ISP/NSP Security Workshop, SANOG 6 16-23 July, 2005 Thimphu, Bhutan

Part 3: Using IPSec between Cisco Router and Unix Host

Meriek Kaeo

IPsec Lab Cisco Configuration

STEP 1 Configure the IKE Phase 1 Policy (ISAKMP Policy)

Note: Useful command is:

```
Router(config)#crypto isakmp ?
  aggressive-mode      Disable ISAKMP aggressive mode
  client               Set client configuration policy
  enable               Enable ISAKMP
  identity             Set the identity which ISAKMP will use
  invalid-spi-recovery Initiate IKE and send Invalid SPI Notify
  keepalive            Set a keepalive interval for use with IOS peers
  key                  Set pre-shared key for remote peer
  nat                  Set a nat keepalive interval for use with IOS peers
  peer                 Set Peer Policy
  policy               Set policy for an ISAKMP protection suite
  profile              Define ISAKMP Profiles
  xauth                Set Extended Authentication values
```

Cisco literature refers to IKE Phase 1 as the ISAKMP policy.
It is configured using the command:

```
crypto isakmp policy <priority #>
```

Multiple policies can be configured and the priority number, which ranges from 1 to 10,000, denotes the order of preference that a given policy will be negotiated with an ISAKMP peer. The lower value has the higher priority. Once in the ISAKMP configuration mode, denoted by the 'Router(config-isakmp)#' prompt, the following parameters can be specified:

- Encryption Algorithm
- Hash Algorithm
- Authentication Method
- Group Lifetime

Note: The following command is useful for helping in configuration

```
Router(config-isakmp)#?
ISAKMP commands:
  authentication  Set authentication method for protection suite
  default         Set a command to its defaults
  encryption      Set encryption algorithm for protection suite
  exit            Exit from ISAKMP protection suite configuration mode
```

```
group           Set the Diffie-Hellman group
hash            Set hash algorithm for protection suite
lifetime        Set lifetime for ISAKMP security association
no          Negate a command or set its defaults
```

STEP 2 Set the ISAKMP Identity

The ISAKMP identity specifies how the IKE Phase 1 peer is identified, which can be either by IP address or host name.

The command to use is:

```
crypto isakmp identity {address | hostname | dn }
```

By default, a peer's ISAKMP identity is the peer's IP address. If you decide to change the default just keep in mind that it is best to always be consistent across your entire IPsec-protected network in the way you choose to define a peer's identity.

STEP 3 Configure the IPsec AH and ESP Parameters

Useful command to is :

```
Router20(config)#crypto ipsec ?
  client                  Configure a client
  df-bit                  Handling of encapsulated DF bit.
  fragmentation           Handling of fragmentation of near-MTU sized packets
  nat-transparency        IPsec NAT transparency model
  optional                Enable optional encryption for IPSec
  profile                 Configure an ipsec policy profile
  security-association    Security association parameters
  transform-set           Define transform and settings
```

The AH and ESP parameters are configured with the following commands:

```
crypto ipsec transform-set transform-set-name <transform 1>
<transform 2>
mode [tunnel | transport]
crypto ipsec security-association lifetime seconds seconds
```

The following options are possible for the transforms:

```
Router20(config)#crypto ipsec transform-set test ?
  ah-md5-hmac       AH-HMAC-MD5 transform
```

ah-sha-hmac	AH-HMAC-SHA transform
comp-lzs	IP Compression using the LZS compression algorithm
esp-3des	ESP transform using 3DES(EDE) cipher (168 bits)
esp-aes	ESP transform using AES cipher
esp-des	ESP transform using DES cipher (56 bits)
esp-md5-hmac	ESP transform using HMAC-MD5 auth
esp-null	ESP transform w/o cipher
esp-seal	ESP transform using SEAL cipher (160 bits)
esp-sha-hmac	ESP transform using HMAC-SHA auth

Note that right after you configure the transforms you get directed to configure further transform-specific parameters:

Router20(cfg-crypto-trans)#?
Crypto transform configuration commands:

default	Set a command to its defaults
exit	Exit from crypto transform configuration mode
mode	encapsulation mode (transport/tunnel)
no	Negate a command or set its defaults

STEP 4 Configure the IPsec Traffic Selectors

The traffic selectors are configured by defining extended access-lists. The permit keyword causes all IP traffic that matches the specified conditions to be protected by IPsec. Note that both directions must be specified

```
access-list 121 permit tcp host 192.168.10.1 host
192.168.10.254 eq telnet
access-list 121 permit tcp host 192.168.10.254 host
192.168.10.1 eq telnet
```

STEP 5 Configure the IKE Phase 2 (IPsec SA) Policy

This step sets up a crypto map which specifies all the necessary parameters to negotiate the IPsec SA policy. The following commands are required:

```
crypto map <crypto-map-name> <seq-num> ipsec-isakmp
match address <access-list-id>
set peer [IP address | hostname]
set transform-set <transform-set-name>
set security-association lifetime seconds <seconds>
set pfs [group1 | group 2]
```

Note: Useful command is:

Router(config)#crypto map LINUXTELNET ?

<1-65535>	Sequence to insert into crypto map entry

```
  client              Specify client configuration settings
  isakmp              Specify isakmp configuration settings
  isakmp-profile      Specify isakmp profile to use
  local-address       Interface to use for local address for this crypto map
  redundancy          High availability options for this map
```

After entering a sequence number you can continue configuring:

```
Router(config)#crypto map LINUXTELNET 5 ?
  ipsec-isakmp     IPSEC w/ISAKMP
  ipsec-manual     IPSEC w/manual keying
  <cr>
```

Note that more complex crypto maps can be configured. But that is for more sophisticated labs ☺

```
Router20(config)#crypto map LINUXTELNET 5 ipsec-isakmp ?
  dynamic     Enable dynamic crypto map support
  profile     Enable crypto map as a crypto-profile
  <cr>
```

So, when we hit <cr> we get a warning to remind us to configure a peer and valid access-list with this crypto map.

```
Router20(config)#crypto map LINUXTELNET 5 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
        and a valid access list have been configured.
```

```
Router20(config-crypto-map)#?
Crypto Map configuration commands:
  default          Set a command to its defaults
  description      Description of the crypto map statement policy
  exit             Exit from crypto map configuration mode
  match            Match values.
  no               Negate a command or set its defaults
  qos              Quality of Service related commands
  reverse-route    Reverse Route Injection.
  set              Set values for encryption/decryption
```

STEP 6 Apply the IPsec Policy to an Interface

The configured crypto map is then applied to the appropriate interface using the ‘crypto map crypto-map-name’ command.

```
Router(config)#interface FastEthernet 0/1
Router(config-if)#crypto map <crypto map name>
```

The resulting configuration should be something like this:

```
!
crypto isakmp policy 2
 encr 3des
 authentication pre-share
 group 2
 lifetime 28800
!
!
crypto ipsec transform-set CISCO2LINUX esp-3des esp-sha-hmac
 mode transport
!
crypto map LINUXTELNET 5 ipsec-isakmp
 ! Incomplete
 set peer 192.168.10.1
 set transform-set CISCO2LINUX
 set pfs group2
 match address 121
!
interface FastEthernet0/1
 description Link to Backbone Two
 ip address 192.168.10.254 255.255.255.0
 duplex auto
 speed auto
 crypto map LINUXTELNET
!
access-list 121 permit tcp host 192.168.10.254 host 192.168.10.1 eq telnet
access-list 121 permit tcp host 192.168.10.1 host 192.168.10.254 eq telnet
```

Useful debugging commands:

```
Router20# debug crypto engine ?
  accelerator  accelerator debug
  error        Crypto Engine Errors
  packet       Crypto Engine Packets
  <cr>

Router# sh crypto ?
  ca             Show certification authority policy
  call           Show crypto call admission info
```

```
  debug-condition  Debug Condition filters
  dynamic-map      Crypto map templates
  eli              Encryption Layer Interface
  engine           Show crypto engine info
  identity         Show crypto identity list
  ipsec            Show IPSEC policy
  isakmp           Show ISAKMP
  key              Show long term public keys
  map              Crypto maps
  mib              Show Crypto-related MIB Parameters
  optional         Optional Encryption Status
  pki              Show PKI
  session          Show crypto sessions (tunnels)
  sockets          Secure Socket Information


Router# sh crypto isakmp ?
  key      Show ISAKMP preshared keys
  peers    Show ISAKMP peer structures
  policy   Show ISAKMP protection suite policy
  profile  Show ISAKMP profiles
  sa       Show ISAKMP Security Associations


Router20# sh crypto ipsec ?
  client               Show Client Status
  policy               Show IPSEC client policies
  profile              Show ipsec profile information
  sa                   IPSEC SA table
  security-association  Show parameters for IPSec security associations
  transform-set        Crypto transform sets
```

Lab: IPsec configuration

Goal: Trying to protect Telnet traffic using IPsec between the LINUX machine and the Cisco router

Use the following Policy specifications for your IKE setup:

IKE Phase 1:
Main Mode
3DES
Sha-1
Diffie-Hellman group2 (1024 bits)
Pre-shared key 'hr5xb8416aa9rb'
SA lifetime 28800 seconds

IKE Phase 2:
3DES
Sha-1
ESP transport mode
Diffie-Hellman group2 (1024 bits)
PFS
SA lifetime 3600 seconds

On the LINUX Machine:

Type command 'man 8 racoon'
Read how to set-up raccoon, the name for this particular IKE software

Type command 'man setkey'
This command is used to set up the SA database

The following files are located in /etc/raccoon:
psk.txt – file which contains the shared secrets
raccoon.conf – file which configures IKE phase 1 and IKE phase 2 parameters

Create a file named 'ipsec.conf' which will be used with setkey to establish the correct security associations. The file should have the following information:

```
flush;
spdflush;
spdadd 192.168.10.1/32 192.168.10.254/32[23] tcp -P out ipsec
esp/transport//require ;
spdadd 192.168.10.254/32[23] 192.168.10.1 any -P in ipsec
esp/transport//require ;
```

Test to see what happens when you try and create an SA database:

Type the following: `setkey –f /etc/racoon/ipsec.conf`

Use the `setkey –P –D` command to see if appropriate entries have been created.

Next, edit the psk.txt file to add the router IP address (the IPsec peer you are communicating with) and the pre-shared secret key:

```
# file for pre-shared keys used for IKE authentication
# format is:  'identifier' 'key'
# For example:
#
#  10.1.1.1          flibbertigibbet
#  www.example.com      12345
#  foo@www.example.com  micropachycephalosaurus

<peer IP address>         hr5xb8416aa9r6
```

Since the psk.txt file contains EXTREMELY sensitive information you will want to make sure that the file is appropriately protected:

```
chmod 600 /etc/raccoon/psk.txt
```

Edit the racoon.conf file to include the appropriate IKE Phase 1 and IKE Phase 2 parameters:

```
# Racoon IKE daemon configuration file.
# See 'man racoon.conf' for a description of the format and
entries.

path include "/etc/racoon";
path pre_shared_key "/etc/racoon/psk.txt";
path certificate "/etc/racoon/certs";

log debug;

remote anonymous
{
     {
             exchange_mode aggressive,main,base;
             lifetime time 8 hour;
             proposal {
                     encryption_algorithm 3des;
                     hash_algorithm sha1;
                     authentication_method pre_shared_key;
                     dh_group 2;
             }
     }


sainfo anonymous
{
     pfs_group 2;
     lifetime time 1 hour ;
     encryption_algorithm 3des, blowfish 448, rijndael ;
     authentication_algorithm hmac_sha1, hmac_md5 ;
     compression_algorithm deflate ;
}
```

Test racoon with the following command:

```
racoon –v –f /etc/racoon/racoon.conf  -l /etc/racoon/test.log
```

The `–l /etc/racoon/test.log` file is used to write any debug information in the event that there are problems. Check to see what the test.log file contains.