



PARSHVANATH CHARITABLE TRUST'S  
**A. P. SHAH INSTITUTE OF TECHNOLOGY**

Department of Information Technology

(NBA Accredited)



## Department of Information Technology

Academic Year: 2025-26

Semester: V

Class / Branch: TEIT/Div C

Subject: Security Lab

Name of Instructor: Prof. Vishal Badgujar

Name of Student: Suyash Y Mane

Student ID: 24204008

Date of Performance: 09/10/2025

Date of Submission: 09/10/2025

### Experiment No. 7

**Aim: To study and implement IPSEC in Linux.**

**Screenshots:**

```
apsit@apsit-HP-Pro-Tower-280-G9-E-PCI-Desktop-PC:~$ sudo apt-get update
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:2 https://pkg.jenkins.io/debian-stable binary/ Release [2,044 B]
Get:3 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Hit:4 https://deb.nodesource.com/node_22.x nodistro InRelease
Err:3 https://pkg.jenkins.io/debian-stable binary/ Release.gpg
  The following signatures couldn't be verified because the public key is not available: NO_PUBKEY 5BA31D57EF5975CA
Hit:5 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:6 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Hit:8 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:9 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:10 https://packages.microsoft.com/repos/code stable InRelease
Fetched 129 kB in 3s (40.2 kB/s)
Reading package lists... Done
W: An error occurred during the signature verification. The repository is not up
dated and the previous index files will be used. GPG error: https://pkg.jenkins.
io/debian-stable binary/ Release: The following signatures couldn't be verified
because the public key is not available: NO_PUBKEY 5BA31D57EF5975CA
W: Failed to fetch https://pkg.jenkins.io/debian-stable/binary/Release.gpg The
following signatures couldn't be verified because the public key is not availabl
e: NO_PUBKEY 5BA31D57EF5975CA
W: Some index files failed to download. They have been ignored, or old ones used
```

```
apsit@apsit-HP-Pro-Tower-280-G9-E-PCI-Desktop-PC:~$ sudo apt-get install strongswan strongswan-starter
[sudo] password for apsit:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
strongswan is already the newest version (5.9.5-2ubuntu2.3).
strongswan-starter is already the newest version (5.9.5-2ubuntu2.3).
The following packages were automatically installed and are no longer required:
  gyp libjs-events libjs-highlight.js libjs-inherits libjs-is-typedarray
  libjs-psl libjs-source-map libjs-sprintf.js libjs-typedarray-to-buffer
  libnode-dev libssl-dev libuv1-dev node-abab node-abbrev node-agent-base
  node-ansi-regex node-ansi-styles node-ansistyles node-aproba node-archy
  node-are-we-there-yet node-argparse node-arrify node-asap node-asynckit
  node-balanced-match node-brace-expansion node-builtins node-chalk
  node-chownr node-clean-yaml-object node-cli-table node-clone
  node-color-convert node-color-name node-colors node-columnify
  node-combined-stream node-commander node-console-control-strings
  node-core-util-is node-cssom node-cssstyle node-debug
  node-decompress-response node-defaults node-delayed-stream node-delegates
  node-depd node-diff node-encoding node-end-of-stream node-err-code
  node-escape-string-regexp node-events node-fancy-log node-foreground-child
  node-fs-write-stream-atomic node-fs.realpath node-function-bind node-gauge
  node-get-stream node-glob node-got node-graceful-fs node-growl node-has-flag
```

```
Open  ↵  ipsec.secrets /etc  Save  ≡  -  □  ×
1
2 # This file holds shared secrets or RSA private keys for authentication.
3
4 # RSA private key for this host, authenticating it to any other host
5 # which knows the public part.
6
7 192.168.15.122  192.168.18.104  : PSK "123456"
```

```

Starting StrongSwan 5.9.3 IPsec [starter]...
apsit@apsit-HP-Pro-Tower-280-G9-E-PCI-Desktop-PC:/etc$ sudo ipsec up red-to-blue
initiating IKE_SA red-to-blue[1] to 192.168.18.104
generating IKE_SA_INIT request 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) N(FRAG_SUP) N(HASH_ALG) N(REDIR_SUP) ]
sending packet: from 192.168.15.122[500] to 192.168.18.104[500] (904 bytes)
received packet: from 192.168.18.104[500] to 192.168.15.122[500] (248 bytes)
parsed IKE_SA_INIT response 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) N(FRAG_SUP) N(HASH_ALG) N(CHDLESS_SUP) N(MULT_AUTH) ]
selected proposal: IKE:AES_CBC_128/HMAC_SHA2_256_128/PRF_AES128_XCBC/CURVE_25519
authentication of '192.168.15.122' (myself) with pre-shared key
establishing CHILD_SA red-to-blue[2]
generating IKE_AUTH request 1 [ IDi N(INIT_CONTACT) IDr AUTH SA TSi TSr N(MOBIKE_SUP) N(ADD_4_ADDR) N(ADD_4_ADDR) N(MULT_AUTH) N(EAP_
sending packet: from 192.168.15.122[4500] to 192.168.18.104[4500] (384 bytes)
received packet: from 192.168.18.104[4500] to 192.168.15.122[4500] (224 bytes)
parsed IKE_AUTH response 1 [ IDr AUTH SA TSi TSr N(MOBIKE_SUP) N(ADD_4_ADDR) ]
authentication of '192.168.18.104' with pre-shared key successful
IKE_SA red-to-blue[1] established between 192.168.15.122[192.168.15.122]...192.168.18.104[192.168.18.104]
scheduling reauthentication in 10193s
maximum IKE_SA lifetime 10733s
selected proposal: ESP:AES_CBC_128/HMAC_SHA2_256_128/NO_EXT_SEQ
CHILD_SA red-to-blue[2] established with SPIs c067cc52_i c67778f6_o and TS 192.168.15.122/32 === 192.168.18.104/32
connection 'red-to-blue' established successfully

```

## Conclusion:

in this experiment, we learnt and understood to implement IPSEC in Linux .