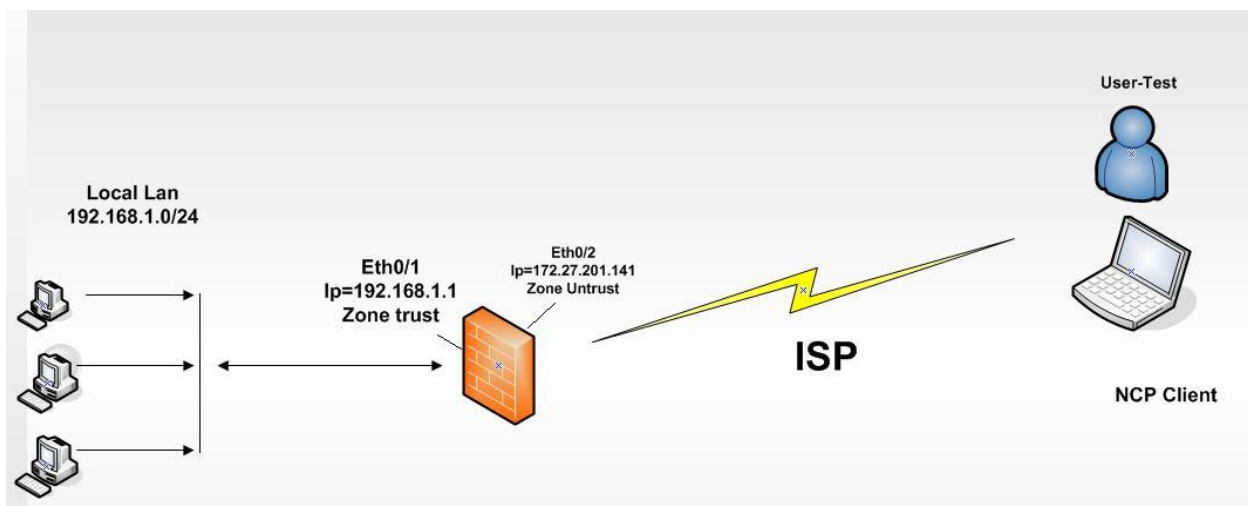


NCP Secure Entry Client Configuration With Certificates

This is a simple configuration of the NCP Client connecting to a Juniper firewall with certificate. Note that there are other possible configurations.

This is useful for those familiar with configuring NS Remote and are new to the NCP client.

Network diagram:



A. Firewall configuration

1. First of all go to ca server and download the ca server public certificate
2. Click on download ca certificate

Microsoft Active Directory Certificate Services -- Home

This Web browser does not support the generation of certificate requests.

Welcome

Use this Web site to request a certificate for your Web browser, e-mail client, or other program. By using a certificate, you can verify your identity to people you communicate with over the Web, sign and encrypt messages, and, depending upon the type of certificate you request, perform other security tasks.

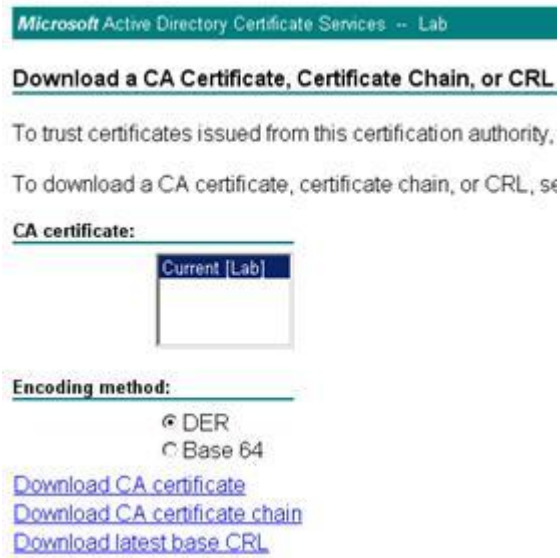
You can also use this Web site to download a certificate authority (CA) certificate, certificate chain, or certificate revocation list (CRL), or to view the status of a pending request.

For more information about Active Directory Certificate Services, see [Active Directory Certificate Services Documentation](#). To generate certificate requests, upgrade to the latest version of Internet Explorer.

Select a task:

[Request a certificate](#) Href="certrqxt.asp" Href="certrqus.asp" >Request a certificate
[View the status of a pending certificate request](#)
[Download a CA certificate, certificate chain, or CRL](#)

3. Then click on
 1. Download ca certificate and save it on your computer
 2. Downlad Latest base CRL and save it on your computer



4. Then open the web gui of the firewall and go to **objects->certificate**

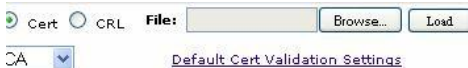
Select Ca in the drop down and browse the ca certificate and hit on load



5. Now select the CRL option and load the ca certificate and hit on load



6. Both The CA and CRL certificate will be loaded in the firewall. Make sure status is active



Issuer	Friendly Name	Type	Serial#	Expired	Status	Config
Lab Server Settings	s	CA	0f075fc09b8065bd488fb295504fb5ae	03-03-2015 13:31	Active	Detail, R
Lab	-	CRL	0000000000000000	05-20-2011 07:42	Active	Detail, R

- Now click on new and create a local certificate By giving the appropriate field as you wish and hit generate

Certificate Subject Information

Name:	Juniper
Phone:	123456
Unit/Department:	support
Organization:	Network
County/Locality:	california
State:	ca
Country:	US
E-mail:	juniper@juniper.net
IP Address:	1.1.1.1
FQDN:	www.juniper.net

Key Pair Information:

RSA DSA ECDSA

Create new key pair of length.

- Copy the generated CSR

Certificate Request

```
-----BEGIN CERTIFICATE REQUEST-----
MIIDWTOCAkECAQAwgcsxCzAJBgNVBAYTA1VTHQswCQYDVQQIEwJjYU90MDEwMDQw
BxMKY2FsaWZvcms5pYEQMA4GA1UEChMHMTM0d29ya2EQMA4GA1UEC3VwG9y
dDEQMA4GA1UEAxMHNS4xLjEUMTEVHBMGA1UEAxMNSk4xNDkwMEExQURCMQ8wDQYD
VQQDEwYxMjMONTYxEDA0BGNVBAWNTB3JzYS1rZXkxGDAWBgNVBAMTD3d55qdW5p
cGVyLm5ldDEQMA4GA1UEAxMHSnVuaXB1c3CCASIDQYJKoZIhvcNAQEBBQADggEP
ADCCAQoCggEBAMBS0uEDQ1OLYvrzGgsXjJ20Vhna2C/ORwon22XRDOXLhFONqOen
s06IyS2gU0o5Mft5PZRoe3Jb/4X6rBCWQop3GfyCg28kdIWHFV1mbWjTVzVvwTMk
Xw37FxxVOr1fyt9r34h02I1Jy2qp0quu3Fa0xPLYFm81qHVR0PTawS0gdwhn5vE9
Vj7ApuVbe7Lw/xnEJoIwuTGDxaWUNMddRYLCHY4duJ34m+r1RU2cN2945trI/FWJ
-----
```

- Now go to the ca server and click on request certificate

This Web browser does not support the generation of certificate requests.

Welcome

Use this Web site to request a certificate for your Web browser, e-mail client, or other program. By using a certificate, you can verify your identity to people you communicate with over the Web, sign and encrypt messages, and, depending upon the type of certificate you request, perform other security tasks.

You can also use this Web site to download a certificate authority (CA) certificate, certificate chain, or certificate revocation list (CRL), or to view the status of a pending request.

For more information about Active Directory Certificate Services, see [Active Directory Certificate Services Documentation](#). To generate certificate requests, upgrade to the latest version of Internet Explorer.

Select a task:

[Request a certificate](#)

Href="certrqst.asp" Href="certreqs.asp" >Request a certificate

[View the status of a pending certificate request](#)

[Download a CA certificate, certificate chain, or CRL](#)

10. Then click on create and submit a request to this ca

Advanced Certificate Request

The policy of the CA determines the types of certificates you can request. Click one of the following links:

[Create and submit a request to this CA.](#)

[Submit a certificate request by using a base-64-encoded CMC or PKCS #10 file.](#)

11. And paste the csr and hit on submit

Submit a Certificate Request or Renewal Request

To submit a saved request to the CA, paste a base-64-encoded CMC or PKCS #10 file into the Saved Request box.

Saved Request:

Base-64-encoded certificate request (CMC or PKCS #10 or PKCS #7):

```

-----BEGIN CERTIFICATE REQUEST-----
MIIDWTCCAkECAQAwwgcsxCzAJBgNVBAYTA1VUMQsw
BxMKY2FsaWZvcms5pYTEQMA4GA1UEChMHMTM0d29y
dDEQMA4GA1UEAxMHMS4xLjEuNTEVMBMGAlUEAxHM
VQQDEwYxMjMONTYxEDA0BgNVBAMTB3JzYS1rZXkx
cGVyLm51dDEQMA4GA1UEAxMHMSnVuaXB1c3CCASIW

```

Additional Attributes:

Attributes:

Submit >

12. Download the certificate and save it

Certificate Issued

The certificate you requested was issued to you.

DER encoded or Base 64 encoded



[Download certificate](#)
[Download certificate chain](#)

- 13. Go to the firewall browse the saved file and load this certificate

Objects > Certificates SSG550:NSR

Load Cert CRL File: C:\Documents and Set Browse... Load

Show Local [Default Cert Validation Settings](#)

Issuer	Friendly Name	Type	Serial#	Expired	Status	Configure
-	-	LOCAL	0000000000000000	-	Key Pair	Detail, Remove Submit Request

- 14. You will see this loaded certificate. Its status will be active

Load Cert CRL File: Browse... Load

Show Local [Default Cert Validation Settings](#)

Issuer	Friendly Name	Type	Serial#	Expired	Status	Configure
Lab	11	LOCAL	5e89574600000000142	03-03-2015 13:31	Active	Detail, Remove

VPN CONFIGURATION

- 15. go to **objects->users->local** and create a new ike user for the ncp client to use
Select the **Ike user** and select **“use distinguished name for id”** and give appropriate fields and hit ok

Auth/IKE/XAuth/L2TP/WAN User

User Name

Status Enable Disable

IKE User Number of Multiple Logins

Simple Identity

Use Distinguished Name For ID

CN

OU

Organization

Location

State

Country

E-mail

Container

WAN User

Authentication User User Password

XAuth User Confirm Password

The IKE user will be created

Objects > Users > Local SSG550: NSRP(M)

List per page Ne

Name	Type	Group	Status	Identity	Configure
test	IKE	-	Enabled	CN=www.juniper.net,OU=support,O=Network,L=california,ST=ca,C=US,Email=juniper@juniper.net,DC=,	Edit Remove

16. Now create a dial up vpn , go to **vpn-> autokey advanced->gateway** and click new and select the dialup user

Gateway Name

Version IKEv1 IKEv2

Remote Gateway

Static IP Address IPv4/v6 Address/Hostname

Dynamic IP Address Peer ID

Dialup User User

Dialup User Group Group

ACVPN-Dynamic Local ID

ACVPN-Profile

OK Cancel Advanced

17. Hit on advance. **Don't give any preshared key**, select outgoing interface and proposals and mode as aggressive

IKEv2 Auth Method

Self

Peer

Preshared Key Use As Seed

Local ID (optional)

Outgoing Interface

Security Level

Predefined Standard Compatible Basic

User Defined Custom

Phase 1 Proposal

Mode (Initiator) Main (ID Protection) **Aggressive**

Enable NAT-Traversal

UDP Checksum

Keepalive Frequency Seconds (0~300)

Peer Status Detection

Heartbeat

Hello Seconds (1~3600, 0: disable)

Reconnect Seconds (60~9999, 0: default)

Threshold (2~9999)

DPD Interval Seconds (2~28800, 0: disable)

18. create the phase 2. Click on **vpn-> autokey ike -> new**
Select the gateway and click on advance

VPN Name:

Remote Gateway: Predefined Create a Simple Gateway

Predefined:

Gateway Name:

Version: IKEv1 IKEv2

Type: Static IP Dynamic IP Dialup User Dialup Group

Address/Hostname:

Peer ID:

User:

Group:

Local ID: (optional)

Preshared Key: Use As Seed:

Security Level: Standard Compatible Essential

Outgoing Interface:

Gateway: Tunnel Towards Hub:

Binding to Tunnel:

Buttons:

19. Select the phase 2 proposals and hit return and ok

Predefined: Standard Compatible Basic

User Defined: Custom

Phase 2 Proposal:

Replay Protection:

Transport Mode:

Bind to: None Tunnel Interface Tunnel Zone

Tunnel Interface:

Tunnel Zone:

Proxy-ID Check:

DSCP Marking: Disable Enable

Dscp Value:

VPN Group: Weight:

VPN Monitor:

Source Interface:

Destination IP:

Optimized:

Rekey:

Buttons:

20. Now create a policy for the dial up vpn to access the local lan.

Name (optional)

Source Address New Address /
 Address Book Entry

Destination Address New Address /
 Address Book Entry

Service

Application

WEB Filtering

Action

Antivirus Profile

Tunnel VPN
 Modify matching bidirectional VPN policy

L2TP

Logging at Session Beginning

Position at Top

Session-limit

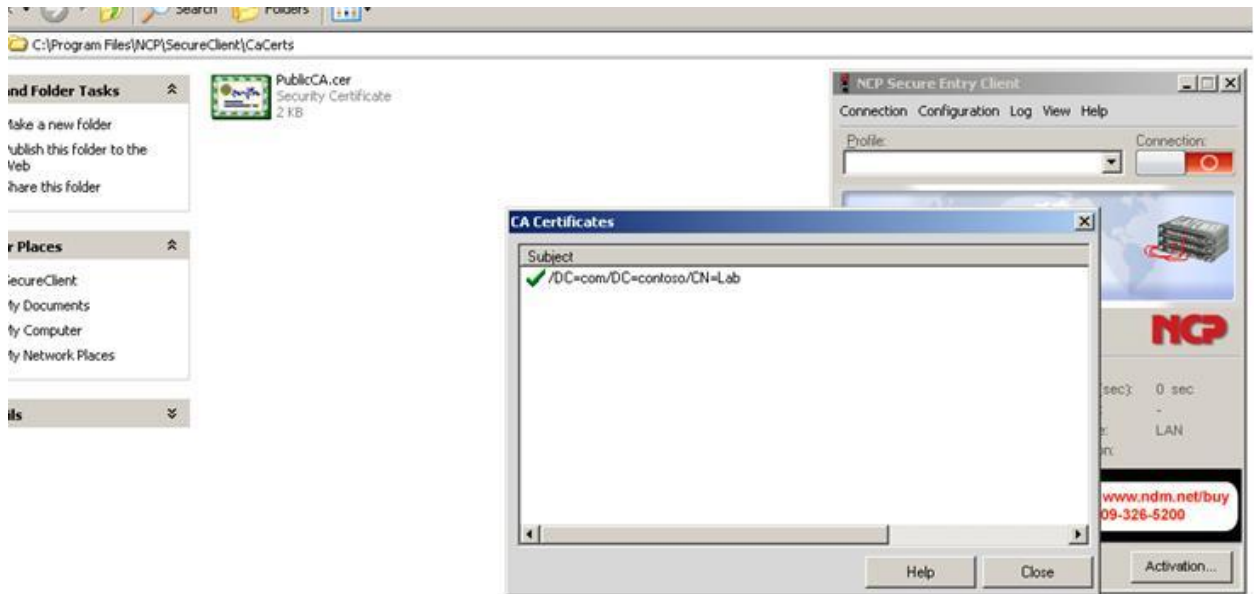
A. NCP configuration

1. Save the ca server public certificate into the ncp ca certificates folder

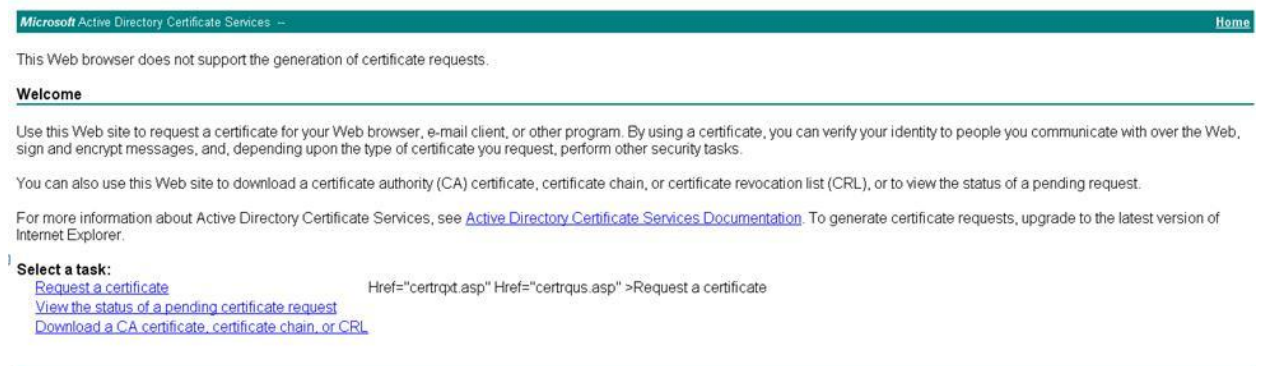
Go to C:\Program Files\NCP\SecureClient\CaCerts

And paste the public ca certificate there

Now in NCP go to **connection-> display ca certificate** you must see the saved ca there which we pasted in ca cert folder



2. For the Local certificate for the NCP client go to the ca server and click on **request certificate**



3. Click create and submit a request to this ca



4. Create the certificate by Entering the fields, It should match with the firewall ike user “test” and make sure to check “mark key as exportable” option and submit .

Microsoft Active Directory Certificate Services -- Lab

Advanced Certificate Request

Identifying Information:

Name:	www.juniper.net
E-Mail:	juniper@juniper.net
Company:	support
Department:	Network
City:	california
State:	ca
Country/Region:	US

Type of Certificate Needed:

Client Authentication Certificate

Key Options:

Create new key set Use existing key set

CSP: Microsoft Enhanced Cryptographic Provider v1.0

Key Usage: Exchange Signature Both

Key Size: 2048 Min: 384 Max: 16384 (common key sizes: 512 1024 2048 4096 8192 16384)

Automatic key container name User specified key container name

Mark keys as exportable

Enable strong private key protection

5. Then install the certificate

Microsoft Active Directory Certificate Services -- Lab

Certificate Issued

The certificate you requested was issued to you.



[Install this certificate](#)

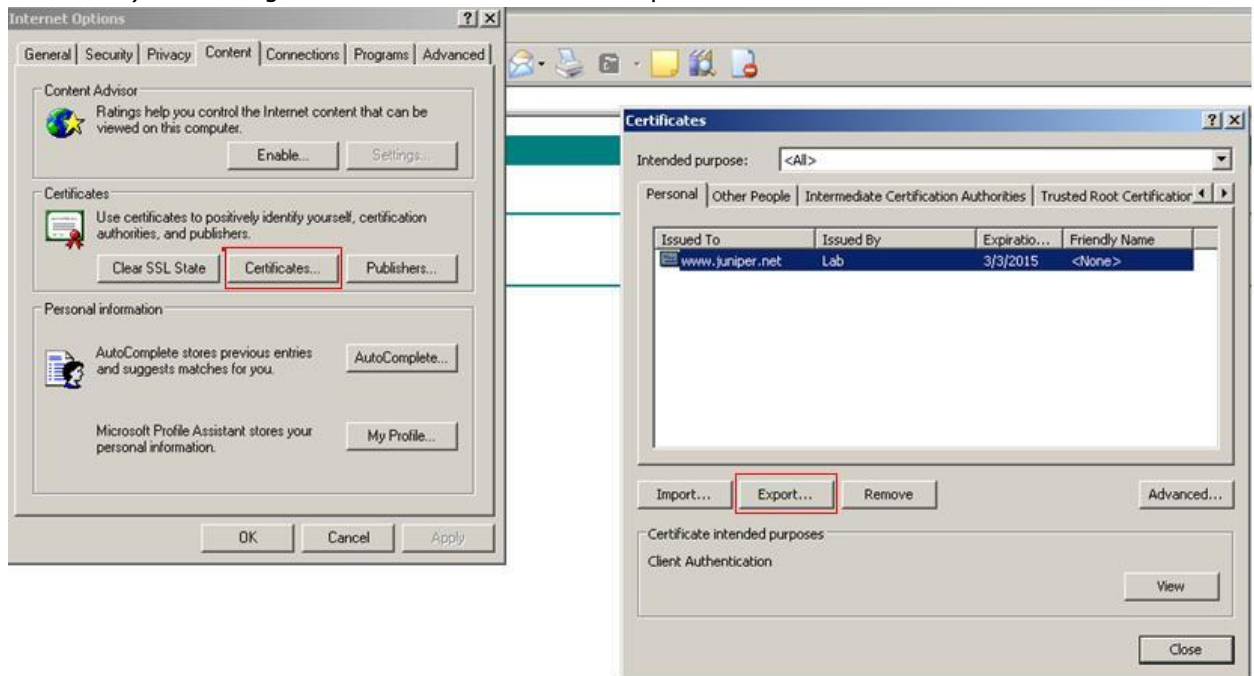
Save response

Click **Yes** for the prompt to install the certificate,

6. Make sure the certificate is installed successfully



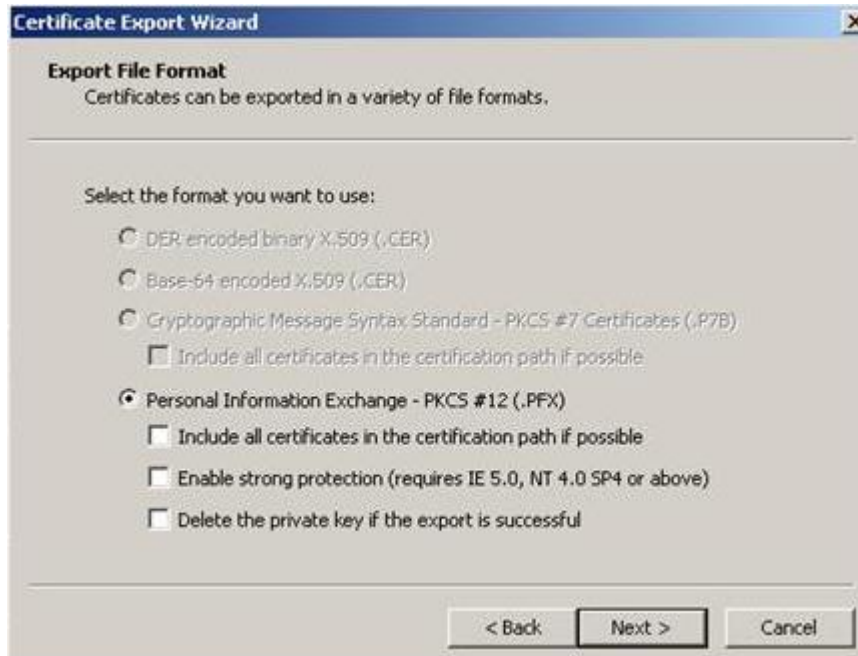
7. In the CA Store (you can open it by internet explorer->tools-> internet option->content->certificate) select the generated certificate and chose Export



8. In the Certificate Export Wizard select "**Yes, export the private key**" and click next



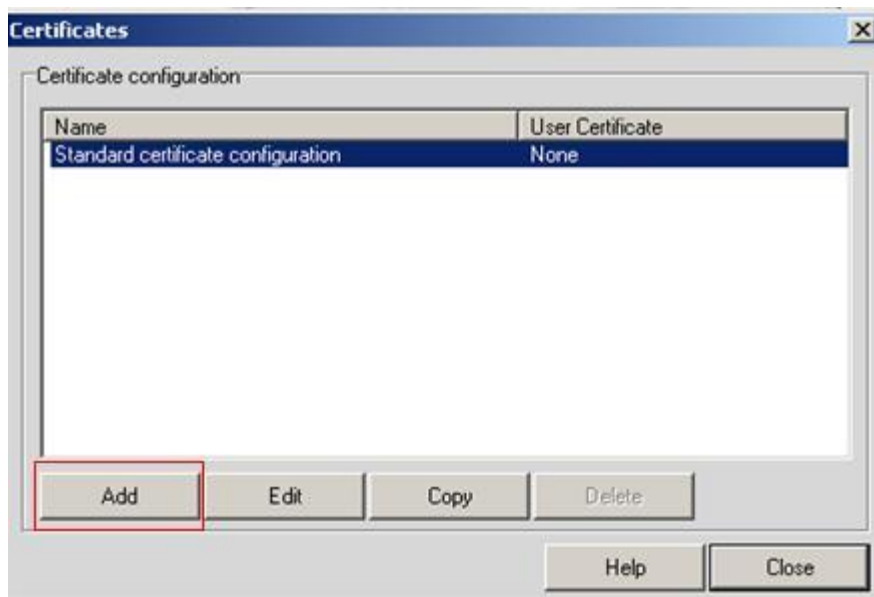
9. Select "**Personal Information Exchange – PKCS #12**" and uncheck other box and hit next



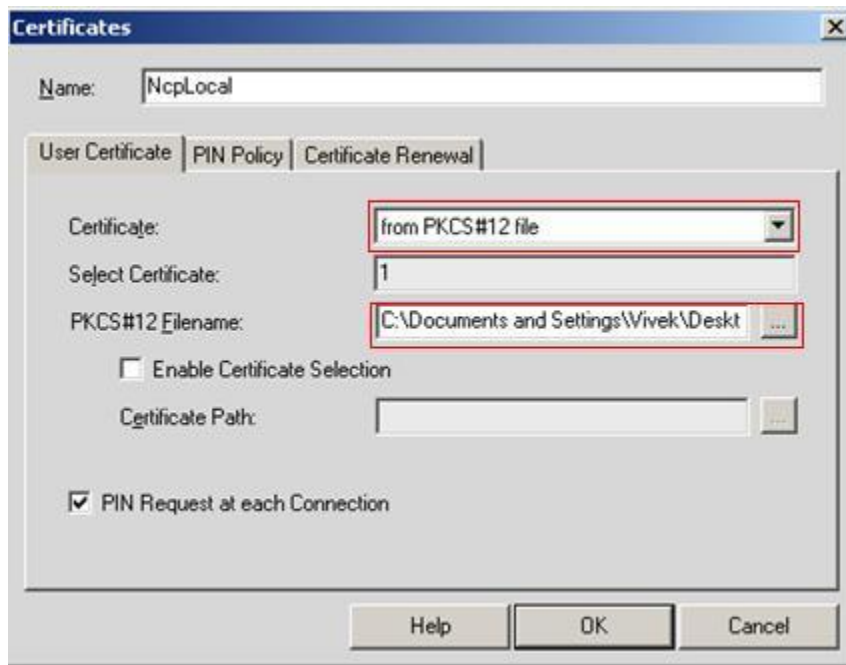
10. Enter the password e.g 12345678



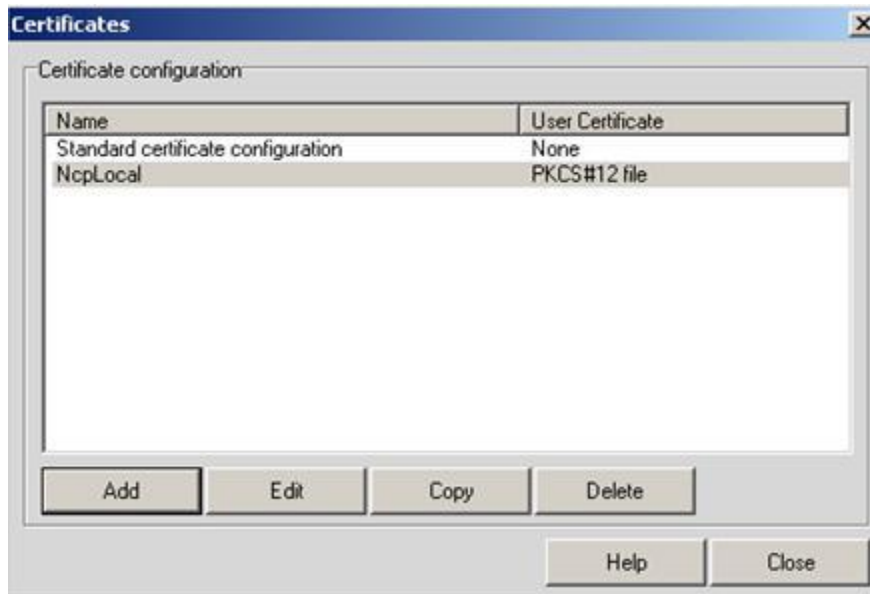
11. Click on next and save the file. The file will be saved as **“.pfx”** extension and Complete the Wizard.
12. Locate the saved file and change the extension to **“.p12”** for e.g from “filename.pfx” change it to “filename.p12”
13. Then Import this certificate to NCP. In ncp client Go to **configuration-> certificate** and add



14. Give certificate name, select **from pkcs#12 in certificate**, browse the “.p12” file which we exported and enable pin request at each connection (pin request is optional) and hit ok



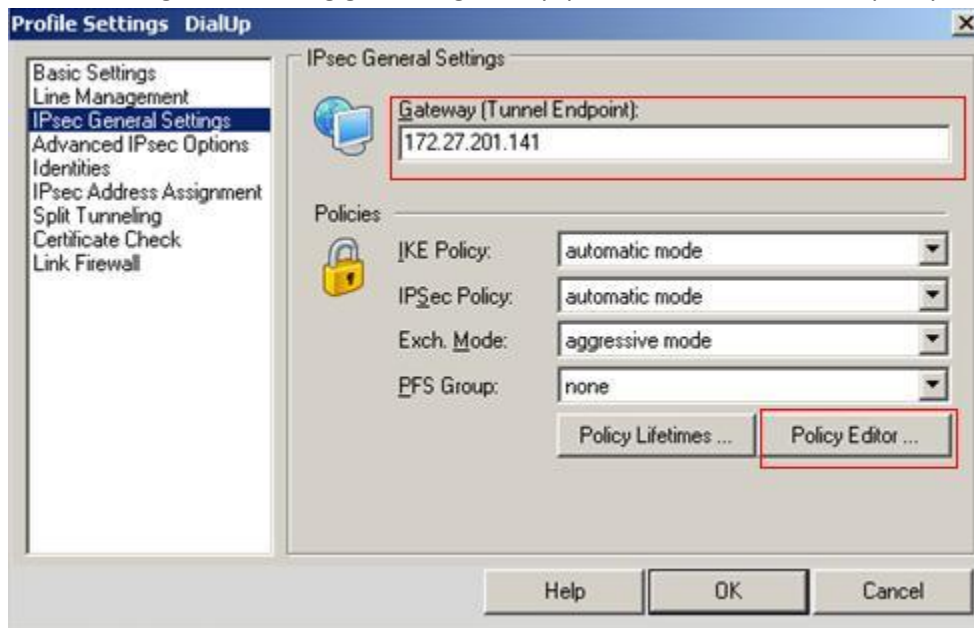
15. The certificate will be imported in NCP



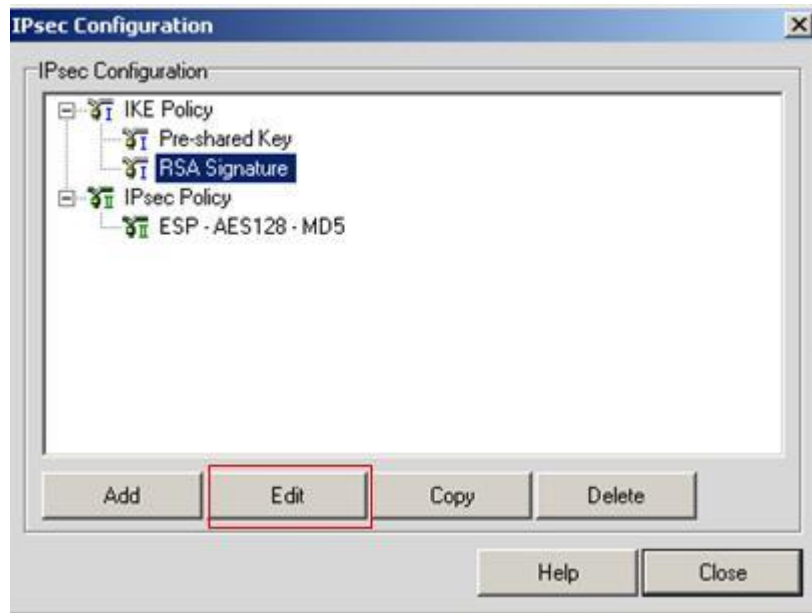
16. Now configure the vpn for ncp client, Give the profile name



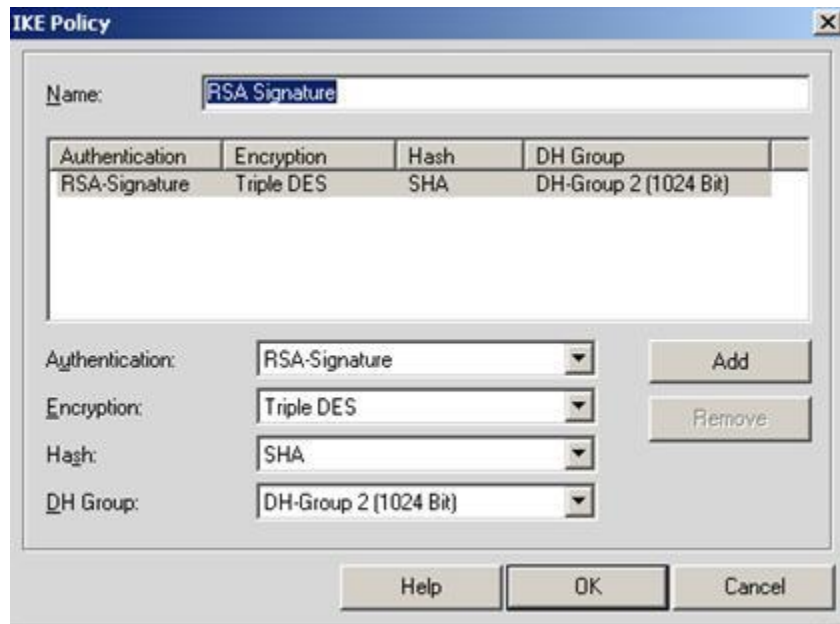
17. Under IPsec general setting give the gateway ip of the firewall and click policy editor



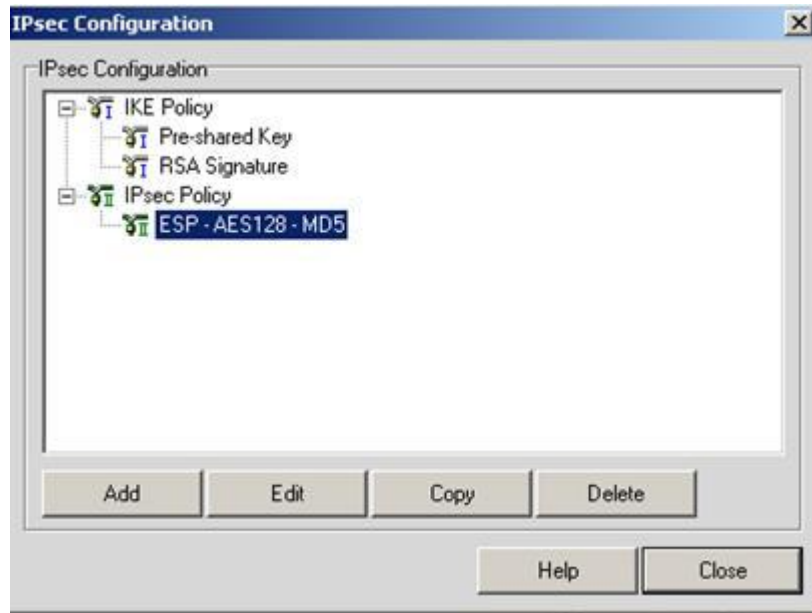
18. Select rsa signature and edit it



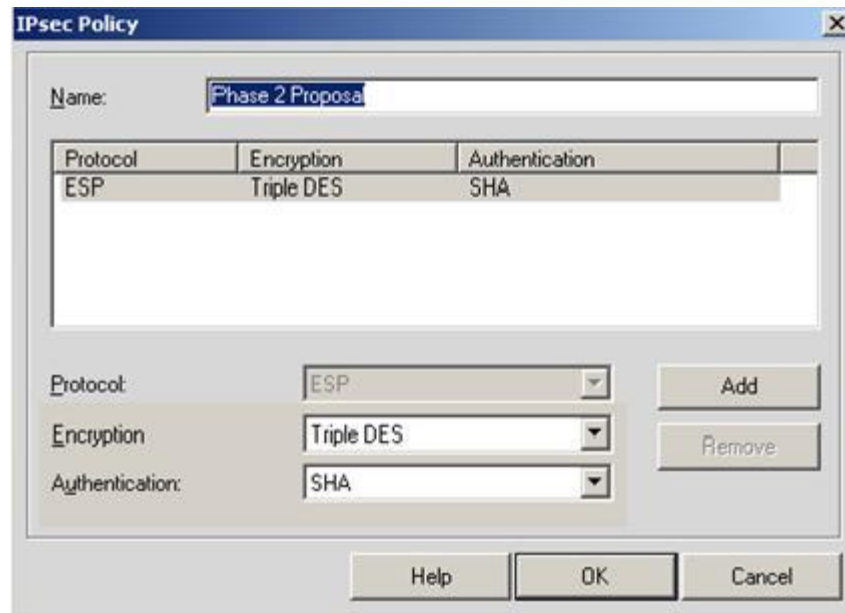
19. Select the proposal given in the phase 1 of firewall and hit ok



20. Now edit the ipsec policy(phase 2 proposal)



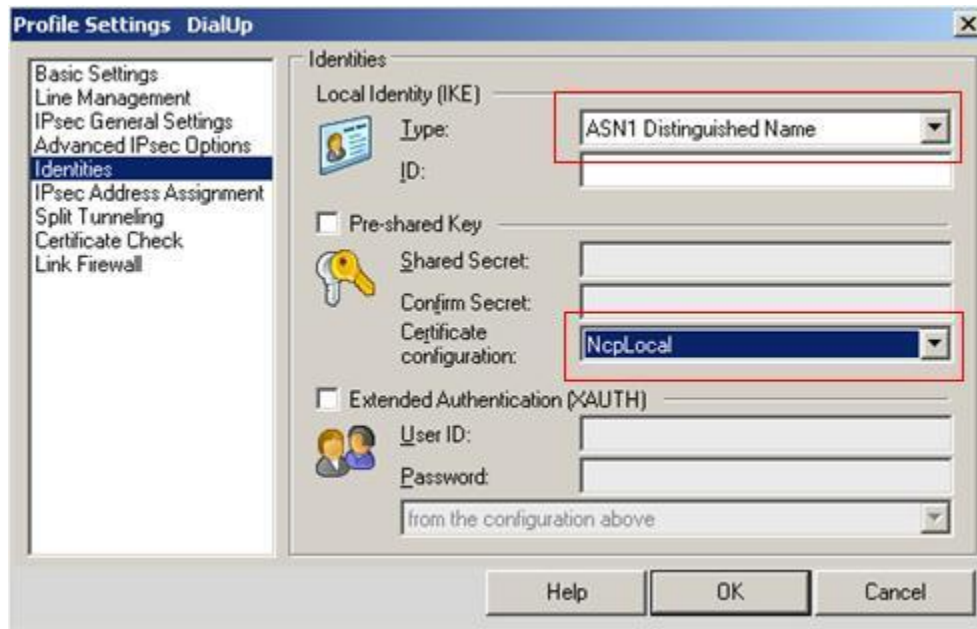
21. Give the proposal as given in the phase 2 of the firewall and hit ok



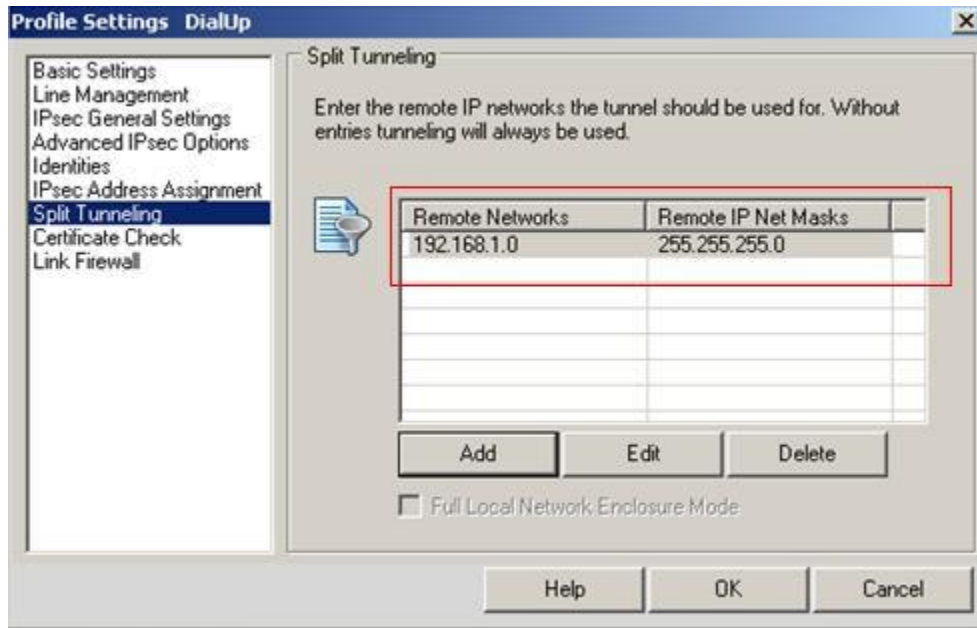
22. Select the appropriate settings like proposals, mode and pfs as per firewall settings



23. Now under Identities select **ASN1 distinguished name** and certificate as the local imported certificate



24. Then click on split tunneling and add the remote subnet which you need to access

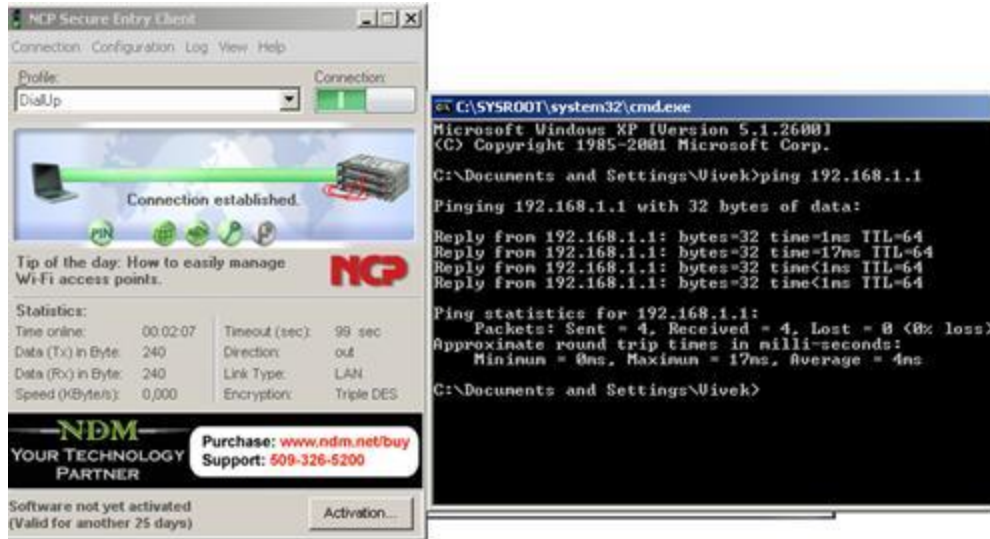


Then hit on ok and connect the ncp client

25. It will prompt for pin, Just enter the pin which we used (12345678)



26. The connection will be established



Basic Troubleshooting

You can run the following command

Debug pki detail

Debug ike detail

And can verify the output of the connection

SSG550-> get db st

```
## 2011-05-17 00:00:13 : IKE<172.27.199.207> ike packet, len 641, action 1
## 2011-05-17 00:00:13 : IKE<172.27.199.207> Catcher: received 613 bytes from socket.
## 2011-05-17 00:00:13 : IKE<172.27.199.207> ***** Recv packet if <ethernet0/2> of vsys <Root> *****
## 2011-05-17 00:00:13 : IKE<172.27.199.207> Catcher: get 613 bytes. src port 500
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > ISAKMP msg: len 613, npx 1[SA], exch 4[AG], flag 00
## 2011-05-17 00:00:13 : IKE<172.27.199.207 > Recv : [SA] [KE] [NONCE] [ID] [CERT-REQ] [VID] [VID] [VID] [VID]
## 2011-05-17 00:00:13 : [VID] [VID] [VID] [VID] [VID] [VID]
## 2011-05-17 00:00:13 : valid id checking, id type:ASN1_DN, len:156. ( We have received the asn1_dn from the
NCP client)
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > Validate (585): SA/52 KE/132 NONCE/44 ID/156 CERT-REQ./5
VID/12 VID/20 VID/20
## 2011-05-17 00:00:13 : IKE<172.27.199.207> Receive Id (type=DN) in AG mode, retrieve
DN=Email=juniper@juniper.net,CN=www.juniper.net,OU=Network,O=support,L=california,ST=ca,C=US
, idlen = 90 ( We have received the field configured on the NCP client local certificate, Next steps firewall is
matching them with the ike user "test" configured field )
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > peer dn has 7 elements.
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > compare user id<1>.
## 2011-05-17 00:00:13 : normalize_user_wildcard_dn_id:
input<CN=www.juniper.net,OU=Network,O=support,L=california,ST=ca,C=US,Email=juniper@juniper.net,DC=,>
## 2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: <0><CN=www.juniper.net>.
## 2011-05-17 00:00:13 : get_dn_element_type_mask: mask<00000001>
## 2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: got <0><8bfff688><CN=www.juniper.net>.
## 2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: <1><OU=Network>.
## 2011-05-17 00:00:13 : get_dn_element_type_mask: mask<00000002>
## 2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: got <1><8bfff69b><OU=Network>.
```

2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: <2><O=support>.
2011-05-17 00:00:13 : get_dn_element_type_mask: mask<00000004>
2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: got <2><8bfff6a6><O=support>.
2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: <3><L=california>.
2011-05-17 00:00:13 : get_dn_element_type_mask: mask<00000008>
2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: got <3><8bfff6b0><L=california>.
2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: <4><ST=ca>.
2011-05-17 00:00:13 : get_dn_element_type_mask: mask<00000010>
2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: got <4><8bfff6bd><ST=ca>.
2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: <5><C=US>.
2011-05-17 00:00:13 : get_dn_element_type_mask: mask<00000020>
2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: got <5><8bfff6c3><C=US>.
2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: <6><Email=juniper@juniper.net>.
2011-05-17 00:00:13 : get_dn_element_type_mask: mask<00000040>
2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: got <6><8bfff6c8><Email=juniper@juniper.net>.
2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: <7><DC=>.
2011-05-17 00:00:13 : get_dn_element_type_mask: remaining after = bad for <DC=>.
2011-05-17 00:00:13 : get_dn_element_type_mask: mask<ffffff>
2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: got <7><00000000><empty>.
2011-05-17 00:00:13 : normalize_one_elem: input<CN=www.juniper.net>
2011-05-17 00:00:13 : normalize_one_elem: content<www.juniper.net>
2011-05-17 00:00:13 : normalize_one: A temp<CN=www.juniper.net,> in_len<15>
2011-05-17 00:00:13 : normalize_one: temp<CN=www.juniper.net,> len<19>
2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: ind<0> elem<CN=www.juniper.net,>len<19>
2011-05-17 00:00:13 : normalize_one_elem: input<OU=Network>
2011-05-17 00:00:13 : normalize_one_elem: content<Network>
2011-05-17 00:00:13 : normalize_one: A temp<OU=Network,> in_len<7>
2011-05-17 00:00:13 : normalize_one: temp<OU=Network,> len<11>
2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: ind<1>
elem<CN=www.juniper.net,OU=Network,>len<30>
2011-05-17 00:00:13 : normalize_one_elem: input<O=support>
2011-05-17 00:00:13 : normalize_one_elem: content<support>
2011-05-17 00:00:13 : normalize_one: A temp<O=support,> in_len<7>
2011-05-17 00:00:13 : normalize_one: temp<O=support,> len<10>
2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: ind<2>
elem<CN=www.juniper.net,OU=Network,O=support,>len<40>
2011-05-17 00:00:13 : normalize_one_elem: input<L=california>
2011-05-17 00:00:13 : normalize_one_elem: content<california>
2011-05-17 00:00:13 : normalize_one: A temp<L=california,> in_len<10>
2011-05-17 00:00:13 : normalize_one: temp<L=california,> len<13>
2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: ind<3>
elem<CN=www.juniper.net,OU=Network,O=support,L=california,>len<53>
2011-05-17 00:00:13 : normalize_one_elem: input<ST=ca>
2011-05-17 00:00:13 : normalize_one_elem: content<ca>
2011-05-17 00:00:13 : normalize_one: A temp<ST=ca,> in_len<2>
2011-05-17 00:00:13 : normalize_one: temp<ST=ca,> len<6>
2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: ind<4>
elem<CN=www.juniper.net,OU=Network,O=support,L=california,ST=ca,>len<59>
2011-05-17 00:00:13 : normalize_one_elem: input<C=US>
2011-05-17 00:00:13 : normalize_one_elem: content<US>
2011-05-17 00:00:13 : normalize_one: A temp<C=US,> in_len<2>

```
## 2011-05-17 00:00:13 : normalize_one: temp<C=US,> len<5>
## 2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: ind<5>
elem<CN=www.juniper.net,OU=Network,O=support,L=california,ST=ca,C=US,>len<64>
## 2011-05-17 00:00:13 : normalize_one_elem: input<Email=juniper@juniper.net>
## 2011-05-17 00:00:13 : normalize_one_elem: content<juniper@juniper.net>
## 2011-05-17 00:00:13 : normalize_one: A temp<Email=juniper@juniper.net,> in_len<19>
## 2011-05-17 00:00:13 : normalize_one: temp<Email=juniper@juniper.net,> len<26>
## 2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: ind<6>
elem<CN=www.juniper.net,OU=Network,O=support,L=california,ST=ca,C=US,Email=juniper@juniper.net,>len<90>
>
## 2011-05-17 00:00:13 : normalize_user_wildcard_dn_id: ind<-1>
elem<CN=www.juniper.net,OU=Network,O=support,L=california,ST=ca,C=US,Email=juniper@juniper.net,DC=,>len<94>
## 2011-05-17 00:00:13 : normalize_user_wildcard_dn_id:
result<CN=www.juniper.net,OU=Network,O=support,L=california,ST=ca,C=US,Email=juniper@juniper.net,DC=,>len<94>ret<0>
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > ct:CN=www.juniper.net
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > ct:OU=Network
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > ct:O=support
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > ct:L=california
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > ct:ST=ca
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > ct:C=US
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > ct:Email=juniper@juniper.net
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > count_num_required_elems: ret num elem<7>. (7 fields configured
on the ike user "test" are matched with 7 field configured on the NCP local certificate)
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > no container identity requirement.
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > wild card identity
matched<CN=www.juniper.net,OU=Network,O=support,L=california,ST=ca,C=US,Email=juniper@juniper.net,DC=
,>.
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > ID match found. ( Match is successful for user test)
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > user id found<1>.
## 2011-05-17 00:00:13 : IKE<172.27.199.207> Found peer entry (NCP_VPN) from 172.27.199.207.
## 2011-05-17 00:00:13 : responder create sa: 172.27.199.207->172.27.201.141
## 2011-05-17 00:00:13 : init p1sa, pidt = 0x0
## 2011-05-17 00:00:13 : change peer identity for p1 sa, pidt = 0x0
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > peer_identity_create_with_uid: uid<0>
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > create peer identity 0x143f4e84
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > peer_identity_add_to_peer: num entry before add <1>
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > peer_identity_add_to_peer: num entry after add <2>
## 2011-05-17 00:00:13 : peer identity 143f4e84 created.
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > EDIPI disabled
## 2011-05-17 00:00:13 : IKE<172.27.199.207> getProfileFromP1Proposal->
## 2011-05-17 00:00:13 : IKE<172.27.199.207> find profile[0]=<00000005 00000002 00000003 00000002> for p1
proposal (id 11), xauth(0)
## 2011-05-17 00:00:13 : IKE<172.27.199.207> responder create sa: 172.27.199.207->172.27.201.141
## 2011-05-17 00:00:13 : IKE<172.27.199.207> Phase 1: Responder starts AGGRESSIVE mode negotiations.
## 2011-05-17 00:00:13 : IKE<172.27.199.207> AG in state OAK_AG_NOSTATE.
## 2011-05-17 00:00:13 : IKE<172.27.199.207> Process [VID]:
## 2011-05-17 00:00:13 : IKE<172.27.199.207 > Vendor ID:
## 2011-05-17 00:00:13 : da 8e 93 78 80 01 00 00
## 2011-05-17 00:00:13 : IKE<172.27.199.207> receive unknown vendor ID payload
```

2011-05-17 00:00:13 : IKE<172.27.199.207> Process [VID]:
2011-05-17 00:00:13 : IKE<172.27.199.207 > Vendor ID:
2011-05-17 00:00:13 : 7d 94 19 a6 53 10 ca 6f 2c 17 9d 92 15 52 9d 56
2011-05-17 00:00:13 : IKE<172.27.199.207> rcv non-NAT-Traversal VID payload.
2011-05-17 00:00:13 : IKE<172.27.199.207> Process [VID]:
2011-05-17 00:00:13 : IKE<172.27.199.207 > Vendor ID:
2011-05-17 00:00:13 : 90 cb 80 91 3e bb 69 6e 08 63 81 b5 ec 42 7b 1f
2011-05-17 00:00:13 : IKE<172.27.199.207> rcv NAT-Traversal VID payload (draft-ietf-ipsec-nat-t-ike-02).
2011-05-17 00:00:13 : IKE<172.27.199.207> Process [VID]:
2011-05-17 00:00:13 : IKE<172.27.199.207 > Vendor ID:
2011-05-17 00:00:13 : 44 85 15 2d 18 b6 bb cd 0b e8 a8 46 95 79 dd cc
2011-05-17 00:00:13 : IKE<172.27.199.207> rcv NAT-Traversal VID payload (draft-ietf-ipsec-nat-t-ike-00).
2011-05-17 00:00:13 : IKE<172.27.199.207> Process [VID]:
2011-05-17 00:00:13 : IKE<172.27.199.207 > Vendor ID:
2011-05-17 00:00:13 : 4a 13 1c 81 07 03 58 45 5c 57 28 f2 0e 95 45 2f
2011-05-17 00:00:13 : IKE<172.27.199.207> rcv non-NAT-Traversal VID payload.
2011-05-17 00:00:13 : IKE<172.27.199.207> Process [VID]:
2011-05-17 00:00:13 : IKE<172.27.199.207 > Vendor ID:
2011-05-17 00:00:13 : af ca d7 13 68 a1 f1 c9 6b 86 96 fc 77 57 01 00
2011-05-17 00:00:13 : IKE<172.27.199.207> Process [VID]:
2011-05-17 00:00:13 : IKE<172.27.199.207 > Vendor ID:
2011-05-17 00:00:13 : eb 4c 1b 78 8a fd 4a 9c b7 73 0a 68 d5 6d 08 8b
2011-05-17 00:00:13 : IKE<172.27.199.207> rcv non-NAT-Traversal VID payload.
2011-05-17 00:00:13 : IKE<172.27.199.207> Process [VID]:
2011-05-17 00:00:13 : IKE<172.27.199.207 > Vendor ID:
2011-05-17 00:00:13 : c6 1b ac a1 f1 a6 0c c1 08 00 00 00 00 00 00
2011-05-17 00:00:13 : IKE<172.27.199.207> rcv non-NAT-Traversal VID payload.
2011-05-17 00:00:13 : IKE<172.27.199.207> Process [VID]:
2011-05-17 00:00:13 : IKE<172.27.199.207 > Vendor ID:
2011-05-17 00:00:13 : 40 48 b7 d5 6e bc e8 85 25 e7 de 7f 00 d6 c2 d3
2011-05-17 00:00:13 : c0 00 00 00
2011-05-17 00:00:13 : IKE<172.27.199.207> receive unknown vendor ID payload
2011-05-17 00:00:13 : IKE<172.27.199.207> Process [VID]:
2011-05-17 00:00:13 : IKE<172.27.199.207 > Vendor ID:
2011-05-17 00:00:13 : 12 f5 f2 8c 45 71 68 a9 70 2d 9f e2 74 cc 01 00
2011-05-17 00:00:13 : IKE<172.27.199.207> rcv non-NAT-Traversal VID payload.
2011-05-17 00:00:13 : IKE<172.27.199.207> Process [SA]:
2011-05-17 00:00:13 : IKE<172.27.199.207> Proposal received: xauthflag 0 (received phase1 proposals)
2011-05-17 00:00:13 : IKE<172.27.199.207> auth(3)<RSA>, encr(5)<3DES>, hash(2)<SHA>, group(2)
2011-05-17 00:00:13 : IKE<172.27.199.207> xauth attribute: disabled
2011-05-17 00:00:13 : IKE<172.27.199.207> Phase 1 proposal [0] selected.
2011-05-17 00:00:13 : IKE<172.27.199.207> SA Life Type = seconds
2011-05-17 00:00:13 : IKE<172.27.199.207> SA lifetime (TV) = 28800
2011-05-17 00:00:13 : IKE<172.27.199.207> DH_BG_consume OK. p1 resp
2011-05-17 00:00:13 : IKE<172.27.199.207> Process [KE]:
2011-05-17 00:00:13 : IKE<172.27.199.207> processing ISA_KE in phase 1.
2011-05-17 00:00:13 : IKE<172.27.199.207> Phase1: his_DH_pub_len is 128
2011-05-17 00:00:13 : IKE<172.27.199.207> Process [NONCE]:
2011-05-17 00:00:13 : IKE<172.27.199.207> processing NONCE in phase 1.
2011-05-17 00:00:13 : IKE<172.27.199.207> Process [ID]:

2011-05-17 00:00:13 : IKE<172.27.199.207> ID received: type=ID_DER_ASN1_DN, DN =
Email=juniper@juniper.net,CN=www.juniper.net,OU=Network,O=support,L=california,ST=ca,C=US, port = 0,
protocol=0

2011-05-17 00:00:13 : IKE<172.27.199.207> peer gateway entry has no peer id configured

2011-05-17 00:00:13 : IKE<172.27.199.207> ID processed. return 0. sa->p1_state = 0.

2011-05-17 00:00:13 : IKE<172.27.199.207> Process [CERT-REQ.]; (certificate request is being processed)

2011-05-17 00:00:13 : IKE<172.27.199.207> processing ISA_CERT_REQ starts, type=4.

2011-05-17 00:00:13 : IKE<172.27.199.207> process_cert_req done.

2011-05-17 00:00:13 : IKE<172.27.199.207> need to wait for offline p1 DH work done.

2011-05-17 00:00:13 : IKE<172.27.199.207> IKE msg done: PKI state<0> IKE state<0/281210a>

2011-05-17 00:00:13 : IKE<0.0.0.0 > finished job pkaidx <0> dh_len<128> dmax<64>

2011-05-17 00:00:13 : IKE<0.0.0.0 > finished job d<e8f4d8a><a45fb260><883b0e3><7249c901>

2011-05-17 00:00:13 : IKE<172.27.199.207> AG in state OAK_AG_NOSTATE.

2011-05-17 00:00:13 : IKE<172.27.199.207> re-enter AG after offline DH done

2011-05-17 00:00:13 : IKE<172.27.199.207> Phase 1 AG Responder constructing 2nd message.

2011-05-17 00:00:13 : IKE<172.27.199.207> Construct ISAKMP header.

2011-05-17 00:00:13 : IKE<172.27.199.207> Msg header built (next payload #1)

2011-05-17 00:00:13 : IKE<172.27.199.207> Construct [SA] for ISAKMP

2011-05-17 00:00:13 : IKE<172.27.199.207> auth(3)<RSA>, encr(5)<3DES>, hash(2)<SHA>, group(2)

2011-05-17 00:00:13 : IKE<172.27.199.207> xauth attribute: disabled

2011-05-17 00:00:13 : IKE<172.27.199.207> lifetime/lifeseize (28800/0)

2011-05-17 00:00:13 : IKE<0.0.0.0 > set_phase1_transform, dh_group(2).

2011-05-17 00:00:13 : IKE<172.27.199.207> Construct NetScreen [VID]

2011-05-17 00:00:13 : IKE<172.27.199.207> Construct custom [VID]

2011-05-17 00:00:13 : IKE<172.27.199.207> Construct custom [VID]

2011-05-17 00:00:13 : IKE<172.27.199.207> Construct [KE] for ISAKMP

2011-05-17 00:00:13 : IKE<172.27.199.207> Construct [NONCE]

2011-05-17 00:00:13 : IKE<172.27.199.207> gen_skeyid()

2011-05-17 00:00:13 : IKE<172.27.199.207> gen_skeyid: returning 0

2011-05-17 00:00:13 : IKE<172.27.199.207> Construct [ID] for ISAKMP

2011-05-17 00:00:13 : found 1 cert of type 3

2011-05-17 00:00:13 : IKE<172.27.199.207> Use FQDN "www.juniper.net" in local certificate subject
alternative name as IKE p1 ID.

2011-05-17 00:00:13 : IKE<172.27.199.207> Construct [CERT]

2011-05-17 00:00:13 : IKE<172.27.199.207> construct_cert(), first cert.

2011-05-17 00:00:13 : found 1 cert of type 3

2011-05-17 00:00:13 : IKE<172.27.199.207> construct_cert(), cert type = 4, certlen = 1548

2011-05-17 00:00:13 : IKE<172.27.199.207> Direct CA, peer wants X509, will send one X509 cert.

2011-05-17 00:00:13 : IKE<172.27.199.207> one X509 cert

2011-05-17 00:00:13 : IKE<172.27.199.207> responder (pki) constructing remote NAT-D

2011-05-17 00:00:13 : IKE<172.27.199.207> Construct [CERT-REQ]

2011-05-17 00:00:13 : IKE<172.27.199.207> responder (pki) constructing remote NAT-D done

2011-05-17 00:00:13 : IKE<172.27.199.207> Construct [SIG]

2011-05-17 00:00:13 : IKE<172.27.199.207> constructing RSA signature.

**## 2011-05-17 00:00:13 : IKE<172.27.199.207> Use FQDN "www.juniper.net" in local certificate subject
alternative name as IKE p1 ID. (certificate FQDN will be used as the IKE id for phase 1)**

2011-05-17 00:00:13 : IKE<172.27.199.207> ID, len=19, type=2, pro=17, port=500,

2011-05-17 00:00:13 : IKE<172.27.199.207>

2011-05-17 00:00:13 : IKE<172.27.199.207>

2011-05-17 00:00:13 : IKE<172.27.199.207 > digest when construct sig

2011-05-17 00:00:13 : fb c8 a7 60 70 cd ca 1f 5d 79 23 48 a2 bd 7d bd

```
## 2011-05-17 00:00:13 : 9a 05 85 c6 02 00 00 00 68 7d 92 14 f0 00 00 00
## 2011-05-17 00:00:13 : key has type <6>.
## 2011-05-17 00:00:13 : IKE<172.27.199.207> [SIG] sig_len=256 and nbytes=260
## 2011-05-17 00:00:13 : IKE<172.27.199.207> throw packet to the peer, paket_len=2165
## 2011-05-17 00:00:13 : IKE<172.27.199.207 > Xmit : [SA] [VID] [VID] [VID] [KE] [NONCE] [ID] [CERT] [CERT-REQ]
## 2011-05-17 00:00:13 : [SIG]
## 2011-05-17 00:00:13 : IKE<172.27.199.207> Responder sending IPv4 IP 172.27.199.207/port 500
## 2011-05-17 00:00:13 : IKE<172.27.199.207> Send Phase 1 packet (len=2165)
## 2011-05-17 00:00:13 : IKE<172.27.199.207> ike packet, len 1896, action 0
## 2011-05-17 00:00:13 : IKE<172.27.199.207> Catcher: received 1868 bytes from socket.
## 2011-05-17 00:00:13 : IKE<172.27.199.207> ***** Recv packet if <ethernet0/2> of vsys <Root> *****
## 2011-05-17 00:00:13 : IKE<172.27.199.207> Catcher: get 1868 bytes. src port 500
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > ISAKMP msg: len 1868, npx 9[SIG], exch 4[AG], flag 01 E
## 2011-05-17 00:00:13 : IKE<172.27.199.207> Decrypting payload (length 1840)
## 2011-05-17 00:00:13 : IKE<172.27.199.207 > Recv*: [SIG] [CERT] [NOTIF]
## 2011-05-17 00:00:13 : IKE<0.0.0.0 > extract payload (1840):
## 2011-05-17 00:00:13 : IKE<172.27.199.207> AG in state OAK_AG_INIT_EXCH.
## 2011-05-17 00:00:13 : IKE<172.27.199.207> Process [CERT]:
## 2011-05-17 00:00:13 : IKE<172.27.199.207> Processing CERT payload. Cert Type = 4, Cert Length = 1542.
## 2011-05-17 00:00:13 : input len <1547>, pos<0>, len<1547>
## 2011-05-17 00:00:13 : Last CERT, wrap up PKCS7.
## 2011-05-17 00:00:13 : :
Email=juniper@juniper.net,CN=www.juniper.net,OU=Network,O=support,L=california,ST=ca,C=US,
## 2011-05-17 00:00:13 : build_ike_pki_mail: key_type=4
## 2011-05-17 00:00:13 : IKE<172.27.199.207> IKE msg done: PKI state<1> IKE state<5/81111f>
## 2011-05-17 00:00:13 : processPkiRequest cmd=0
## 2011-05-17 00:00:13 : certReqHandler: req=149287fc task=194bbc0
## 2011-05-17 00:00:13 : PKI_CID_VERIFY_CERT_REQ for task 0
## 2011-05-17 00:00:13 : verify_LDAP_p7_Init
## 2011-05-17 00:00:13 : To verify EE cert:
Email=juniper@juniper.net,CN=www.juniper.net,OU=Network,O=support,L=california,ST=ca,C=US,
## 2011-05-17 00:00:13 : certReqHandler:new pLdapState=253db9c size=376
## 2011-05-17 00:00:13 : build_untrust_chain:
## 2011-05-17 00:00:13 : issuer cert found in device.
## 2011-05-17 00:00:13 : Add trusted CA: CN=Lab,DC=contoso,DC=com, (CA certificate is checked)
## 2011-05-17 00:00:13 : x509_validate_proc
## 2011-05-17 00:00:13 : Next CA: CN=Lab,DC=contoso,DC=com,
## 2011-05-17 00:00:13 : ldaplkeInit
## 2011-05-17 00:00:13 : x509_dss_verify
## 2011-05-17 00:00:13 : Verifying cert:
Email=juniper@juniper.net,CN=www.juniper.net,OU=Network,O=support,L=california,ST=ca,C=US,
## 2011-05-17 00:00:13 : Cert issued by: CN=Lab,DC=contoso,DC=com,
## 2011-05-17 00:00:13 : ASN1_verify -> algo_index=65[alg=0x1441ec84, RSA-SHA1] type=1CC1240
## 2011-05-17 00:00:13 : ASN1_verify -> inl=1262
## 2011-05-17 00:00:13 : ASN1_verify -> siglen=256
## 2011-05-17 00:00:13 : EVP_VerifyFinal -> key_type = 6
## 2011-05-17 00:00:13 : RSA_padding_check_PKCS1_type_1: num<256> flen<255> <00000001>
## 2011-05-17 00:00:13 : cmp time input<110514025948Z> to current<110516230013Z>
## 2011-05-17 00:00:13 : cmp time input<150303133143Z> to current<110516230013Z>
## 2011-05-17 00:00:13 : pass the certificate DSS check
## 2011-05-17 00:00:13 : use per CA revocation resource.
```

2011-05-17 00:00:13 : checking for revocation. (CRL check is being done)
2011-05-17 00:00:13 : ldapStart
2011-05-17 00:00:13 : ldapRetrieveCRL
2011-05-17 00:00:13 : cmp time input<110520074204Z> to current<110516230013Z>
2011-05-17 00:00:13 : CRL is not expired and within device refresh requirement.
2011-05-17 00:00:13 : ldapRetrieveCRL: CRL current state =0.
2011-05-17 00:00:13 : check_local_crl: found the CRL in database.
2011-05-17 00:00:13 : ldapStart: exit (0).
2011-05-17 00:00:13 : certReqHandler: next in chain
2011-05-17 00:00:13 : move_to_next_cert_in_chain, cur<0> total<2>
2011-05-17 00:00:13 : move_to_next_cert_in_chain: at end.
2011-05-17 00:00:13 : cert_path_success
2011-05-17 00:00:13 : Top of chain verified ok:
Email=juniper@juniper.net,CN=www.juniper.net,OU=Network,O=support,L=california,ST=ca,C=US,
2011-05-17 00:00:13 : cert verified ok:
Email=juniper@juniper.net,CN=www.juniper.net,OU=Network,O=support,L=california,ST=ca,C=US,
2011-05-17 00:00:13 : ldap_reply_cert_verify_ok:
2011-05-17 00:00:13 : ldap_reply_cert_verify_ok: public key len=270
2011-05-17 00:00:13 : ldap_reply_cert_verify_ok: public key len=270 cert_len=1542
2011-05-17 00:00:13 : build_ike_pki_mail: key_type=6
2011-05-17 00:00:13 : pki mail received.
2011-05-17 00:00:13 : IKE<0.0.0.0 > I got hit by mail. 1
2011-05-17 00:00:13 : IKE<0.0.0.0 > message from PKI, msg id=f001
2011-05-17 00:00:13 : message from PKI, msg id=f001
2011-05-17 00:00:13 : IKE<172.27.199.207> enter PKI_CID_VERIFY_CERT_RSP
2011-05-17 00:00:13 : IKE<172.27.199.207> AG in state OAK_AG_INIT_EXCH.
2011-05-17 00:00:13 : IKE<172.27.199.207> Process [CERT]:
2011-05-17 00:00:13 : IKE<172.27.199.207> Processing CERT payload. Cert Type = 4, Cert Length = 1542.
2011-05-17 00:00:13 : IKE<172.27.199.207> in cert, name
<Email=juniper@juniper.net,CN=www.juniper.net,OU=Network,O=support,L=california,ST=ca,C=US>
2011-05-17 00:00:13 : IKE<172.27.199.207> recv cert with IP(0.0.0.0), FQDN(none), RFC822(none)
2011-05-17 00:00:13 : IKE<0.0.0.0 > Cert NotAfter=Mar 3 13:31:43 2015 GMT
2011-05-17 00:00:13 : IKE<172.27.199.207> Cert_time(573312703) current(453513613)
2011-05-17 00:00:13 : IKE<172.27.199.207> Process [SIG]:
2011-05-17 00:00:13 : IKE<172.27.199.207> processing xSA_SIG. SIG_TYPE=3 (RSA)
2011-05-17 00:00:13 : IKE<172.27.199.207> ***** Got public key for 172.27.199.207 *****
2011-05-17 00:00:13 : IKE<172.27.199.207> processing RSA sig
2011-05-17 00:00:13 : IKE<172.27.199.207> ID, len=152, type=9, pro=0, port=0,
2011-05-17 00:00:13 : IKE<172.27.199.207>
2011-05-17 00:00:13 : IKE<172.27.199.207 > his_digest
2011-05-17 00:00:13 : aa 35 a4 44 61 14 f8 fc fb d6 c0 54 5a 58 f1 c9
2011-05-17 00:00:13 : 25 6f 3b 25 00 00 00 00 89 0b 41 00 50 f1 cc 03
2011-05-17 00:00:13 : RSA_padding_check_PKCS1_type_1: num<256> flen<255> <00000001>
2011-05-17 00:00:13 : IKE<172.27.199.207> Process [NOTIF]:
2011-05-17 00:00:13 : IKE<172.27.199.207> Received notify message for DOI <1> <24578> <INITIAL-CONTACT>.
2011-05-17 00:00:13 : IKE<172.27.199.207> Received initial contact notification and removed Phase 2 SAs.
2011-05-17 00:00:13 : clear phase 2 sa of peer NCP_VPN.
2011-05-17 00:00:13 : IKE<172.27.199.207> deactive p2 sa 3 send_delete 0
2011-05-17 00:00:13 : IKE<172.27.199.207> process notify exit with <0>.
2011-05-17 00:00:13 : IKE<172.27.199.207> pki_msg: pki state<0>ike state<6/81113f>

```
## 2011-05-17 00:00:13 : IKE<172.27.199.207> completing Phase 1
## 2011-05-17 00:00:13 : IKE<172.27.199.207> sa_pidt = 143f4e84
## 2011-05-17 00:00:13 : IKE<172.27.199.207> found existing peer identity 143f4bd0
## 2011-05-17 00:00:13 : IKE<172.27.199.207> peer_identity_unregister_p1_sa.
## 2011-05-17 00:00:13 : IKE<0.0.0.0    > delete peer identity 0x143f4e84
## 2011-05-17 00:00:13 : IKE<0.0.0.0    > peer_identity_remove_from_peer: num entry before remove <2>
## 2011-05-17 00:00:13 : IKE<172.27.199.207> peer_idt.c peer_identity_unregister_p1_sa 685: pidt deleted.
## 2011-05-17 00:00:13 : IKE<172.27.199.207> phase 1 sa timeout value reduced <28666> to <30>.
## 2011-05-17 00:00:13 : IKE<172.27.199.207> Phase 1: Completed for ip <172.27.199.207>, user<test>
## 2011-05-17 00:00:13 : IKE<172.27.199.207> Phase 1: Completed Aggressive mode negotiation with a
<28800>-second lifetime. (Phase 1 completed)
## 2011-05-17 00:00:13 : IKE<172.27.199.207> AG start phase 2 in 1000 ms
## 2011-05-17 00:00:14 : IKE<172.27.199.207> ike packet, len 200, action 0
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Catcher: received 172 bytes from socket.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> ***** Recv packet if <ethernet0/2> of vsys <Root> *****
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Catcher: get 172 bytes. src port 500
## 2011-05-17 00:00:14 : IKE<0.0.0.0    > ISAKMP msg: len 172, nxp 8[HASH], exch 32[QM], flag 01 E
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Create conn entry...
## 2011-05-17 00:00:14 : IKE<172.27.199.207> ...done(new d54d03a4)
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Phase 2 msg-id <a4034dd5>: Responded to the first peer
message.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Decrypting payload (length 144)
## 2011-05-17 00:00:14 : IKE<172.27.199.207 > Recv*: [HASH] [SA] [NONCE] [ID] [ID]
## 2011-05-17 00:00:14 : valid id checking, id type:IP Address, len:12.
## 2011-05-17 00:00:14 : valid id checking, id type:IP Subnet, len:16.
## 2011-05-17 00:00:14 : IKE<0.0.0.0    > extract payload (144):
## 2011-05-17 00:00:14 : valid id checking, id type:IP Address, len:12.
## 2011-05-17 00:00:14 : valid id checking, id type:IP Subnet, len:16.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> QM in state OAK_QM_SA_ACCEPT.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> receive init proxy id type ID_IPV4_ADDR with mask 0: force mask
to all 1.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Start by finding matching member SA (verify -1/-1)
## 2011-05-17 00:00:14 : IKE<172.27.199.207> IKE: Matching policy: gw ip <172.27.199.207> peer entry id<4>
## 2011-05-17 00:00:14 : IKE<0.0.0.0    > protocol matched expected<0>.
## 2011-05-17 00:00:14 : IKE<0.0.0.0    > port matched expect l:<0>, r<0>.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Peer is dial up.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> configured ID for sa(9):
## 2011-05-17 00:00:14 : IKE<172.27.199.207> local 192.168.1.0/24 prot<0> port<0> type<4>
remote 172.27.199.207/32 prot<0> port<0> type<1>
## 2011-05-17 00:00:14 : IKE<172.27.199.207> member without dynamic policy found, match local address only
## 2011-05-17 00:00:14 : ipvx = IPV4
## 2011-05-17 00:00:14 : rcv_local_addr = 192.168.1.0, rcv_local_mask = 255.255.255.0, p_rcv_local_real =
192.168.1.0
## 2011-05-17 00:00:14 : rcv_remote_addr = 172.27.199.207, rcv_remote_mask = 255.255.255.255,
p_rcv_remote_real = 172.27.199.207
## 2011-05-17 00:00:14 : ike_p2_id->local_ip = 192.168.1.0, cfg_local_mask = 255.255.255.0, p_cfg_local_real =
192.168.1.0
## 2011-05-17 00:00:14 : ike_p2_id->remote_ip = 172.27.199.207, cfg_remote_mask = 255.255.255.255,
p_cfg_remote_real = 172.27.199.207
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Proxy ID match: Located matching Phase 2 SA <9>.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Start by finding matching member SA (verify -1/-1)
```

```
## 2011-05-17 00:00:14 : IKE<172.27.199.207> IKE: Matching policy: gw ip <172.27.199.207> peer entry id<4>
## 2011-05-17 00:00:14 : IKE<0.0.0.0 > protocol matched expected<0>.
## 2011-05-17 00:00:14 : IKE<0.0.0.0 > port matched expect l:<0>, r<0>.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Peer is dial up.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> configured ID for sa(9):
## 2011-05-17 00:00:14 : IKE<172.27.199.207> local 192.168.1.0/24 prot<0> port<0> type<4>
    remote 172.27.199.207/32 prot<0> port<0> type<1> (checking the proxy id)
## 2011-05-17 00:00:14 : IKE<172.27.199.207> member without dynamic policy found, match local address only
## 2011-05-17 00:00:14 : ipvx = IPV4
## 2011-05-17 00:00:14 : rcv_local_addr = 192.168.1.0, rcv_local_mask = 255.255.255.0, p_rcv_local_real =
192.168.1.0
## 2011-05-17 00:00:14 : rcv_remote_addr = 172.27.199.207, rcv_remote_mask = 255.255.255.255,
p_rcv_remote_real = 172.27.199.207
## 2011-05-17 00:00:14 : ike_p2_id->local_ip = 192.168.1.0, cfg_local_mask = 255.255.255.0, p_cfg_local_real =
192.168.1.0
## 2011-05-17 00:00:14 : ike_p2_id->remote_ip = 172.27.199.207, cfg_remote_mask = 255.255.255.255,
p_cfg_remote_real = 172.27.199.207
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Proxy ID match: Located matching Phase 2 SA <9>.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Process [SA]:
## 2011-05-17 00:00:14 : IKE<0.0.0.0 > Check P2 Proposal (checking phase 2 proposals)
## 2011-05-17 00:00:14 : IKE<172.27.199.207> SA life type = seconds
## 2011-05-17 00:00:14 : IKE<0.0.0.0 > SA life duration (TV) = 28800
## 2011-05-17 00:00:14 : IKE<0.0.0.0 > encap mode from peer = 1.
## 2011-05-17 00:00:14 : IKE<0.0.0.0 > encap mode after converting it to private value = 1.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Phase 2 received:
## 2011-05-17 00:00:14 : IKE<172.27.199.207> atts<00000003 00000000 00000003 00000002 00000001
00000000>
## 2011-05-17 00:00:14 : IKE<172.27.199.207> proto(3)<ESP>, esp(3)<ESP_3DES>, auth(2)<SHA>,
encap(1)<TUNNEL>, group(0)
## 2011-05-17 00:00:14 : IKE<172.27.199.207> P2 proposal [0] selected.
## 2011-05-17 00:00:14 : IKE<0.0.0.0 > add sa list for msg id <a4034dd5>
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Process [NONCE]:
## 2011-05-17 00:00:14 : IKE<172.27.199.207> processing NONCE in phase 2.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Process [ID]:
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Process [ID]:
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Phase 2 Responder constructing 2nd message.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Construct ISAKMP header.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Msg header built (next payload #8)
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Construct [HASH]
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Construct [SA] for IPSEC
## 2011-05-17 00:00:14 : IKE<0.0.0.0 > Set IPSEC SA attrs tunnel(1) SHA-1 grp0 lifetime(28800/0)
## 2011-05-17 00:00:14 : IKE<0.0.0.0 > Before NAT-T attr unmap: P2 prop tunnel = 1.
## 2011-05-17 00:00:14 : IKE<0.0.0.0 > After NAT-T attr unmap: P2 prop tunnel = 1.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> IP<172.27.199.207> mask<255.255.255.255> prot<0> port<0>
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Initiator P2 ID built: ....
## 2011-05-17 00:00:14 : IKE<192.168.1.0> IP<192.168.1.0> mask<255.255.255.0> prot<0> port<0>
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Responder P2 ID built: ....
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Construct [NONCE] for IPsec
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Construct [ID] for Phase 2
## 2011-05-17 00:00:14 : id payload constructed. type(1),ip(172.27.199.207),mask(255.255.255.255), prot(0),
port(0)
```

```
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Construct [ID] for Phase 2
## 2011-05-17 00:00:14 : id payload constructed. type(4),ip(192.168.1.0),mask(255.255.255.0), prot(0), port(0)
## 2011-05-17 00:00:14 : IKE<172.27.199.207> send out RESPONDER_LIFETIME notification. prot=3,
## 2011-05-17 00:00:14 : IKE<172.27.199.207> life_sec=3600
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Construct [NOTIF] (RESPONDER-LIFETIME) for IPSEC
## 2011-05-17 00:00:14 : IKE<172.27.199.207> construct QM HASH
## 2011-05-17 00:00:14 : IKE<172.27.199.207 > Xmit*: [HASH] [SA] [NONCE] [ID] [ID] [NOTIF]
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Encrypt P2 payload (len 196)
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Responder sending IPv4 IP 172.27.199.207/port 500
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Send Phase 2 packet (len=204)
## 2011-05-17 00:00:14 : IKE<172.27.199.207> oakley_process_quick_mode():exit
## 2011-05-17 00:00:14 : IKE<0.0.0.0 > proc_other_session_notify->
## 2011-05-17 00:00:14 : IKE<0.0.0.0 > process Notify Payload: doi(1), msg(24578), txt<INITIAL-CONTACT>
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Received initial contact notification and removed Phase 1 SAs.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> IKE msg done: PKI state<0> IKE state<6/81113f>
## 2011-05-17 00:00:14 : IKE<172.27.199.207> ike packet, len 80, action 0
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Catcher: received 52 bytes from socket.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> ***** Recv packet if <ethernet0/2> of vsys <Root> *****
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Catcher: get 52 bytes. src port 500
## 2011-05-17 00:00:14 : IKE<0.0.0.0 > ISAKMP msg: len 52, npx 8[HASH], exch 32[QM], flag 01 E
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Decrypting payload (length 24)
## 2011-05-17 00:00:14 : IKE<172.27.199.207 > Recv*: [HASH]
## 2011-05-17 00:00:14 : IKE<0.0.0.0 > extract payload (24):
## 2011-05-17 00:00:14 : IKE<172.27.199.207> QM in state OAK_QM_AUTH_AWAIT.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> xauth_cleanup()
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Done cleaning up IKE Phase 1 SA
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Start by finding matching member SA (verify 3/3)
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Verify sa: index 3
## 2011-05-17 00:00:14 : IKE<172.27.199.207> IKE: Matching policy: gw ip <172.27.199.207> peer entry id<4>
## 2011-05-17 00:00:14 : IKE<0.0.0.0 > protocol matched expected<0>.
## 2011-05-17 00:00:14 : IKE<0.0.0.0 > port matched expect l:<0>, r<0>.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Peer is dial up.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> configured ID for sa(9):
## 2011-05-17 00:00:14 : IKE<172.27.199.207> local 192.168.1.0/24 prot<0> port<0> type<4>
    remote 172.27.199.207/32 prot<0> port<0> type<1>
## 2011-05-17 00:00:14 : IKE<172.27.199.207> member without dynamic policy found, match local address only
## 2011-05-17 00:00:14 : ipvx = IPV4
## 2011-05-17 00:00:14 : rcv_local_addr = 192.168.1.0, rcv_local_mask = 255.255.255.0, p_rcv_local_real =
192.168.1.0
## 2011-05-17 00:00:14 : rcv_remote_addr = 172.27.199.207, rcv_remote_mask = 255.255.255.255,
p_rcv_remote_real = 172.27.199.207
## 2011-05-17 00:00:14 : ike_p2_id->local_ip = 192.168.1.0, cfg_local_mask = 255.255.255.0, p_cfg_local_real =
192.168.1.0
## 2011-05-17 00:00:14 : ike_p2_id->remote_ip = 172.27.199.207, cfg_remote_mask = 255.255.255.255,
p_cfg_remote_real = 172.27.199.207
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Proxy ID match: Located matching Phase 2 SA <9>.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> sa ID for phase 2 sa is <9>. IP version is 4.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Single user entry.
## 2011-05-17 00:00:14 : IKE<0.0.0.0 > life (sec or kb): lcl 3600, peer 28800, set 3600.
## 2011-05-17 00:00:14 : IKE<0.0.0.0 > life (sec or kb): lcl 0, peer 0, set 0.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> gen_qm_key()
```

```

## 2011-05-17 00:00:14 : IKE<172.27.199.207> load_sa_keys(): enter.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> gen_qm_key()
## 2011-05-17 00:00:14 : IKE<172.27.199.207> load_sa_keys(): enter.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> ikmpd.c 3911. sa ID for phase 2 sa is <9>. IP version is 4.
## 2011-05-17 00:00:14 : IKE<0.0.0.0    > spi hash node removed: type<2>,spi<2068bb0f>,ip<172.27.201.141>
## 2011-05-17 00:00:14 : IKE<0.0.0.0    > spi hash node removed: type<2>,spi<d15cebb0>,ip<172.27.199.207>
## 2011-05-17 00:00:14 : IKE<172.27.199.207> clean_all_sa_state_node_from_list->
## 2011-05-17 00:00:14 : IKE<172.27.199.207> no relocate earlier SA-state, not active.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> key_modify: sa index <3> bk_idx <3>.
## 2011-05-17 00:00:14 : IKE<0.0.0.0    > insert_sa_state_to_spi_hash spi<2068bb10>, sa_index<3>, Incoming
## 2011-05-17 00:00:14 : IKE<0.0.0.0    > insert_sa_state_to_spi_hash spi<d570e295>, sa_index<3>, Outgoing
## 2011-05-17 00:00:14 : IKE<172.27.199.207> update acvpn flags for sa 3
## 2011-05-17 00:00:14 : IKE<172.27.199.207> update acvpn flags for sa 3 - 0x400033
## 2011-05-17 00:00:14 : IKE<172.27.199.207>
crypto_ctx 22, 8, 24, 8, 0, 0, 16, 0, 12, 48
## 2011-05-17 00:00:14 : IKE<172.27.199.207> modify esp tunnel: src (peer) ipv4 <172.27.199.207>
## 2011-05-17 00:00:14 : IKE<172.27.199.207> modifying esp tunnel: self <ipv4 172.27.201.141>
## 2011-05-17 00:00:14 : IKE<172.27.199.207> update auto NHTB status for sa 3
## 2011-05-17 00:00:14 : IKE<172.27.199.207> after mod, out nsptunnel <08443980>.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Phase 2 msg-id <a4034dd5>: Completed Quick Mode
negotiation with SPI <2068bb10>, tunnel ID <9>, and lifetime <3600> seconds/<0> KB. (phase 2 completed)
## 2011-05-17 00:00:14 : IKE<172.27.199.207> Application sa installed.
## 2011-05-17 00:00:14 : IKE<172.27.199.207> oakley_process_quick_mode():exit
## 2011-05-17 00:00:14 : IKE<172.27.199.207> IKE msg done: PKI state<0> IKE state<6/81113f>
## 2011-05-17 00:00:15 : IKE<172.27.199.207> entering ag_begin_qm()
## 2011-05-17 00:00:15 : IKE<172.27.199.207> ike state 6/81113f
## 2011-05-17 00:00:15 : IKE<172.27.201.140> nhtb_list_update_status: vpn 172.27.201.140
## 2011-05-17 00:00:15 : IKE<172.27.201.140> ** link ready return 8
## 2011-05-17 00:00:15 : IKE<172.27.201.140> sa_link_status_for_tunl_ifp: saidx 0, preliminary status 8
## 2011-05-17 00:00:15 : IKE<172.27.201.140> ** link ready return 8
## 2011-05-17 00:00:15 : IKE<172.27.201.140> sa_link_status_for_tunl_ifp: saidx 0, preliminary status 8
## 2011-05-17 00:00:16 : IKE<0.0.0.0    > finished job pkaidx <0> dh_len<128> dmax<64>
## 2011-05-17 00:00:16 : IKE<0.0.0.0    > finished job d<a11a8cde><39d15864><663fea39><28db8341>
## 2011-05-17 00:00:16 : IKE<0.0.0.0    > BN, top32 dmax64 zero<no>

```

Note

The above configurations are an example, It might change according to customer's requirement