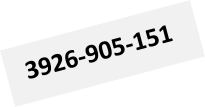
###### Hop By Hop Mode on 3926

This mode only support ETTP-based encryption in 10.6. This is standard MACsec implementation where all the traffic between two directly connected node is MACsec encrypted.

In hop-by-hop mode, all underlying technologies are supported as all the encrypted traffic will be decrypted at the receiver side (i.e MPLS /SR /G8032 Ring..).

* **Scenario1: Vlan-based**

MKA session



**7**

**8**

**7**

**8**

**MKA session**

**L2-Switching**

***Objective***

The objective of this test is to establish hop-by-hop MACsec connection-association between the two nodes over an L2 switching circuit and verify that the traffic running is being encrypted.

***Procedure:***

* Go to the configuration terminal
  + config
* Setup the Forwarding Domains, classifiers and flow points in CLI
  + fds fd fd2 mode vpls
  + classifiers classifier cvid-2 filter-entry classifier: vtag-stack vtags 1 vlan-id 2
  + fps fp fp8 logical-port 8 fd-name fd2 classifier-list cvid- 2
* Create a **key-chain** for authentication and key establishment, it’s used to generate SAK keys to be used to program hardware. It can use both 128 and 256 cryptographic algorithms. The pre-shared key must match on both MACsec devices for the session to be established
  + macsec key-chains key-chain KC1 mka-keys mka-key 01 key 0123456789abcdef0123456789abcdef0123456789abcdef0123456789a bcdef

macsec key-chains key-chain KC1 mka-keys mka-key 01 cryptographic-algorithm AES\_128\_CMAC

exit

* Create a **profile** that combines various user configurable parameters for security
  + macsec macsec-profiles profile pf1

encryption-on true

key-server-priority 10 **(On the server only to generate the SAK)**

macsec-cipher-suite GCM\_AES\_128 replay-window-size 2

sak-rekey-interval 30

* Enable MACsec on interface. We can exclude L2 protocols not to be MACsec encrypted when configuring the interface
  + macsec config interfaces interface 7 strict-mode-on true

exclude-protocols lacp exclude-protocols lldp

* Create a **connection-association** which is a security relationship, established and maintained by key agreement protocols
  + macsec config connection-association CA1 macsec-admin-state enabled

key-chain KC1

macsec-profile pf1 ettp-name 7

The above commands need to be configured at both MACsec nodes to perform encryption.

Once the configuration is performed at both nodes and CA connection is setup, MACsec configuration can be viewed using below commands.

* + show macsec key-chains key-chain KC1

+ KEY-CHAIN +

| Key-Chain | KC1 |

+ + +

| MKA Name | 01 |

| MKA Crypto Algo | AES\_128\_CMAC |

+ + +

* + show macsec profiles profile pf1

+ MACSEC-PROFILES +

| KEY | VALUE |

+ + +

| Profile Name | pf1 |

| Cipher Suite | GCM\_AES\_128 |

| Conf Offset | 0\_BYTES |

| Replay Window Size | 2 |

| Additional Bytes In Clear | 0 |

| Encryption | True |

| Key Server Priority | 10 |

| SAK Rekey Interval | 30 |

+ + +

* + show macsec connection-associations

+ CONNECTION-ASSOCIATION +

| CA Name | Admin State | Oper State | Service Type | Macsec Profile | Key Chain |

+ + + + + + +

| CA1 | **enabled** | **enabled** | ETTP | pf1 | KC1 |

+ + + + + + +

* + show macsec connection-associations connection-association CA1

+ CONNECTION-ASSOCIATION +

| Parameter | Value |

+ + +

| CA Name | CA1 |

| Admin State | enabled |

| Oper State | enabled |

| Oper State Reason | **Operational Up.** |

| Key Server | True |

| Destination Address | 01:80:C2:00:00:03 |

| Mka Ethertype | 0x888e |

| Macsec Profile | pf1 |

| Key Chain Name | KC1 |

| Service Type | ETTP |

| Service Name | 8 |

+ + +

| Peer Secure Channel | |

+ + +

| Mac Address | 94:43:4d:86:7f:d1 |

| Port Identifier | 01-00 |

+ + +

| MKA Statistics | |

+ + +

| In EAPOL MKA invalid CKN Len Frames | 0 |

| In EAPOL MKA invalid Frames | 0 |

| In EAPOL MKA Frames | 44946 |

| Out EAPOL MKA Frames | 50136 |

| In Version Mismatch Frames | 0 |

| In CKN Mismatch Frames | 0 |

| In ICV Mismatch Frames | 0 |

+ + +

| Data Statistics | |

+ + +

| In Valid Packets | 92898269 |

| In Error Packets | 0 |

| In Transform Error Packets | 0 |

| In Control Packets | 80 |

| In Untagged Packets | 0 |

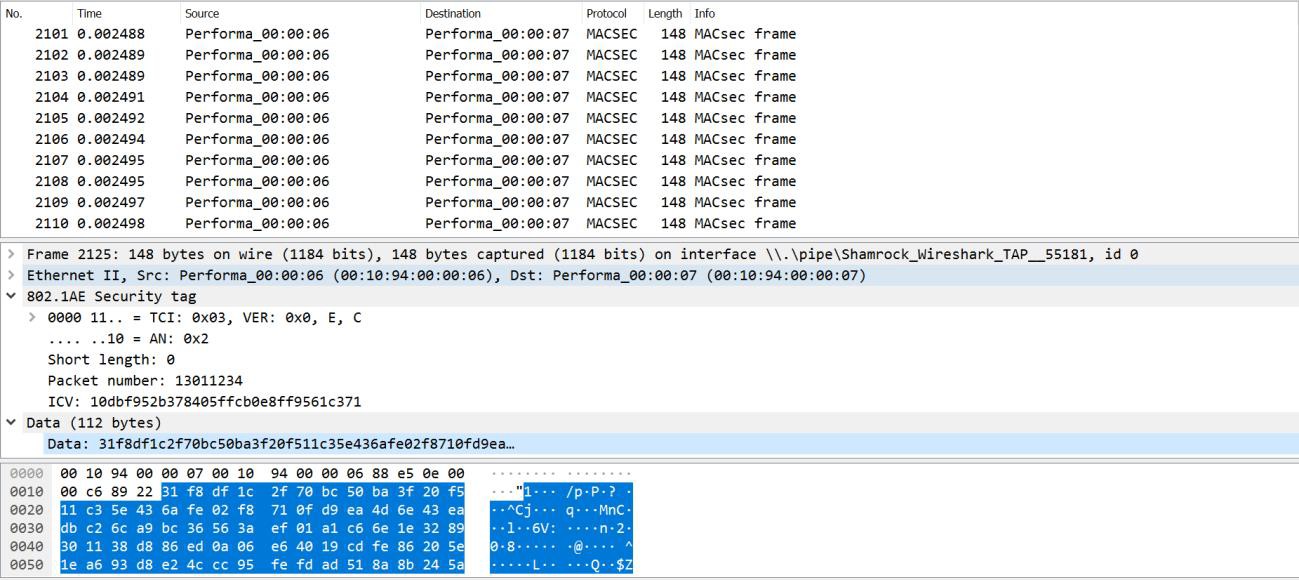
| In No Tag Packets | 0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| | | In | Bad Tag Packets |  | | | | 0 | | |
| | | In | No SCI Packets |  | | | | 0 | | |
| | | In | Unknown Packets |  | | | | 0 | | |
| | | In | Unused SA Packets |  | | | | 0 | | |
| | | In | Unused SA Discarded | Packets | | | | 0 | | |
| | | In | Overrun Discarded Packets | |  | | | 0 | | |
| | | In | Unchecked Packets | |  | | | 0 | | |
| | | In | Invalid Packets | |  | | | 0 | | |
| | | In | Invalid with Sectag C-bit=1 | | Packets | | | 0 | | |
| | | In | Delayed Packets | |  | | | 0 | | |
| | | In | Late Packets | |  | | | 0 | | |
| | | In | Decrypted Octets | |  | | | 12448383971 | | |
| | | In | Validated Octets | |  | | | 0 | | |
| | | Out Encrypted Packets | | | | | | 24988129 | | |
| | | Out Protected Packets | | | | | | 24988129 | | |
| | | Out Control Packets | | | | | | 76 | | |
| | | Out Untagged Packets | | | | | | 0 | | |
| | | Out Transform Error Packets | | | | | | 0 | | |
| | | Out Too Long Discarded Packets | | | | | | 0 | | |
| | | Out Encrypted Octets | | | | | | 0 | | |
| |  + | Out Protected Octets | | | |  + | | 3348411075 | |  + |

We can test the encrypted traffic using an inline packet capture device between the two nodes.

Below is a packet capture of encrypted MACsec traffic, we can see that the MACSsec 802.1AE header

**(w/ Ethertype 0x88e5)** is prepended immediately after SA/DA.



Test Case Results:

Passed: Yes No Verified by Date/Time Comments

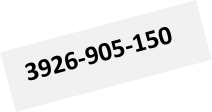
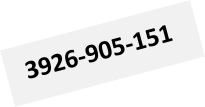
* + **Scenario 2: MPLS based**

MACsec encryption is also supported over MPLS transport for hop-by-hop mode.

The test-case below is using an L2VPN service to test the encrypted traffic and it’s assumed that L2VPN

and MPLS/LDP are already running on underlying IGP.

**IS-IS/MPLS-LDP**



**MKA session**

**8**

**8**

If8

If8

**L2VPN**

***Objective***

The objective of this test is to establish a hop-by-hop MACsec connection-association between the two nodes running MPLS and verify that the traffic running is being encrypted.

***Procedure:***

* Create a **key-chain** for authentication and key establishment, it’s used to generate SAK keys to be used to program hardware. It can use both 128 and 256 cryptographic algorithms. The pre-shared key must match on both MACsec devices for the session to be established
  + macsec key-chains key-chain KC1 mka-keys mka-key 01 key 0123456789abcdef0123456789abcdef0123456789abcdef0123456789a bcdef

macsec key-chains key-chain KC1 mka-keys mka-key 01 cryptographic-algorithm AES\_128\_CMAC

* Create a **profile** that combines various user configurable parameters for security.
  + macsec macsec-profiles profile pf1 encryption-on true

key-server-priority 10 **(On the server only to generate the SAK)**

macsec-cipher-suite GCM\_AES\_128

additional-bytes-in-clear 4 **(Optional, to leave C-tag in clear)**

replay-window-size 2

sak-rekey-interval 30

* Enable MACsec on **interface**
  + macsec config interfaces interface 7 strict-mode-on true

exclude-protocols lacp

exclude-protocols lacp exit

* Create a **connection-association** which is a security relationship, established and maintained by key agreement protocols
  + macsec config connection-association CA1 macsec-admin-state enabled

key-chain KC1

macsec-profile pf1 ettp-name 7

Once the configuration is performed at both the nodes and CA connection is setup between the nodes, it can be viewed using below commands.

* + 3926\_0150> show macsec connection-associations

+ CONNECTION-ASSOCIATION +

| CA Name | Admin State | Oper State | Service Type | Macsec Profile | Key Chain |

+ + + + + + +

| CA1 | **enabled** | **enabled** | ETTP | pf1 | KC1 |

+ + + + + + +

* + show macsec connection-associations connection-association CA1

+ CONNECTION-ASSOCIATION +

| Parameter | Value |

+ + +

| CA Name | CA1 |

| Admin State | enabled |

| Oper State | enabled |

| Oper State Reason | **Operational Up.** |

| Key Server | True |

| Destination Address | 01:80:C2:00:00:03 |

| Mka Ethertype | 0x888e |

| Macsec Profile | pf1 |

| Key Chain Name | KC1 |

| Service Type | ETTP |

| Service Name | 8 |

+ + +

| Peer Secure Channel | |

+ + +

| Mac Address | 94:43:4d:86:7f:d1 |

| Port Identifier | 01-00 |

+ + +

| MKA Statistics | |

+ + +

| In EAPOL MKA invalid CKN Len Frames | 0 |

| In EAPOL MKA invalid Frames | 0 |

| In EAPOL MKA Frames | 44946 |

| Out EAPOL MKA Frames | 50136 |

| In Version Mismatch Frames | 0 |

| In CKN Mismatch Frames | 0 |

| In ICV Mismatch Frames | 0 |

+ + +

| Data Statistics | |

+ + +

| In Valid Packets | 92898269 |

| In Error Packets | 0 |

| In Transform Error Packets | 0 |

| In Control Packets | 80 |

| In Untagged Packets | 0 |

| In No Tag Packets | 0 |

| In Bad Tag Packets | 0 |

| In No SCI Packets | 0 |

| In Unknown Packets | 0 |

| In Unused SA Packets | 0 |

| In Unused SA Discarded Packets | 0 |

| In Overrun Discarded Packets | 0 |

| In Unchecked Packets | 0 |

| In Invalid Packets | 0 |

| In Invalid with Sectag C-bit=1 Packets | 0 |

| In Delayed Packets | 0 |

| In Late Packets | 0 |

| In Decrypted Octets | 12448383971 |

| In Validated Octets | 0 |

| Out Encrypted Packets | 24988129 |

| Out Protected Packets | 24988129 |

| Out Control Packets | 76 |

| Out Untagged Packets | 0 |

| Out Transform Error Packets | 0 |

|  |  |  |
| --- | --- | --- |
| | Out Too Long Discarded Packets | | 0 | | |
| | Out Encrypted Octets | | 0 | | |
| | Out Protected Octets | | 3348411075 | | |
| + | + | + |

Other underlying protocols and services such as PW, LDP, ISIS and MPLS can also be viewed on below outputs:

* + 3926\_0150> show pseudowires

+ PSEUDOWIRES STATE +

| PW Id | Name | Peer IP | Oper State | In Label | Out Label | Flags |

+ + + + + + + +

| 100 | PW1 | 2.2.2.2 | Up | 36008 | 36008 | dUMaF |

+ + + + + + + +

* + 3926\_0150> show isis neighbors

+ ISIS NEIGHBOR STATE +

| Neighbor | | | | | Hold | | |

| Type | System ID | Interface | SNPA | State | Time (s) | Type | Protocol |

+ + + + + + + + +

| P2P | 0101.8110.2151 | if8 | 9443.4d86.7fc5 | Up | 27 | L2 | IS-IS |

+ + + + + + + + +

* + 3926\_0150> show mpls forwarding-table

+ MPLS FORWARDING TABLE +

| | | FTN | | Label | | |

| Code | FEC | ID | LSP-Type | In | Out | Out Intf | Next Hop |

+ + + + + + + + +

| L> | 2.2.2.2/32 | 0 | LSP\_DEFAULT | - | 3 | if8 | 172.18.8.2 |

+ + + + + + + + +

* + 3926\_0150> show ldp adjacencies

+ LDP ADJACENCY STATE +

| | | Hold | | |

| IP Addr | Interface | Time (s) | LDP Identifier | Type |

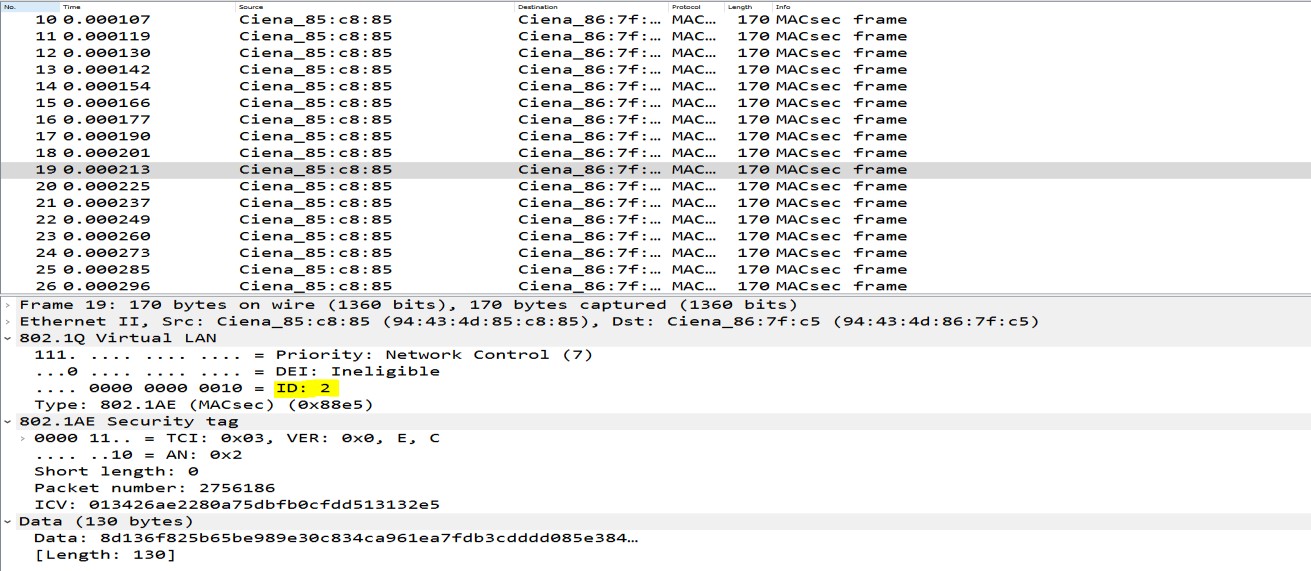
+ + + + + +

| 172.18.8.2 | if8 | 15 | 2.2.2.2:0 | LINK HELLO |

| 2.2.2.2 | if8 | 45 | 2.2.2.2:0 | TARGETED\_HELLO |

+ + + + + +

The encrypted traffic is depicted on the packet capture below. In this case the VLAN tag is left in clear:



**Test Case Results:**

Passed: Yes No Verified by Date/Time Comments