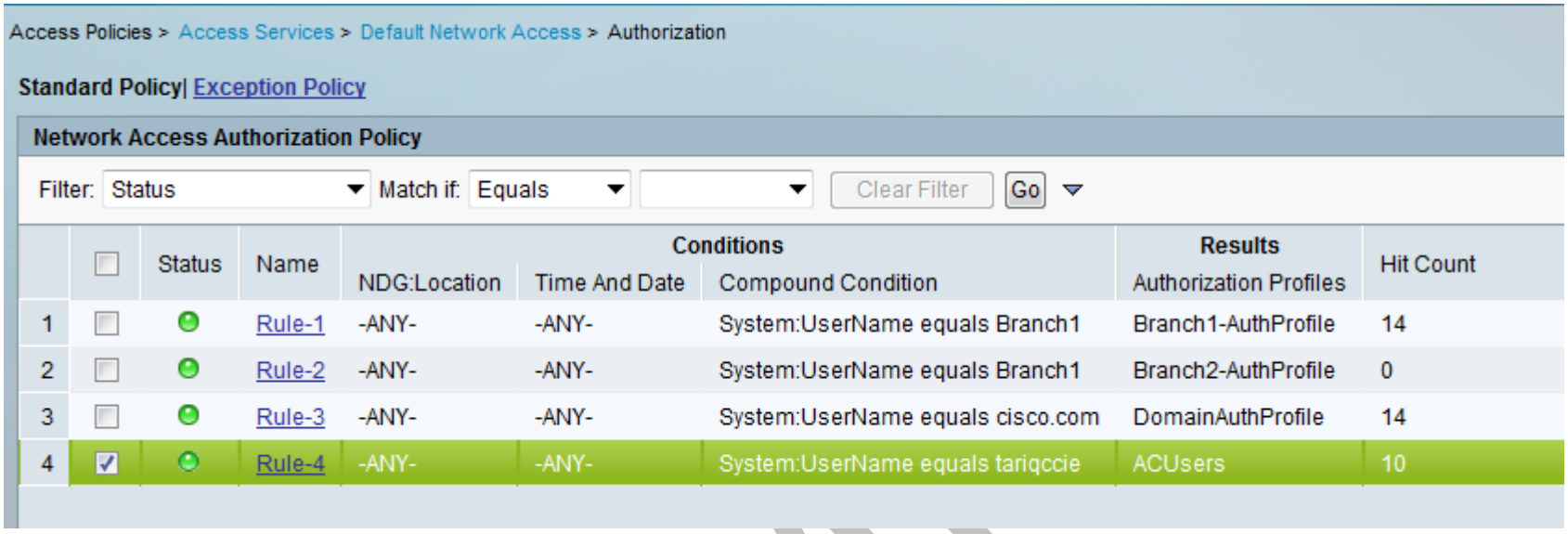
```
Auth Profile:     ACUsers     [Cisco-AV-Pair: ipsec:route-accept=any]
                              [Cisco-AV-Pair: ipsec:route-set=interface]
                              [Cisco-AV-Pair: ipsec:addr-pool=FlexPool]
                              [Cisco-AV-Pair: ipsec:route-set=access-list99]
```

Navigate to *Policy Elements -> Authorization and Permissions -> Network Access -> Authorization Profiles*

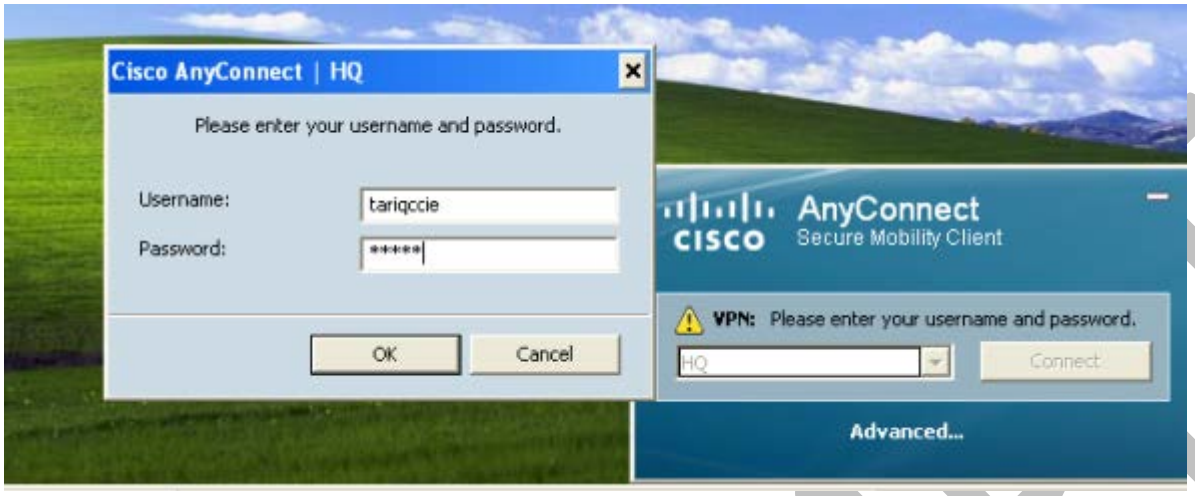




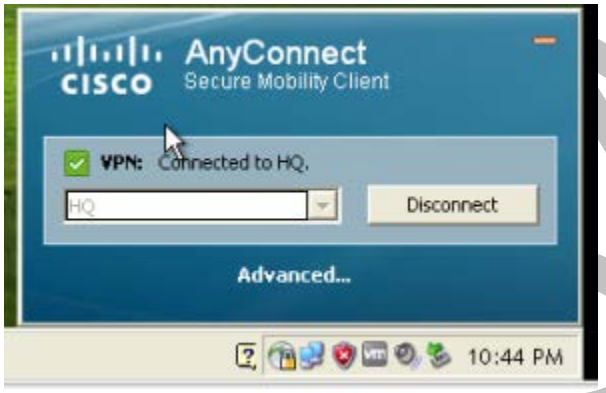




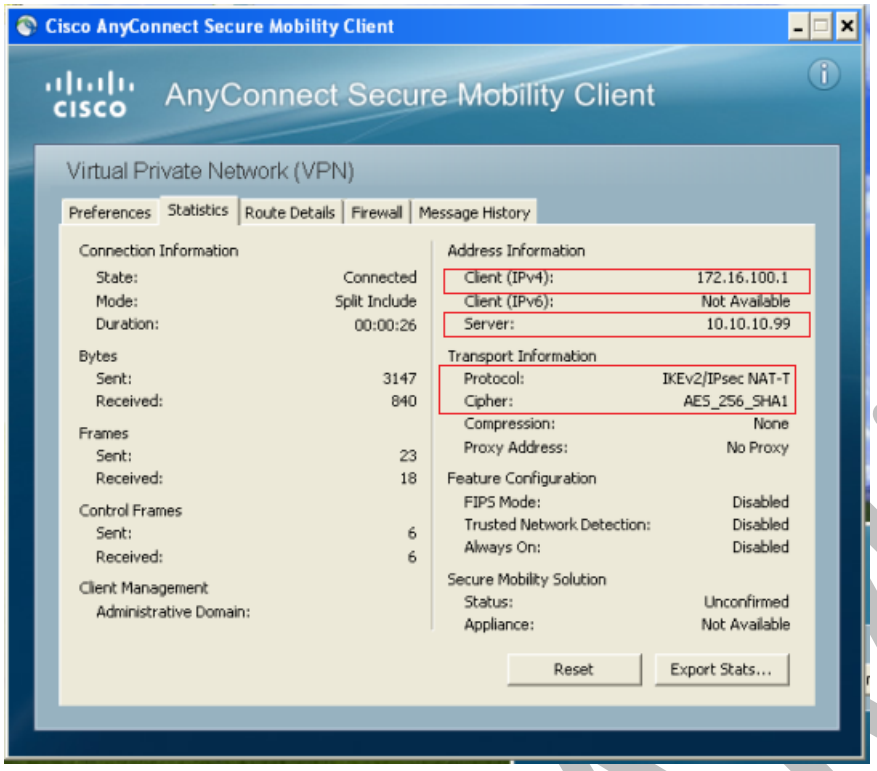
		Status	Name	Conditions			Results	Hit Count
				NDG:Location	Time And Date	Compound Condition	Authorization Profiles	
1	☐	●	Rule-1	-ANY-	-ANY-	System:UserName equals Branch1	Branch1-AuthProfile	14
2	☐	●	Rule-2	-ANY-	-ANY-	System:UserName equals Branch1	Branch2-AuthProfile	0
3	☐	●	Rule-3	-ANY-	-ANY-	System:UserName equals cisco.com	DomainAuthProfile	14
4	☑	●	Rule-4	-ANY-	-ANY-	System:UserName equals tariqccie	ACUsers	10



If you configured everything properly, you will be connected to VPN HeadEnd.



Notice the fields highlighted in RED. **Client IPv4 address (assigned), Protocol/Cipher & Split Tunneling ACL**



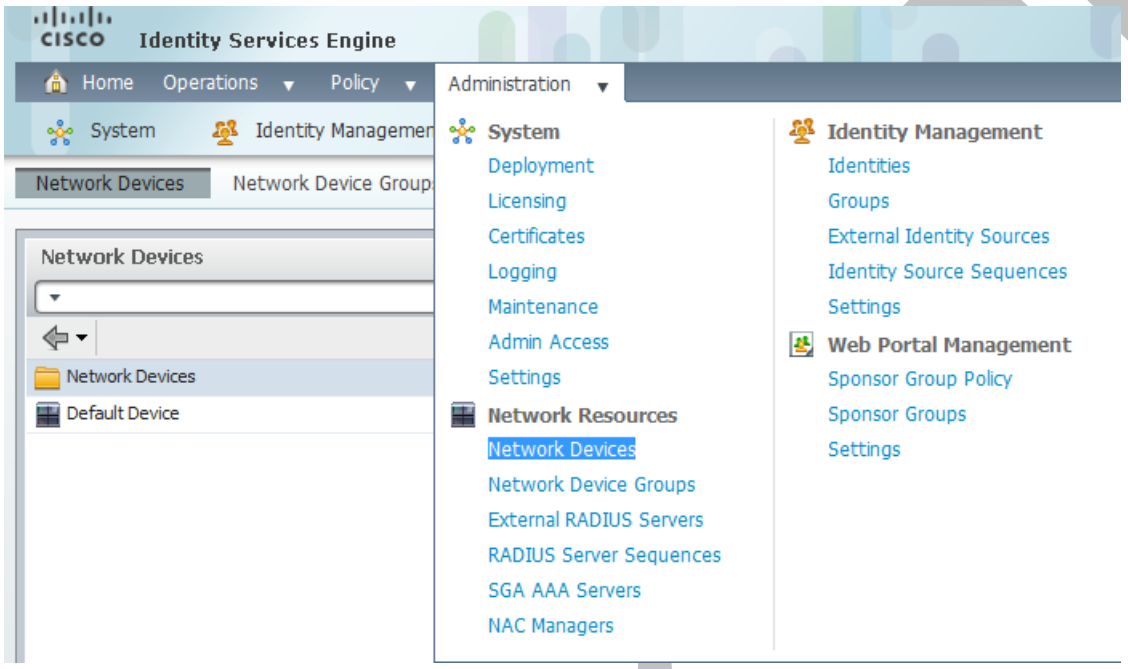
Cisco Identity Services Engine 1.1.2 CONFIGURATION :

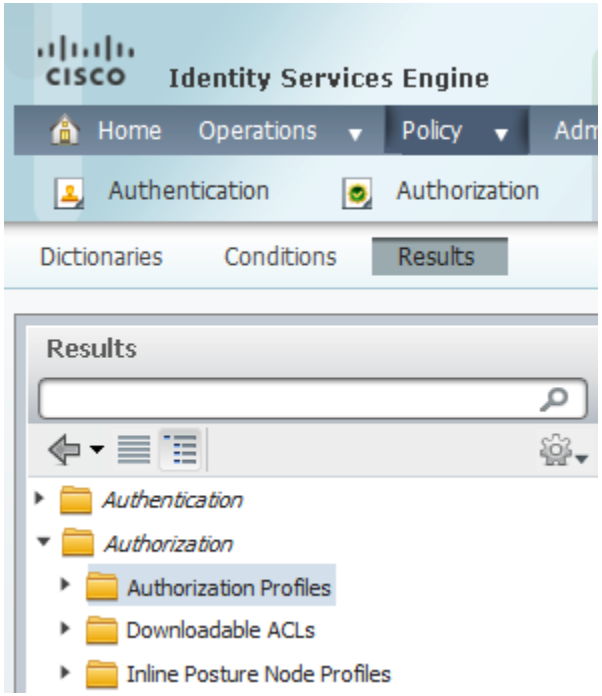
Here are the items you need to configure on RADIUS Server (**Cisco ISE 1.1.X**)

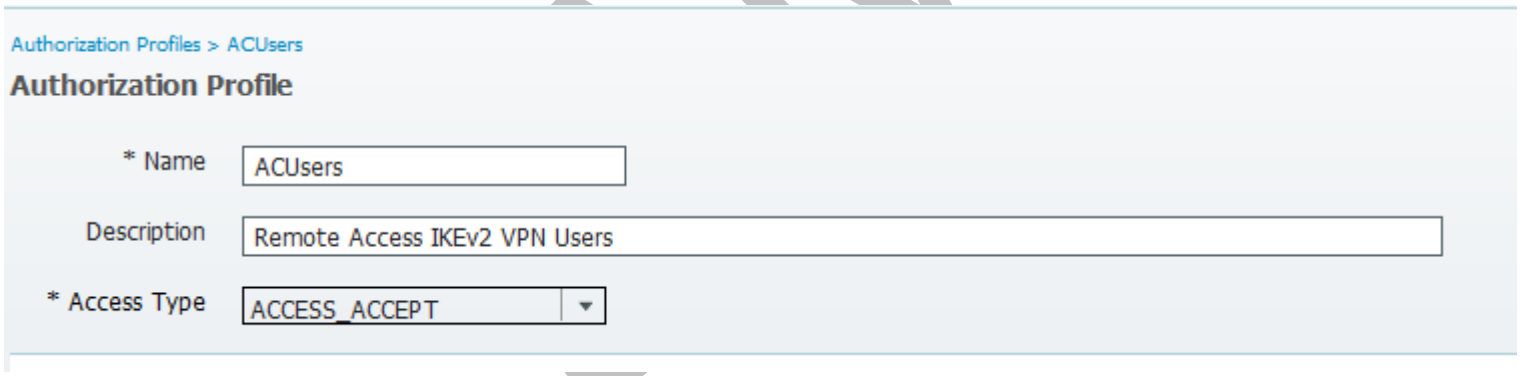
- **Add HQ as a Network Device in ISE**

Network Devices (VPN HeadEnd- **HQ**) will be created by navigating to following link:

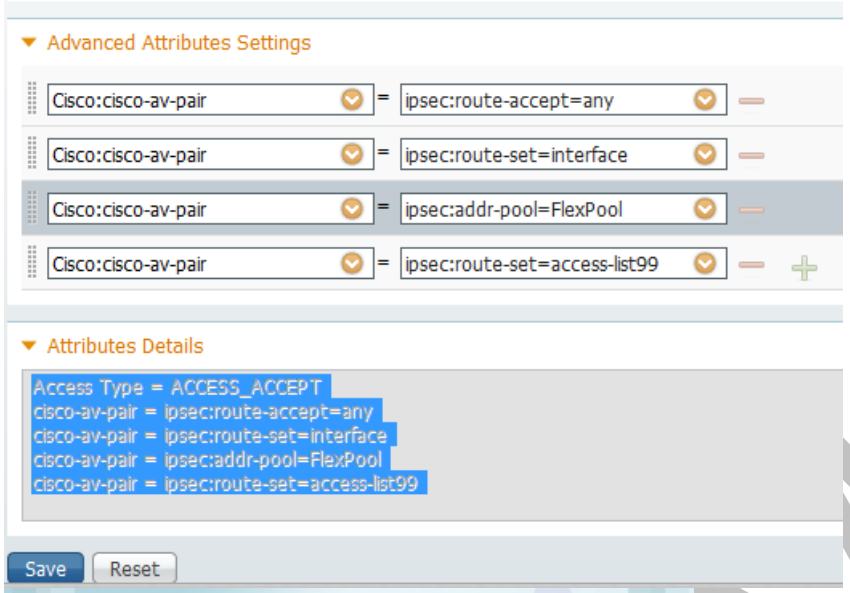
Navigate to *Administration -> Network Resources -> Network Devices*

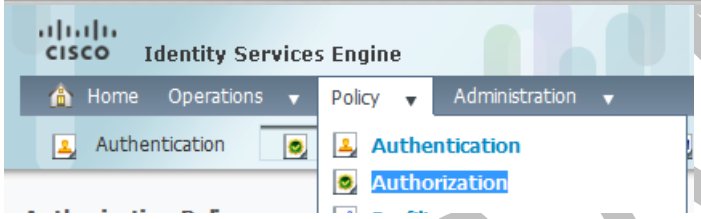


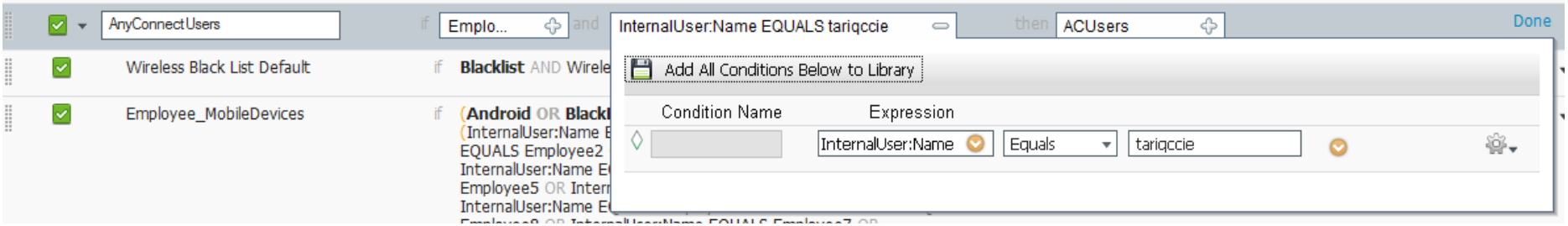


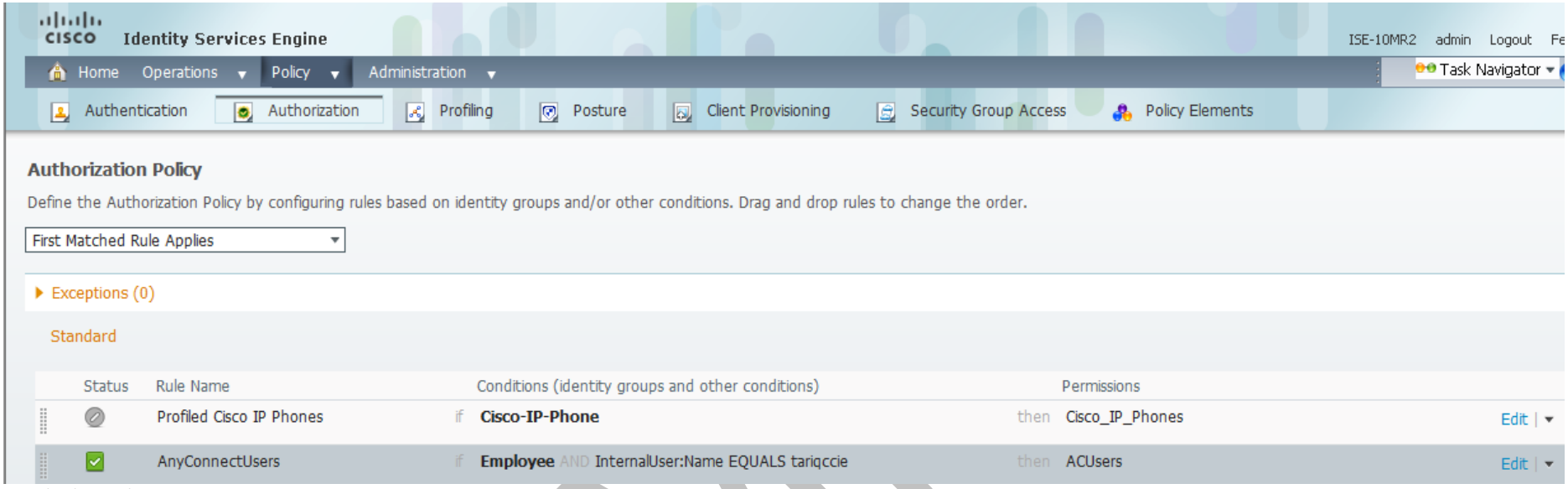


Add the following attributes under **'Advanced Attribute Settings'** & click **save.**





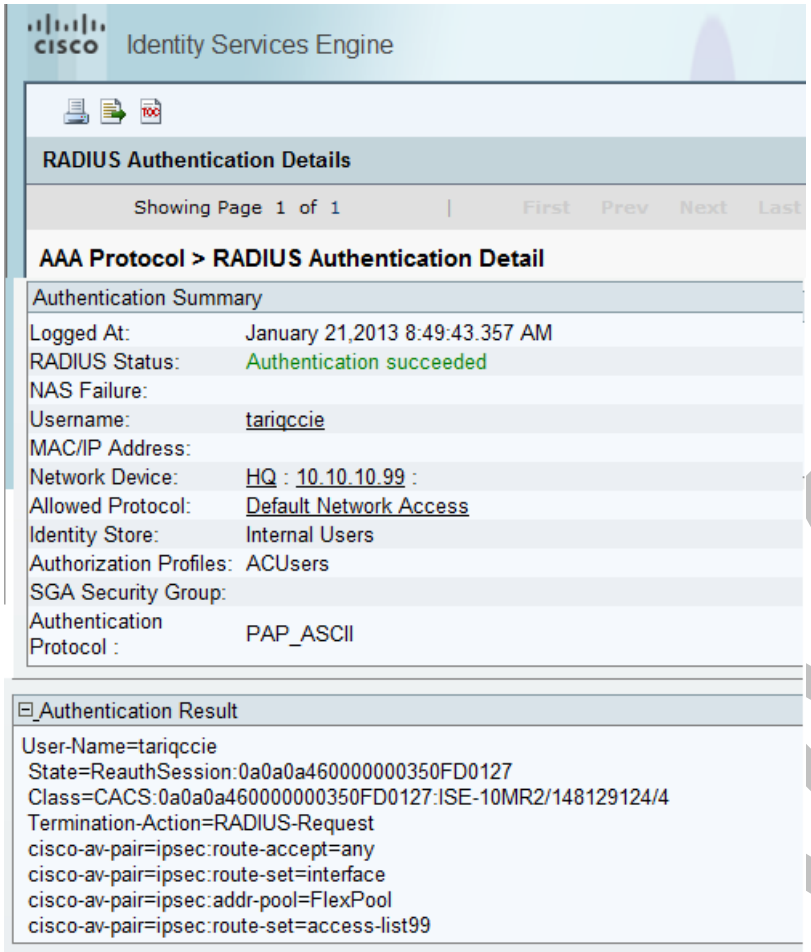




Verification :

First test & make sure that **AnyConnect VPN User** can successfully connect to **VPN Head-End (HQ)**

```
HQ#test aaa group AAA tariqccie Cisco123 new-code
User successfully authenticated

USER ATTRIBUTES

username              0   "tariqccie"
Termination-Action    0    True
route-accept          0   "any"
route-set             0   "interface"
addr-pool             0   "FlexPool"
route-set             0   "access-list99"
```



Only traffic to 172.16.1.0 will be encrypted.