

TACACS+ Over IPSec

- Revision History, on page 1
- Feature Description, on page 1
- How it Works, on page 3
- Configuring TACACS+ over IPSec, on page 6
- Monitoring and Troubleshooting, on page 9

Revision History

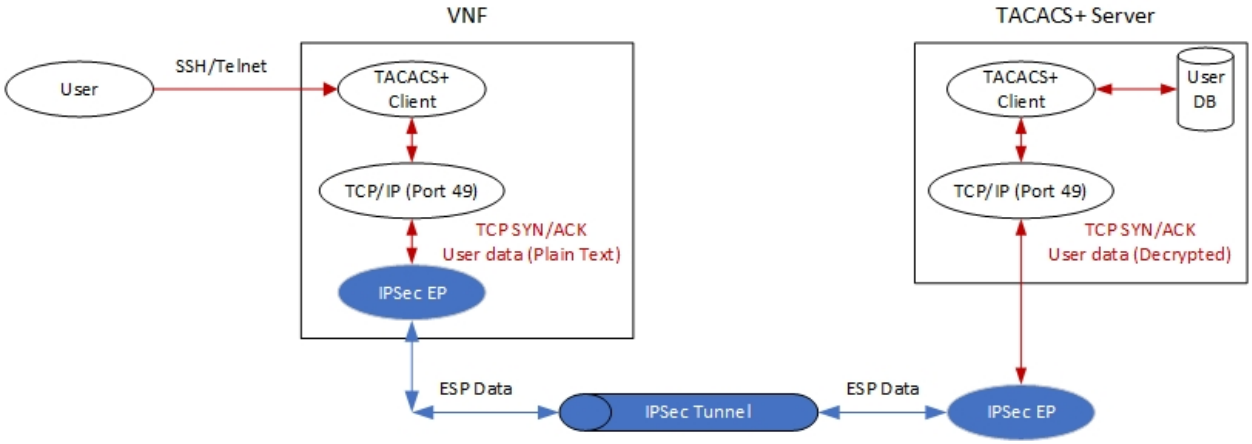
Revision Details	Release
First introduced	21.24

Feature Description

The Terminal Access Controller Access Control Server Plus (TACACS+) is a security protocol that is used for authenticating user access permissions on StarOS. To secure the authentication data that are sent over TACACS+ client and servers, CUPS VNFs support TACACS+ over IPSec for encrypting the authentication data.

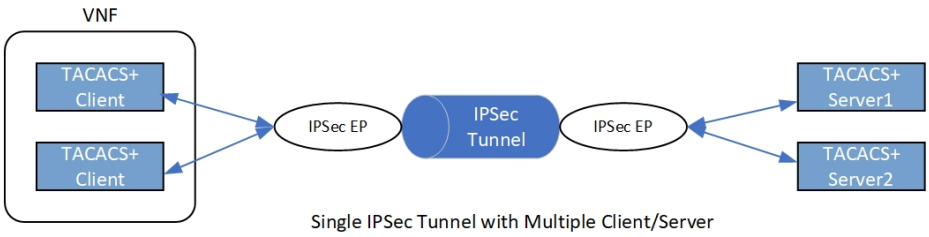
Architecture

The following diagram illustrates a secured TACACS+ architecture.

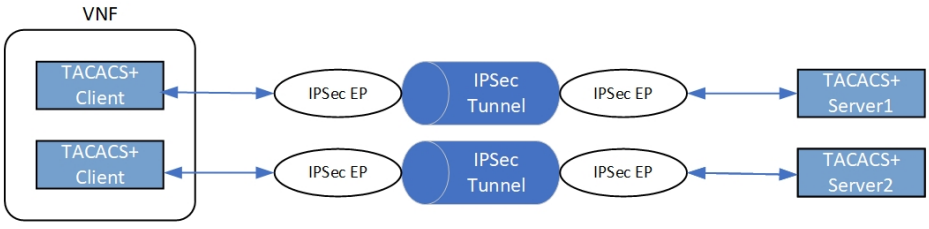


Deployment Architecture

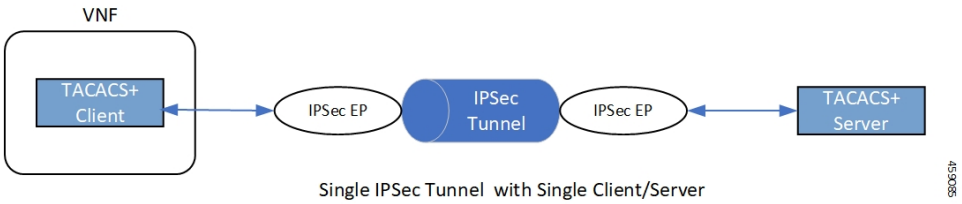
There are multiple ways you can use TACACS+ client/server in a secured way. You can either have single or multiple TACACS+ servers. A single VNF can host single or multiple clients. The TACACS+ over IPSec solution can handle multiple clients on a single VNF.



Single IPSec Tunnel with Multiple Client/Server



Multiple IPSec Tunnel with Multiple Client/Server



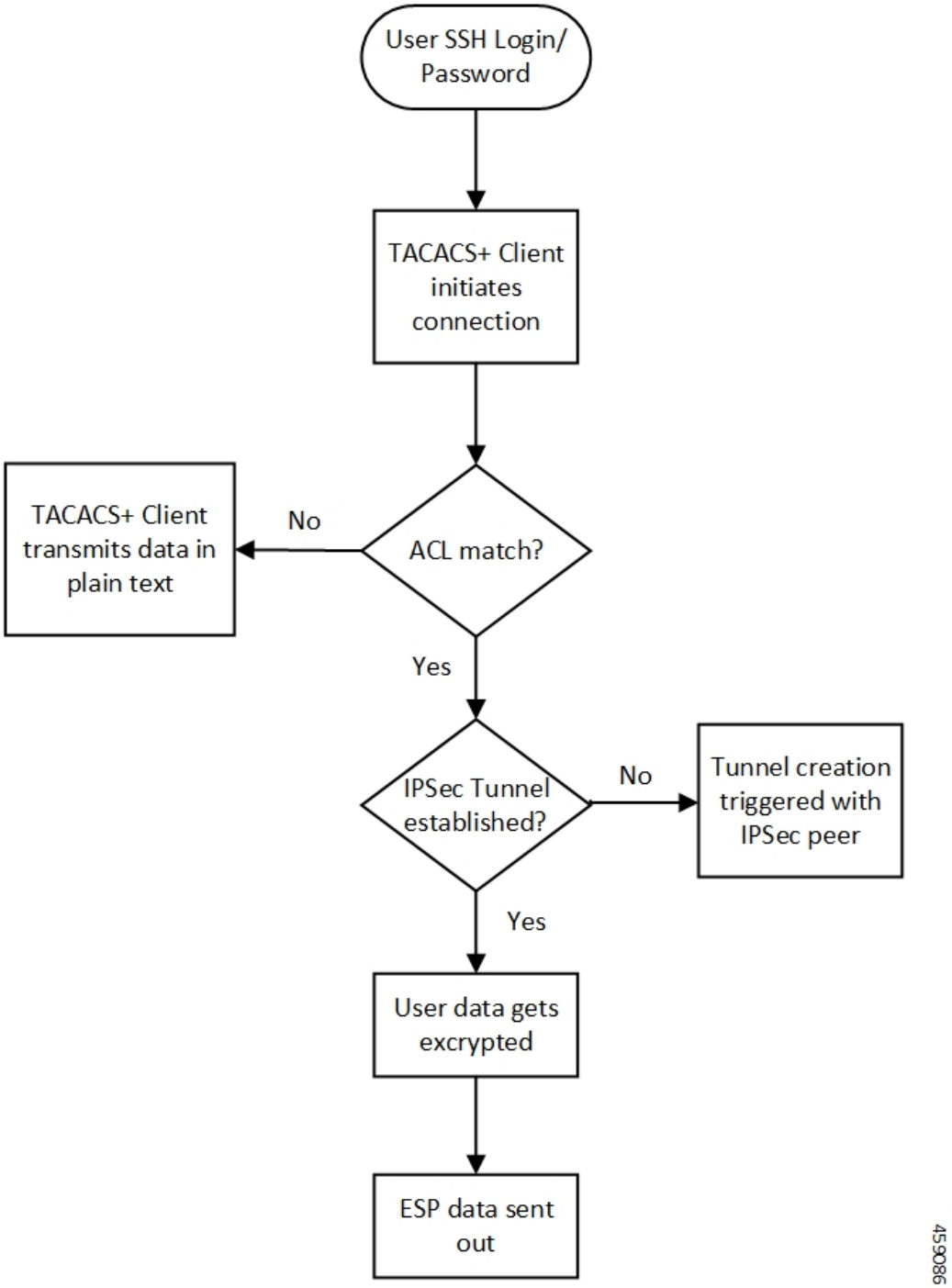
Single IPSec Tunnel with Single Client/Server

How it Works

Depending on the deployment requirement, multiple applications that must be secured has independent ACL rules configured as part of a single crypto-map or separate crypto-map. In both the cases, multiple TUN interfaces are created which are attached to each application requiring encryption.

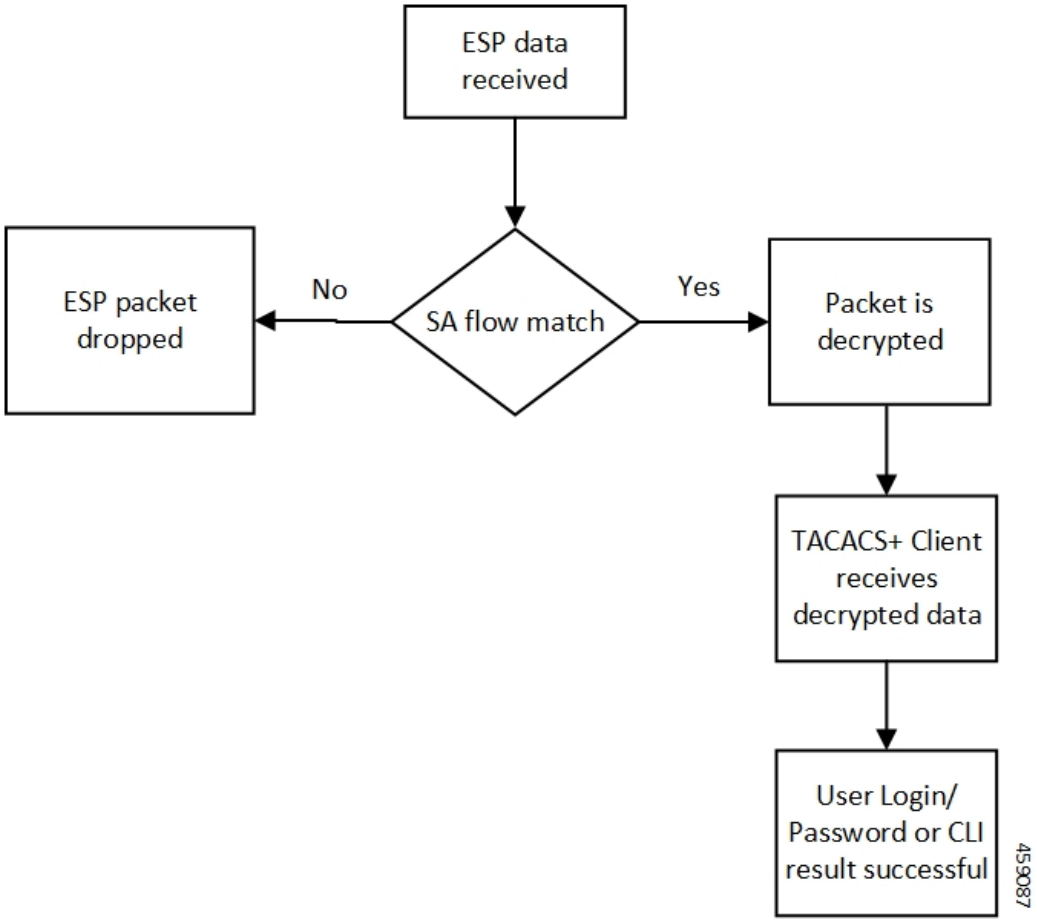
Encryption of TACACS+ Client Data

The following diagram illustrates the tunnel establishment and packet encryption.



Decryption of TACACS+ Server Data

The following diagram illustrates the packet decryption.



The following steps describe the packet flow to achieve TACACS+ data security through IPSec.

1. TACACS+/application initiates TCP connection with the TACACS+ server in the form of first TCP-SYN packet.
2. SYN packet is routed to TUN interface where it's directly read by the IpsecMgr in local context.
3. IpsecMgr sends the TCP-SYN packet to the first instance of NPUSIM for ACL match.
   a. If ACL entry matches with the TCP-SYN packet, it sends the packet back to the IpsecMgr/local.
   b. If the packet doesn't match the ACL entry, NPUSIM sends the packet to the local management interface bypassing the need to encrypt the packet.
4. IpsecMgr/local context receives the packet from NPUSIM after ACL match. It triggers the formation of IPSec tunnel with its peer by exchanging the IKE-INIT/IKE-AUTH packets using local raw socket created in local context.
5. The first TCP-SYN packet is dropped in the IpsecMgr/local after triggering the IPSec tunnel creation.
6. TACACS+/application sends another TCP-SYN packet and steps 2-3b are repeated.
7. When IpsecMgr receives the second TCP-SYN packet after ACL match from NPUSIM and the tunnel is already established, it encrypts the TCP-SYN packet and sends out through ESP raw socket created in the local context by the IpsecMgr/local.

8. IpsecMgr also listens for any ESP packets coming from ESP raw sockets in the local context via management ports.
9. On receiving any ESP packets, IpsecMgr/local sends ESP packet to NPUSIM for any SA flow processing.
10. If the SA flow matches in the NPUSIM, the ESP packet is sent to the IpsecMgr/local which does the decryption of the packet.
11. This packet could be TCP-SYN-ACK which could be the response of the second TCP-SYN packet sent from TACACS+ client to the TACACS+ server.
12. The decrypted packet is sent back to the same TUN interface from where it's sent back to the TACACS+/application.
13. The 2-way communication will be established, by the TACACS+/application which sends out the TCP-ACK packet. The above steps will be repeated to achieve the data security for all subsequent packets.

Recovery

IPsec tunnels are established between TACACS+ client on Active and the TACACS+ server application. There's no IPsec tunnel established between Standby and TACACS+ server. In usual scenario, IPsec endpoints exchange informational (heartbeat) messages to check the health of the IPsec tunnels. If an Active VNF goes down, IPSec endpoint at the TACACS+ server detects dead peer detection (DPD) of the IPsec endpoint on the Active VNF where DPD timeout is also configurable. DPD triggers the clearance of the tunnels on the TACACS+ server side. Once the Standby VNF comes back as Active and TACACS+ application starts to exchange data with the TACACS+ server application, a new IPsec tunnel is established between new Active VNF and the TACACS+ server.

Limitation

Following are the known limitations of the feature:

- TACACS+ using IPv6 is not supported with IPSec that uses IPv6 tunnel endpoints. However, without IPSec, TACACS+ using IPv6 is supported. Also, TACACS+ using IPv4 is supported with and without IPSec using IPv4 tunnel endpoints.
- The crypto maps in the local context must be pre-configured to be part of Day-0/Day-1 configuration. That is, crypto maps in local context, if any, must be configured before crypto maps are configured in any other context.

Configuring TACACS+ over IPSec

This section describes how to configure the TACACS+ over IPSec feature.

Configuring the feature involves the following steps:

1. Configuring TACACS+ Configuration Mode.
2. Provisioning TACACS+ with IPSec.
3. Provisioning TACACS+ with IPSec in Tunnel Mode.

4. Provisioning TACACS+ with IPSec in Transport Mode

Configuring TACACS+ Configuration Mode

Configuration to provision TACACS+ on StarOS/VNF remains the same as was done in non-CUPS architecture. However, for tunnel establishment in "IPSec Tunnel Mode", it's mandatory to provision the **src-ip**. You must reserve one extra Source IP address (*src_ip*) for TACACS+ communication and secure its communication.

For tunnel establishment in "IPSec Transport Mode", there's no requirement to provision an extra **src-ip**. The management interface IP address is picked as the **src-ip**.

The following is a sample configuration:

```
configure
   context context_name
      tacacs mode
         server priority priority_number ip-address server_ip_address password
text_password src_ip
         accounting command
         authorization prompt
      #exit
   aaa tacacs+
end
```

Provisioning TACACS+ with IPSec

The following configuration ensures that all IKE/ESP packets are handled in the user-space IpsecMgr/local and not by the IpsecMgr of non-local context and underlying data-plane like VPP, IFtask, or NPU.

```
configure
   require crypto ikev1-acl software context context
   require crypto ikev2-acl software context context
end
```

Provisioning TACACS+ with IPSec in Tunnel Mode

The following example configuration creates crypto map in the local context in Tunnel mode wherein **209.165.201.1** and **209.165.200.225** is assumed as the TACACS+ server and client IP address respectively.

Note Currently, Tunnel mode is supported only in IKEv2.

```
configure
  context local
    ip access-list foo
      permit ip 209.165.200.225 1 0.0.0.0 209.165.201.1 0.0.0.0
    #exit
    ipsec transform-set B-foo
      group 14
    #exit
    ikev2-ikesa transform-set ikesa-foo
      group 14
```

```
#exit
crypto map foo ikev2-ipv4
  match address foo
  authentication local pre-shared-key encrypted key EncryptedKey1
  authentication remote pre-shared-key encrypted key EncryptedKey2
  ikev2-ikesa max-retransmission 3
  ikev2-ikesa retransmission-timeout 2000
  ikev2-ikesa transform-set list ikesa-foo
  ikev2-ikesa rekey
  payload foo-sa0 match ipv4
    ipsec transform-set list B-foo
    rekey keepalive
  #exit
  peer 209.165.200.226
  ikev2-ikesa policy error-notification
#exit
interface local1
  ip address 209.165.200.227 255.255.255.224
  ipv6 address 2001:420:2c7f:f620::83/64 secondary
  crypto-map foo
#exit
```

Provisioning TACACS+ with IPSec in Transport Mode

The following example configuration creates crypto map in the local context in Transport mode wherein **209.165.200.229** is assumed as the TACACS+ server IP address.

Note Currently, Transport mode is supported only in IKEv1.

```
configure
  context local
    ip access-list foo
      permit tcp 209.165.200.228 0.0.0.0 209.165.200.229 0.0.0.0
    #exit
    ip routing shared-subnet
    ikev1 keepalive dpd interval 3600 timeout 10 num-retry 3
    crypto ipsec transform-set A-foo esp hmac sha1-96 cipher aes-cbc-128
      mode transport
    #exit
    ikev1 policy 1
    #exit
    crypto map foo ipsec-ikev1
      match address foo
      set peer 209.165.200.229
      set ikev1 encrypted preshared-key EncryptedKey1
      set pfs group2
      set transform-set A-foo
    #exit
    interface local1
      ip address 209.165.200.228 255.255.255.224
      ipv6 address 2001:420:2c7f:f620::84/64 secondary
      crypto-map foo
    #exit
```

Monitoring and Troubleshooting

Show Commands and Outputs

The following show CLI commands are available in support of this feature.

- **show crypto map**
- **show crypto ikev2-ikesa security-associations** *summary*
- **show crypto ikev1 security-associations** *summary*
- **show crypto statistics**
- **show crypto ipsec security-associations** *summary*