Implementation

Aryan Gupta

June 2020

1 Implementing IPSec (using PISKES) in SCION Architecture

There are two ways in which IPSec can be used in SCION.

1. Between an end host and the AS' Path Server(PS), Beacon Server(BS) and Certificate Server(CS)
2. Between border routers of different ASes

In the current SCION implementation, the various components of the AS, i.e., path server, beacon server and certificate servers are setup on a single system, an IPSec tunnel set up between the system on which AS is running and the end host will suffice if we are to use the first method. This would secure intra-AS communication.
The second method would be able to secure inter-AS communication.

The host in a SCION implementation has-

1. **SCION Dispatcher**- This packages the information received from the applications into a SCION Packet, attaching all the necessary headers. Since currently SCION is used as an overlay network on the current Internet implementation, this packet is then packaged into an IP packet and sent to the destination.
2. **SCION Daemon**- This runs as a background program and supplies the necessary information, for e.g., the paths to the destination, PISKES key, the certificates etc.

We want to implement an IPSec tunnel using PISKES key as the preshared key. This would mean deriving the key, using SCION Daemon to fetch it, and use this key as a preshared key.

2 How?

strongSwan is an OpenSource IPsec implementation. It enables easy setting up of an IPSec tunnel between two endpoints. strongSwan provides a suitable range of options for various configurations. It also provides an option to select between IKEv1 and IKEv2. After installing strongSwan, whenever we want to establish a new connection, we basically have to edit two files-

1. **ipsec.conf** - This file contains all the information about the IPSec connection. It involves the name of the connection, the source and destionation IP address, preffered mode of authentication, which cryptographic protocols to use etc.
2. **ipsec.secrets** - It contains the necessary information for the mode of authentication chosen in ipsec.conf file, for e.g., the shared secrets, or the credentials or in our case, the PISKES key.

We can use strongSwan for fulfilling our purpose.

3 Development in scionproto

In scionproto implementation, there is an option to run each AS in a separate docker container. If we install strongSwan in each of the containers, we will be able to configure each of the ASes separately as required.
I have written scripts to automatically setup IPsec tunnels between all the links the border router of one AS has to border routers of other ASes.
Since PISKES keys change after regular intervals, I have used cron to automatically and periodically change the PISKES key in the ipsec.secrets file.
There are 3 scripts-

1. **startall.sh** - This starts all the connections available in the ipsec.conf file.

2. **add.sh** - This script fills the ipsec.conf and ipsec.secrets files according to the information given by "connect.sh".

3. **connect.sh** - This is the main script. It gets all the required information about the connections, gives the command to derive the PISKES keys and sends this information to add.sh. If an argument "establish" is passed to the script, it fills the ipsec.conf file. Since this file does not need to be changed further, this command is used only once. Using this argument also starts cron. After this no argument is passed and it give the commands to rekey the tunnel.

install-strongswan.sh installs strongswan and other files required. It also runs connect.sh script.
Cron is used to periodically change the PISKES keys. It runs connect.sh.
To change the configuration of strongswan, add.sh can be edited as per use.