University of Rome Tor Vergata
ICT and Internet Engineering

Network and System Defense

Alessandro Pellegrini, Angelo Tulumello

A.A. 2023/2024

Lecture 4: 802.1x

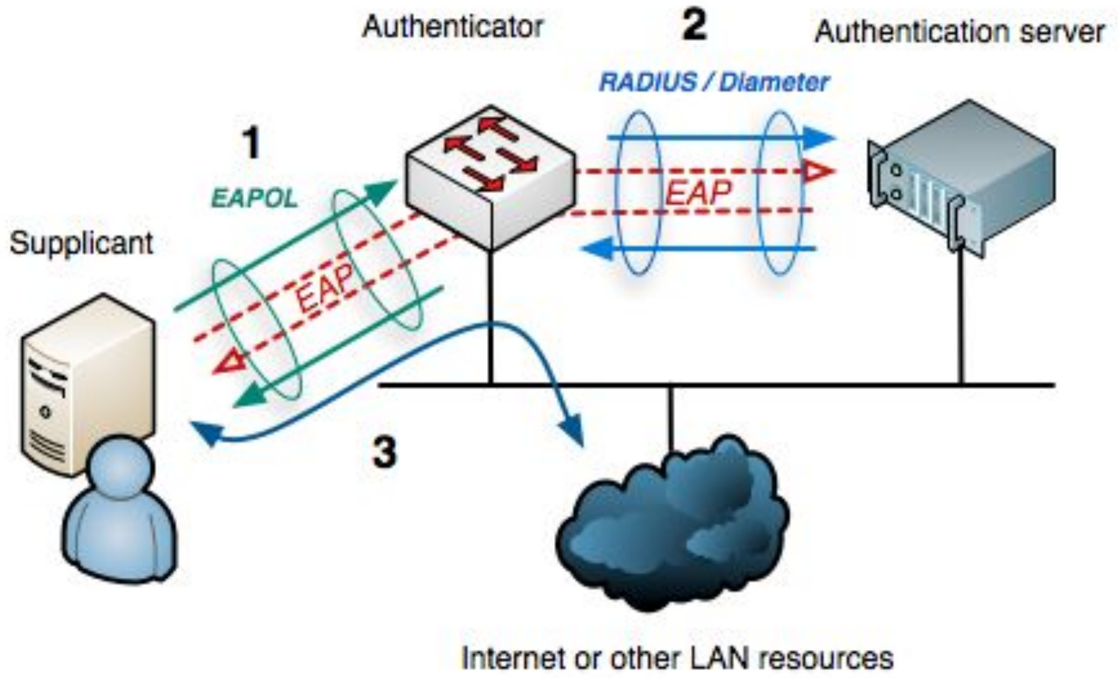
Angelo Tulumello

Slides by Prof. Marco Bonola

802.1x and EAPOL

- ***IEEE 802.1X*** is an IEEE Standard for ***port-based Network Access Control*** (PNAC). It is part of the IEEE 802.1 group of networking protocols.
- It provides an authentication mechanism to devices wishing to attach to a LAN or WLAN.
- 802.1X defines the encapsulation of the ***Extensible Authentication Protocol (EAP) over LAN or EAPOL***.
- EAPOL was originally designed for IEEE 802.3 Ethernet in 802.1X-2001, but was clarified to suit other IEEE 802 LAN technologies such as IEEE 802.11 wireless and Fiber Distributed Data Interface (ANSI X3T9.5/X3T12 and ISO 9314) in 802.1X-2004
- EAP is independent from 802.1x
    - it’s even standardized by a different body (i.e. the IETF, see next slide)
- 802.1x **benefits**
    - 802.1X is a Layer 2 protocol and does not involve Layer 3 processing. It does not require high performance of access devices, reducing network construction costs.
    - Authentication packets and data packets are transmitted through different logical interfaces, improving network security.

802.1x at a glance



Extensible Authentication Protocol

- ***Extensible Authentication Protocol (EAP)*** is an authentication framework frequently used in network and internet connections. It is defined in RFC 3748, which made RFC 2284 obsolete, and is updated by RFC 5247.
- EAP is an authentication framework for providing the transport and usage of material and parameters generated by EAP methods.
- EAP is not a *wire protocol*; ***instead it only defines the information of the interface and the formats.*** Each protocol that uses EAP defines a way to encapsulate the user EAP messages within that protocol's messages.
- ***EAP is an authentication framework, not a specific authentication mechanism.***
  - It provides some common functions and negotiation of authentication methods called EAP methods.
  - There are currently about 40 different methods defined. Methods defined in IETF RFCs include EAP-MD5, EAP-POTP, EAP-GTC, EAP-TLS, EAP-IKEv2, EAP-SIM, EAP-AKA, and EAP-AKA'.
  - Additionally, a number of vendor-specific methods and new proposals exist. Commonly used modern methods capable of operating in wireless networks include EAP-TLS, EAP-SIM, EAP-AKA, LEAP and EAP-TTLS.

RADIUS

- ***Remote Authentication Dial-In User Service (RADIUS)*** is another protocol involved in the 802.1x architecture
  - ***but not defined by 802.1x***
- RADIUS is a networking protocol that provides centralized authentication, authorization, and accounting (AAA) management for users who connect and use a network service
  - standardized by the IETF (rfc 2865)
- RADIUS is a client/server protocol that runs in the application layer, and can use either TCP or UDP.
- Network access servers, which control access to a network, usually contain a RADIUS client component that communicates with the RADIUS server.
- As already said, RADIUS is often the ***back-end of choice for 802.1X authentication***
  - although also DIAMETER is considered ...

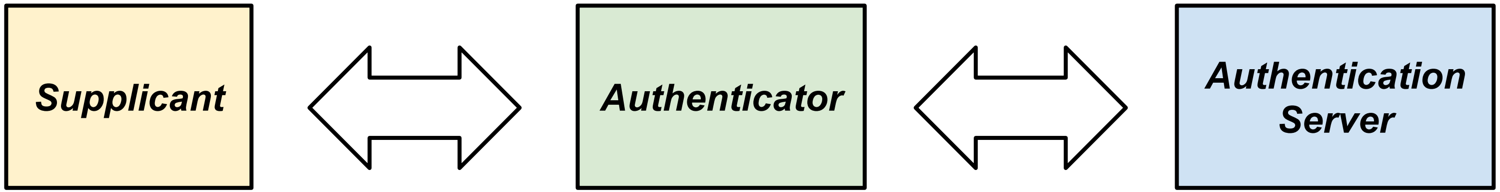
RADIUS

- ***Remote Authentication Dial-In User Service (RADIUS*** … nvolved in the 802.1x architecture
  - ***but not defined by 802.1x***
- RADIUS is a networking pro … ion, and accounting (AA …
  - stand …
- R … can use either T …
- N … twork, usually contain a RADIUS clie … e RADIUS server.
- As a … the ***back-end of choice for 802.1X authentication***

RADIUS is covered by G. Bianchi's course CNS and it is not object of this course

802.1X overview

- The 802.1X authentication system uses a standard client/server architecture with three components:
  - ***Client (AKA 802.1x Supplicant):*** the client is usually a user terminal or a GUI. The user triggers 802.1X authentication using client software. The client must support Extensible Authentication Protocol over LAN (EAPoL)
  - ***Access Device (AKA 802.1x Authenticator):*** the access device is usually a network device that supports the 802.1X protocol. It provides a port, either physical or logical, for the client to access the LAN
  - ***Authentication Server:*** the authentication server, typically a RADIUS server, carries out authentication, authorization, and accounting (AAA) on users



802.1X overview

- In the 802.1X authentication system, the client, access device, and authentication server exchange information using the Extensible Authentication Protocol (EAP).
- EAP can run without an IP address over various bottom layers, including the data link layer and upper-layer protocols (such as UDP and TCP).
    - This offers great flexibility to 802.1X authentication.
- The EAP packets transmitted between the client and access device are encapsulated in EAPoL format and transmitted across the LAN.
- ***2 authentication modes between the access device and authentication server*** based on the client support and network security requirements:
    - ***EAP termination mode:*** The access device terminates EAP packets and encapsulates them into RADIUS packets. The authentication server then uses the standard RADIUS protocol to implement authentication, authorization, and accounting.
    - ***EAP relay mode:*** The access device directly encapsulates the received EAP packets into RADIUS using EAP over RADIUS (EAPoR) packets, and then transmits these packets over a complex network to the authentication server

EAPOL packet format

PAE Ethernet Type	Protocol Version	Type	Length	Packet Body

Field	Bytes	Description
PAE Ethernet Type	2	Indicates the protocol type. The value is fixed at 0x888E.
Protocol Version	1	Indicates the protocol version number supported by the EAPoL packet sender. • 0x01: 802.1X-2001 • 0x02: 802.1X-2004 • 0x03: 802.1X-2010
Type	1	Indicates the type of an EAPoL data packet: • 00: EAP-Packet, which is an authentication packet that carries authentication information. • 01: EAPoL-Start, which an authentication start packet sent by a client. • 02: EAPoL-Logoff, which is a logout request packet sent by a client. • 03: EAPoL-Key, which carries key information. Note that the EAPoL-Start, EAPoL-Logoff, and EAPoL-Key packets are transmitted only between the client and access device.
Length	2	Indicates the data length, that is, the length of the Packet Body field, in bytes. The value 0 indicates that the Packet Body field does not exist. For the EAPoL-Start and EAPoL-Logoff packets, the values of the Length field are both 0.
Packet Body	2	Indicates the data content.

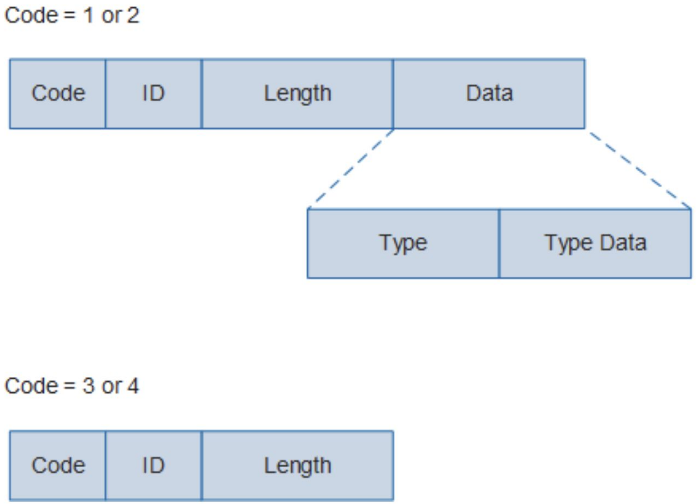
EAP packet format

Code (1 byte): Indicates the type of an EAP data packet:
1:Request, ***2:Response***, ***3:Success***, ***4:Failure***

ID (1 byte): Is used to match a Response packet with the corresponding Request packet.

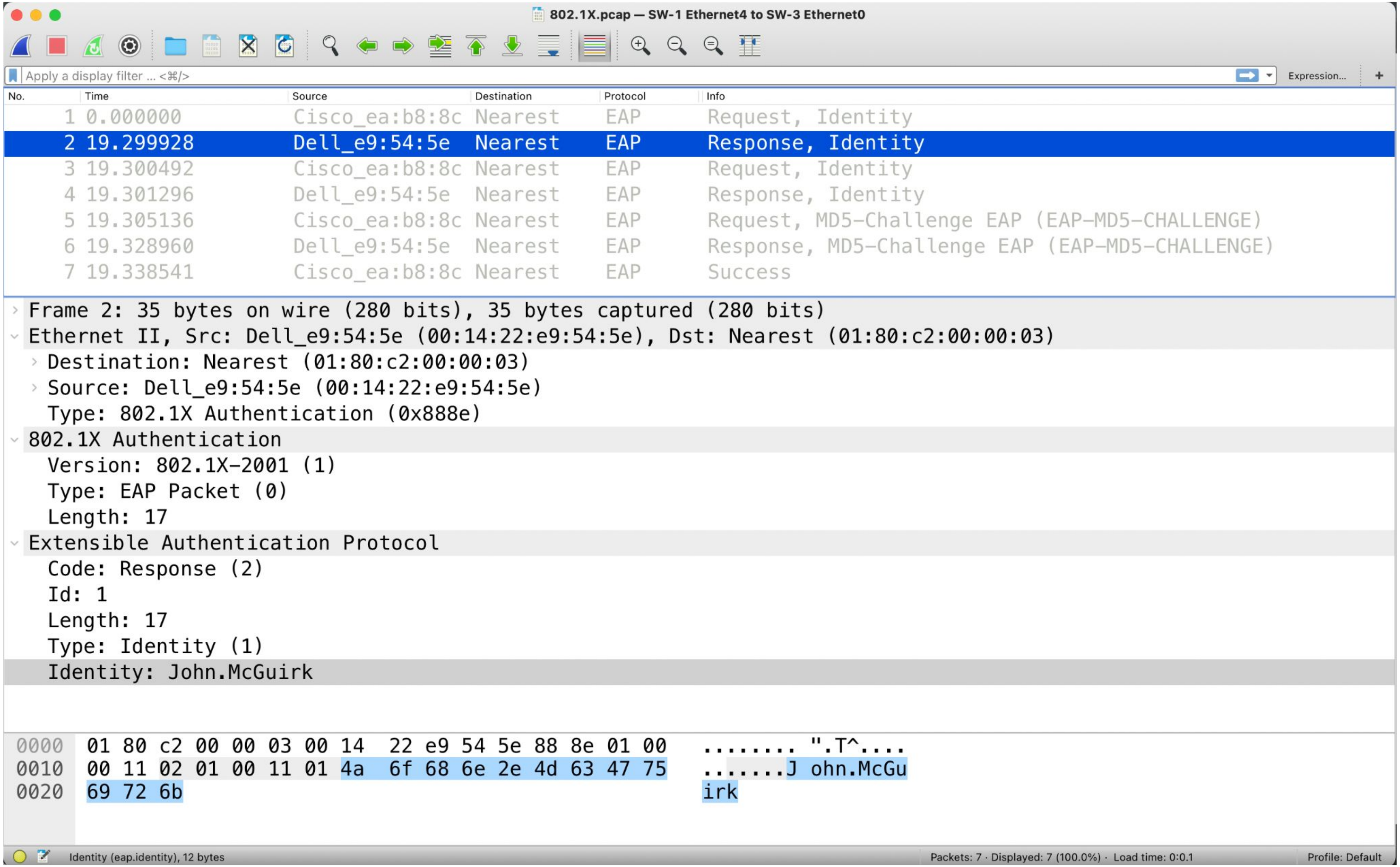
Length (2 byte): Indicates the length of an EAP data packet, including the Code, ID, Length, and Data fields. Bytes outside the range of the Length field are treated as padding at the data link layer and ignored on reception.

Data (variable): When the value of the Code field is 1 or 2 the Type field is one byte long and indicates the type of the Request or Response packet. The Type Data field is multiple bytes long and its value is determined by the Type field.



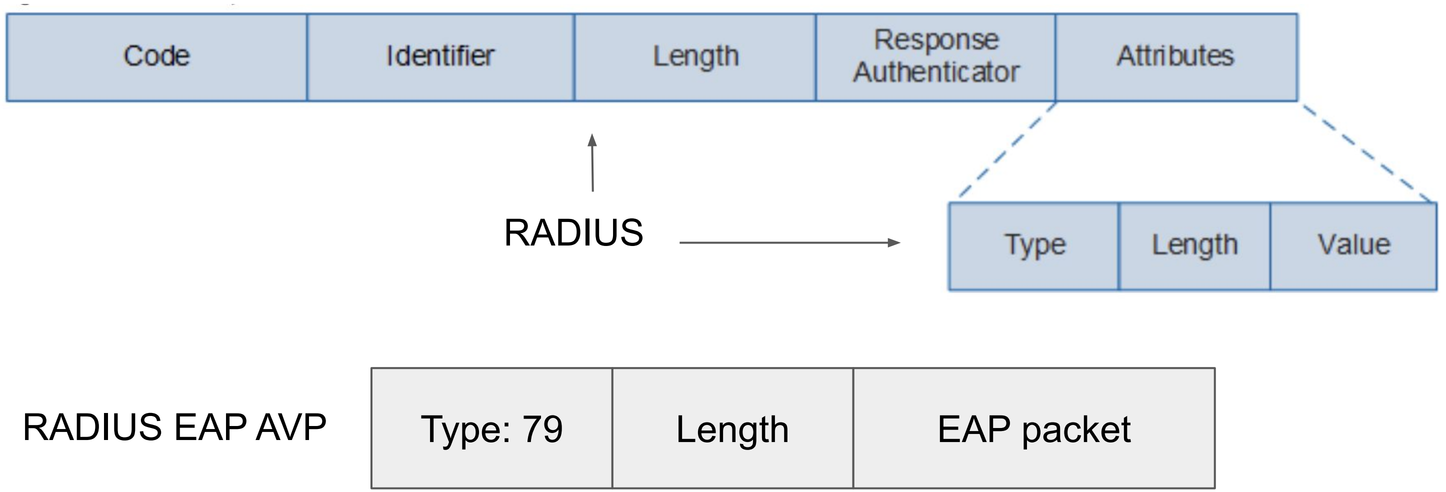
Common values of the Type field

Type Field Value	Packet Type	Description
1	Identity	Requests or returns the user name information entered by a user.
2	Notification	Transmits notification information about some events, such as password expiry and account locking. It is an optional message.
3	NAK	Indicates negative acknowledgment and is used only in a Response packet. For example, if the access device uses an authentication method not supported by the client to initiate a request, the client can send a Response/NAK packet to notify the access device of the authentication methods supported by the client.
4	MD5-Challenge	Indicates that the authentication method is MD5-Challenge.
5	OTP	Indicates that the authentication method is One-Time Password (OTP). For example, during e-banking payment, the system sends a one-time password through an SMS message.
6	GTC	Indicates that the authentication method is Generic Token Card (GTC). A GTC is similar to an OTP except that a GTC usually corresponds to an actual device. For example, many banks in China provide a dynamic token for users who apply for e-banking. This token is a GTC.
13	EAP-TLS	Indicates that the authentication method is EAP-TLS.
21	EAP-TTLS	Indicates that the authentication method is EAP-TTLS.
25	EAP-PEAP	Indicates that the authentication method is EAP-PEAP.
254	Expanded Types	Indicates an expanded type, which can be customized by vendors.
255	Experimental use	Indicates a type for experimental use.


802.1X.pcap — SW-1 Ethernet4 to SW-3 Ethernet0
Apply a display filter … <⌘/>
Expression…
No. Time Source Destination Protocol Info
1 0.000000 Cisco_ea:b8:8c Nearest EAP Request, Identity
2 19.299928 Dell_e9:54:5e Nearest EAP Response, Identity
3 19.300492 Cisco_ea:b8:8c Nearest EAP Request, Identity
4 19.301296 Dell_e9:54:5e Nearest EAP Response, Identity
5 19.305136 Cisco_ea:b8:8c Nearest EAP Request, MD5-Challenge EAP (EAP-MD5-CHALLENGE)
6 19.328960 Dell_e9:54:5e Nearest EAP Response, MD5-Challenge EAP (EAP-MD5-CHALLENGE)
7 19.338541 Cisco_ea:b8:8c Nearest EAP Success
Frame 2: 35 bytes on wire (280 bits), 35 bytes captured (280 bits)
Ethernet II, Src: Dell_e9:54:5e (00:14:22:e9:54:5e), Dst: Nearest (01:80:c2:00:00:03)
Destination: Nearest (01:80:c2:00:00:03)
Source: Dell_e9:54:5e (00:14:22:e9:54:5e)
Type: 802.1X Authentication (0x888e)
802.1X Authentication
Version: 802.1X-2001 (1)
Type: EAP Packet (0)
Length: 17
Extensible Authentication Protocol
Code: Response (2)
Id: 1
Length: 17
Type: Identity (1)
Identity: John.McGuirk
0000 01 80 c2 00 00 03 00 14 22 e9 54 5e 88 8e 01 00 .... ".T^....
0010 00 11 02 01 00 11 01 4a 6f 68 6e 2e 4d 63 47 75 ....J ohn.McGu
0020 69 72 6b irk
Identity (eap.identity), 12 bytes
Packets: 7 · Displayed: 7 (100.0%) · Load time: 0:0.1
Profile: Default

EAPOR packet format

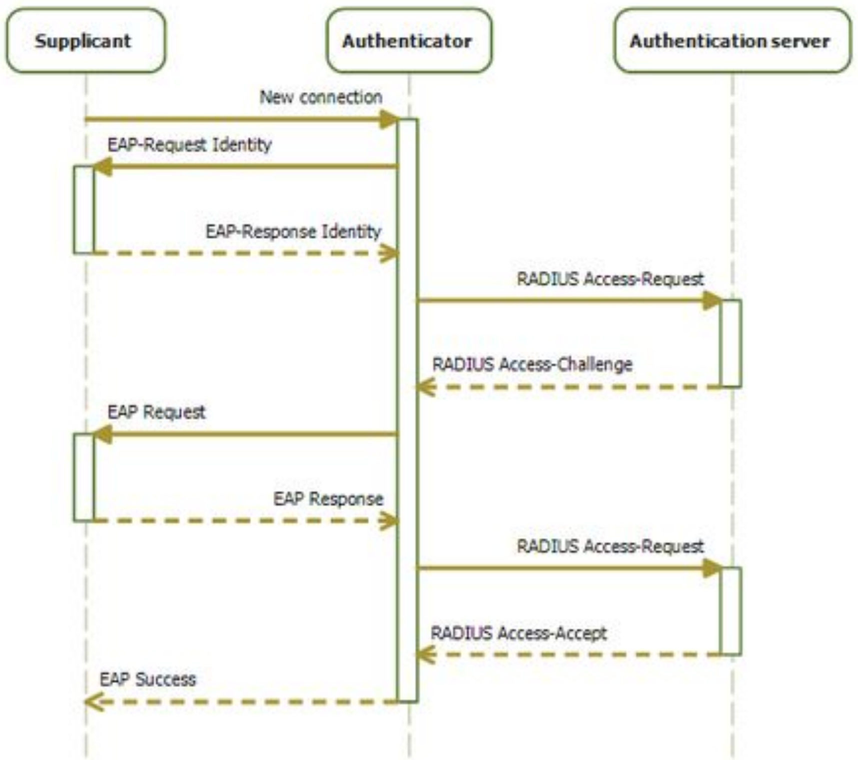
- To support EAP relay, the following RADIUS attributes are added to the RADIUS protocol:
  - EAP-Message: is used to encapsulate EAP packets.
  - Message-Authenticator: is used to authenticate and verify authentication packets to protect against spoofing of invalid packets.



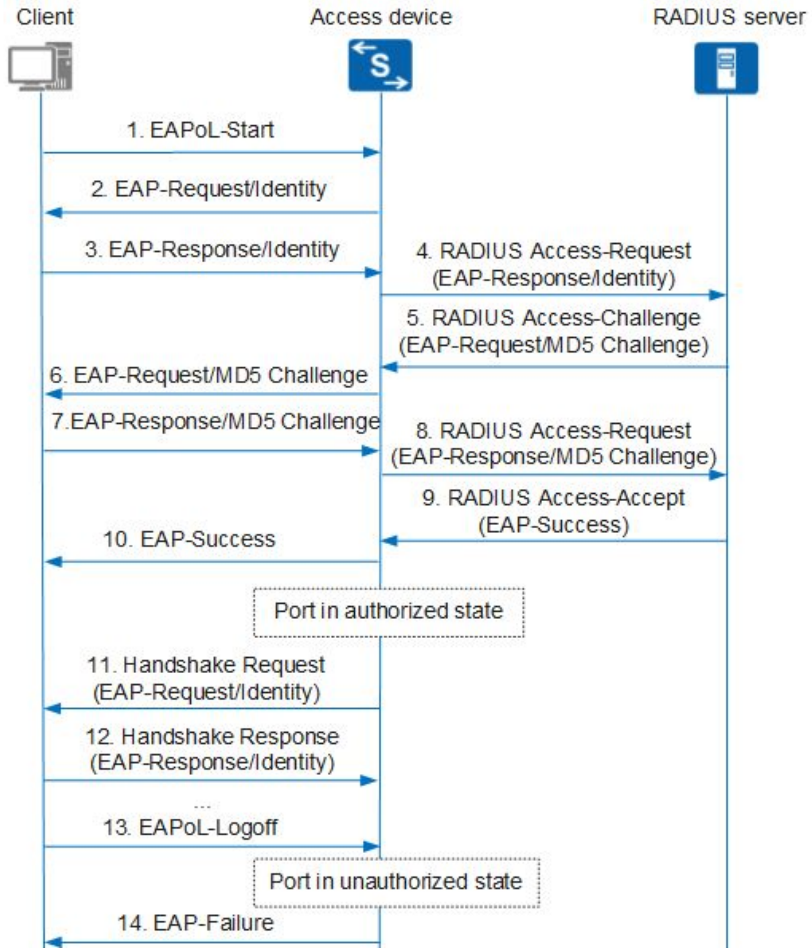
```
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> User Datagram Protocol, Src Port: 53031, Dst Port: 1812
v RADIUS Protocol
    Code: Access-Request (1)
    Packet identifier: 0x67 (103)
    Length: 87
    Authenticator: 40b664dbf5d681b2adbd1769515118c8
    [The response to this request is in frame 2]
  v Attribute Value Pairs
    > AVP: l=7 t=User-Name(1): steve
    > AVP: l=18 t=User-Password(2): Encrypted
    > AVP: l=6 t=NAS-IP-Address(4): 192.168.0.28
    > AVP: l=6 t=NAS-Port(5): 123
    > AVP: l=18 t=Message-Authenticator(80): 5f0f8647e8c89bd881364268fcd04532
    v AVP: l=12 t=EAP-Message(79) Last Segment[1]
        Type: 79
        Length: 12
        EAP fragment: 0266000a017374657665
      v Extensible Authentication Protocol
          Code: Response (2)
          Id: 102
          Length: 10
          Type: Identity (1)
          Identity: steve
```

Authentication Process in EAP Termination Mode

Message exchange (high level)

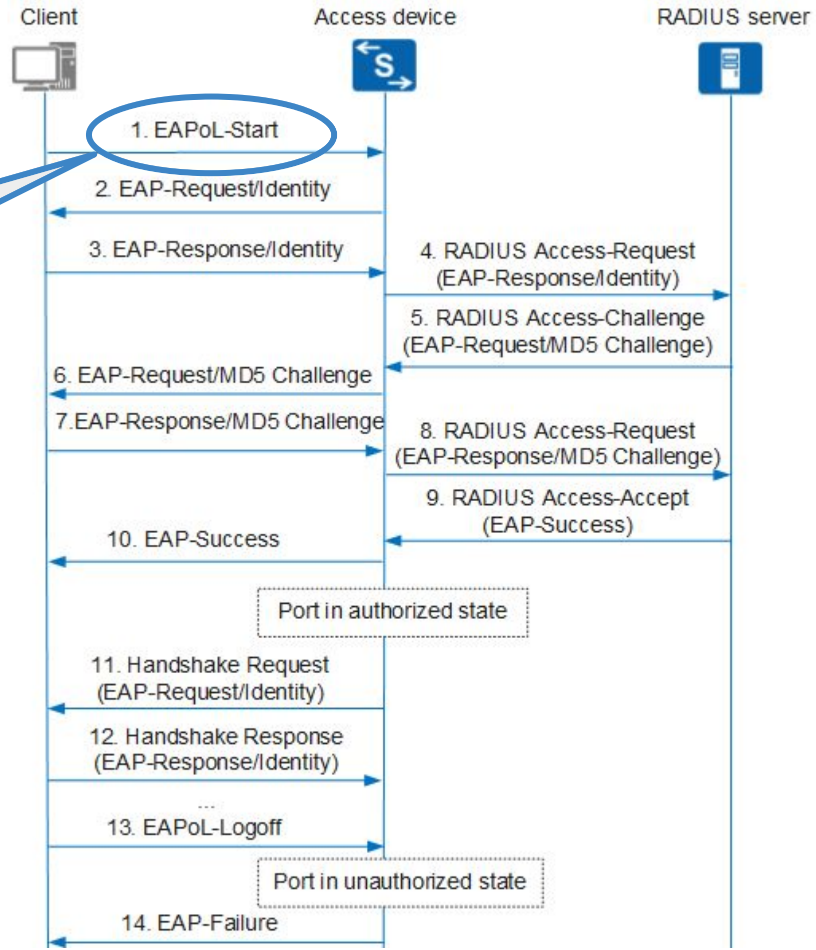


Relay Mode (MD5-challenge)



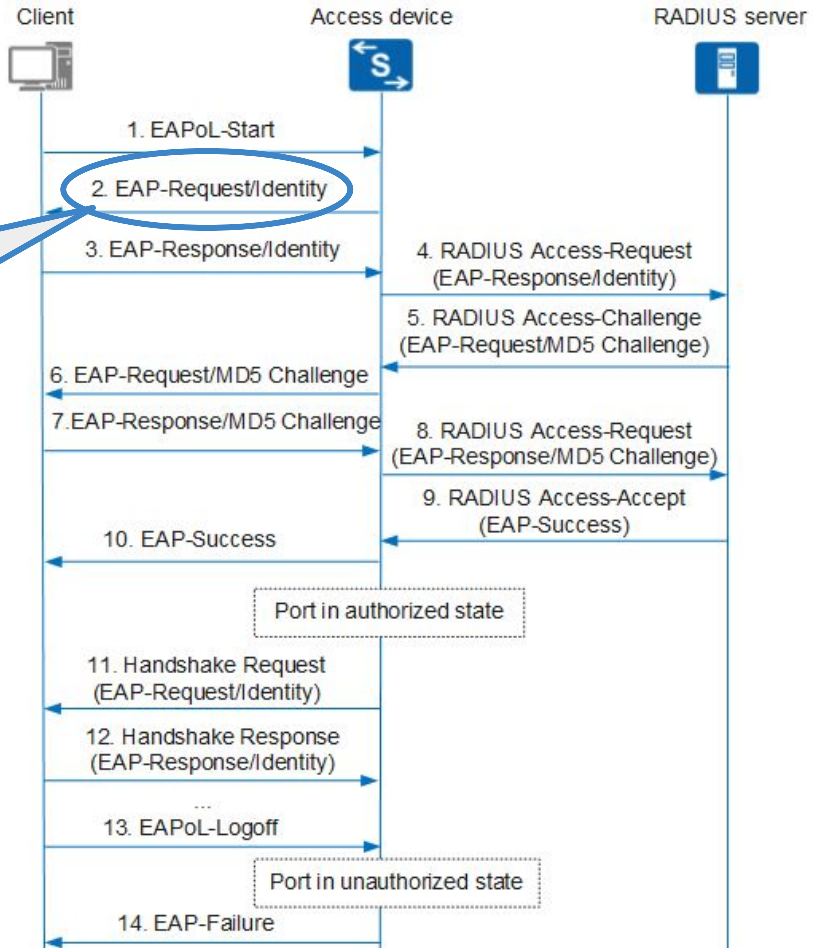
Relay Mode (MD5-challenge)

To access an extranet, a user starts the 802.1X client program, enters the applied and registered user name and password, and initiates a connection request. At this time, the client sends an ***EAPoL-Start*** packet to the access device to start the authentication process



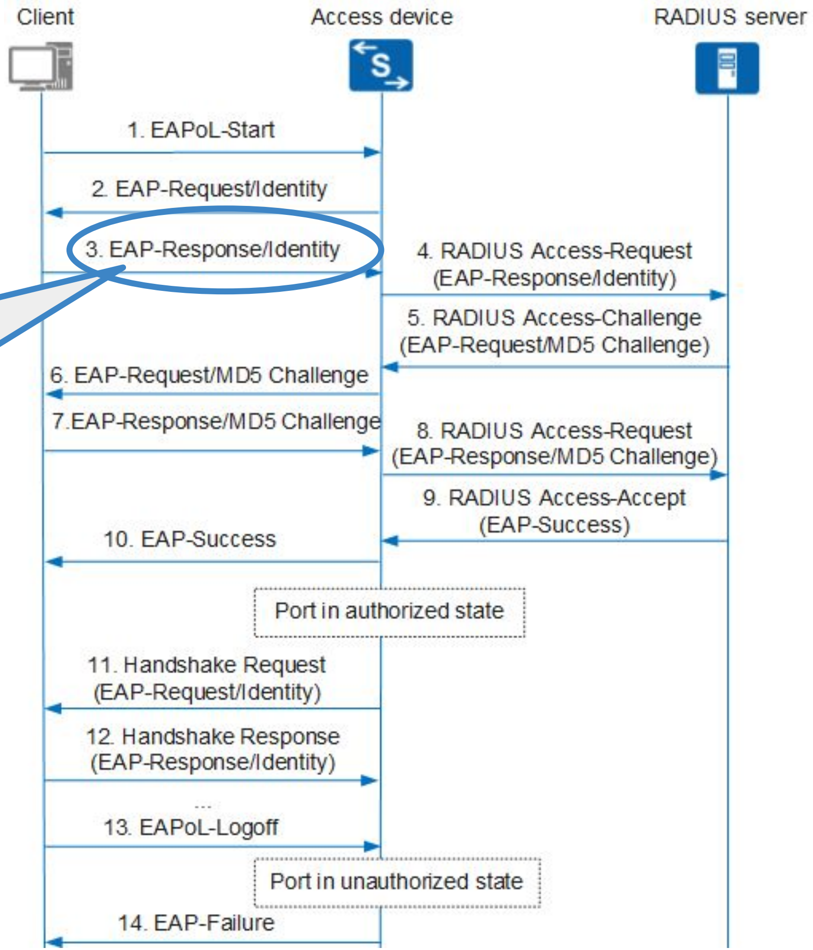
Relay Mode (MD5-challenge)

After receiving the EAPoL-Start packet, the access device returns an ***EAP-Request/Identity*** packet to the client for its identity

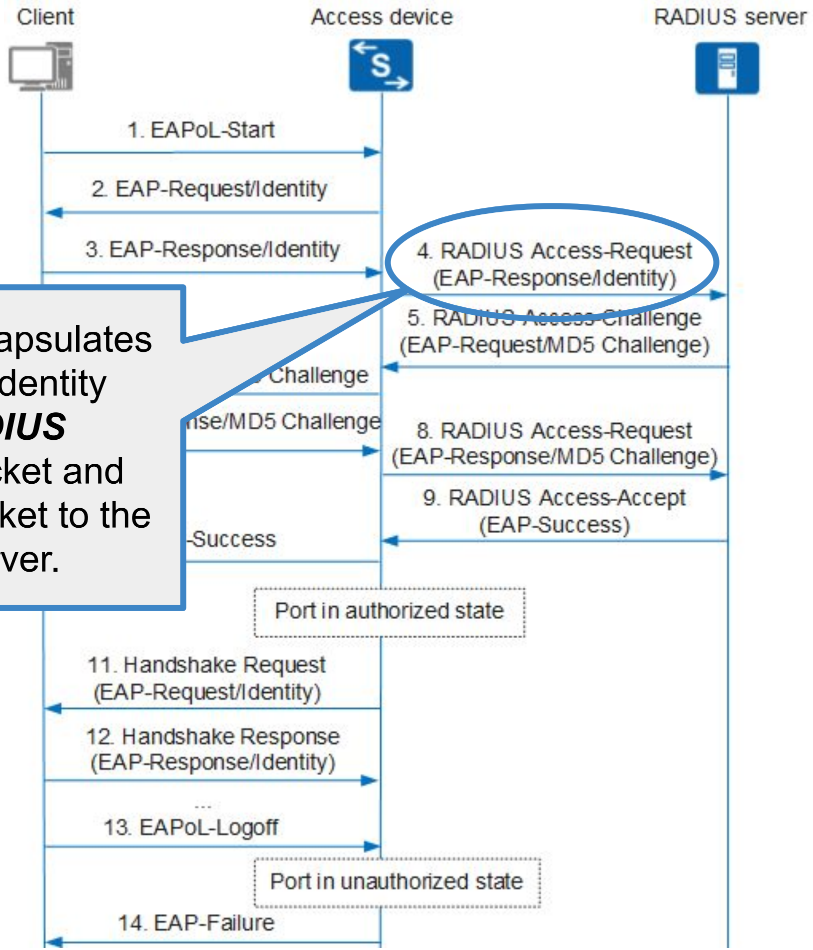


Relay Mode (MD5-challenge)

Upon receipt of the EAP-Request/Identity packet, the client sends an ***EAP-Response/Identity*** packet that contains the user name to the access device.

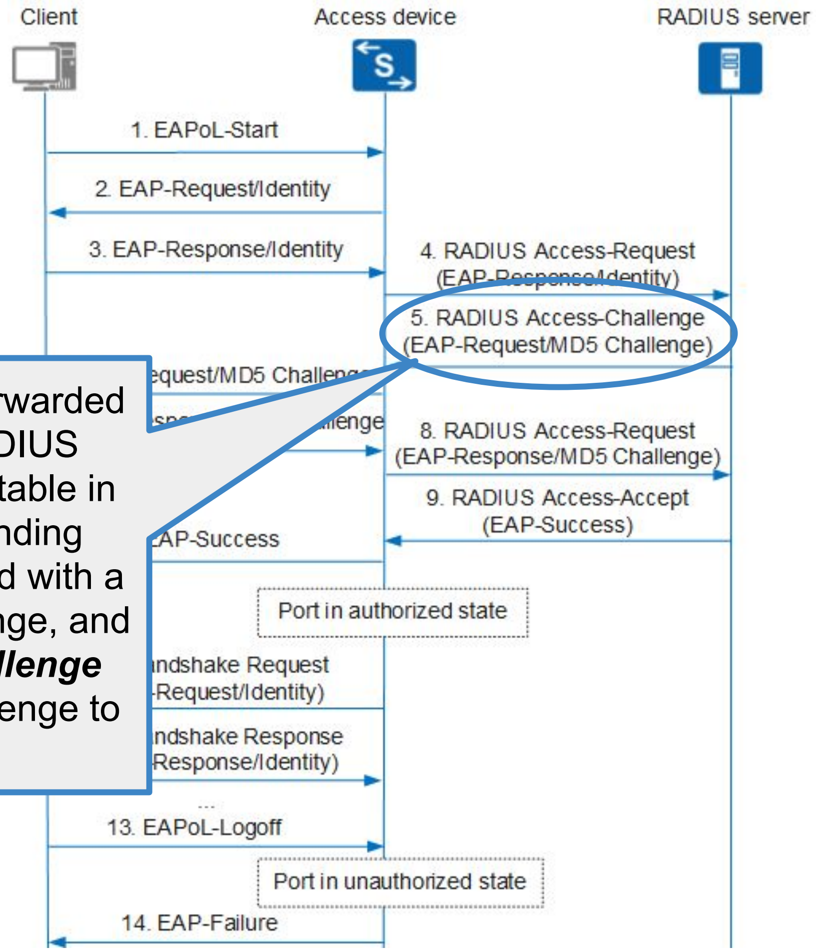


Relay Mode (MD5-challenge)



The access device encapsulates the EAP-Response/Identity packet into a ***RADIUS Access-Request*** packet and sends the RADIUS packet to the authentication server.

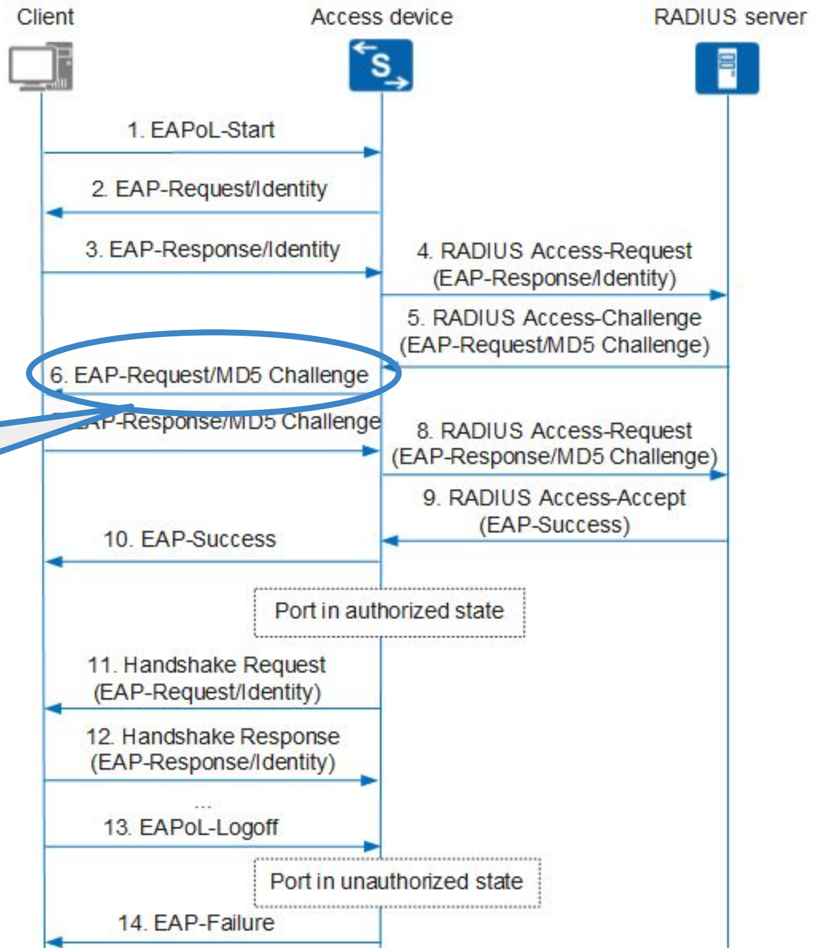
Relay Mode (MD5-challenge)



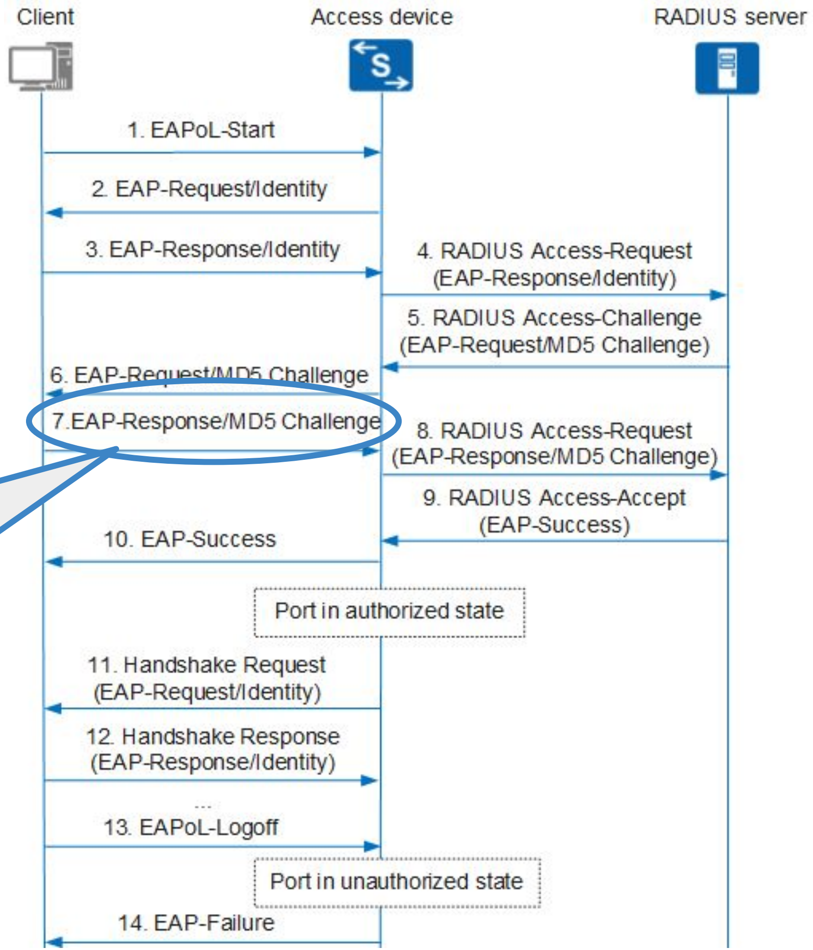
After receiving the user name forwarded by the access device, the RADIUS server searches the user name table in the database for the corresponding password, encrypts the password with a randomly generated MD5 challenge, and sends a ***RADIUS Access-Challenge*** packet containing the MD5 challenge to the access device.

Relay Mode (MD5-challenge)

The access device forwards the ***MD5 challenge*** sent by the RADIUS server to the client



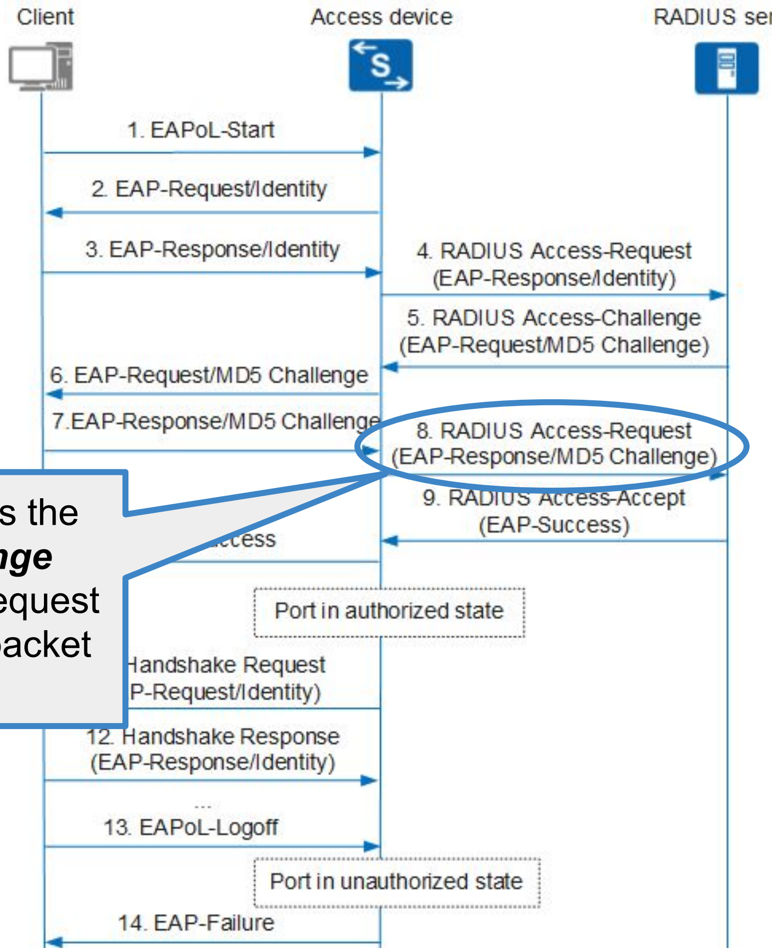
Relay Mode (MD5-challenge)



Upon receipt of the MD5 challenge, the client encrypts the password with the MD5 challenge, generates an ***EAP-Response/MD5-Challenge*** packet, and sends the packet to the access device:

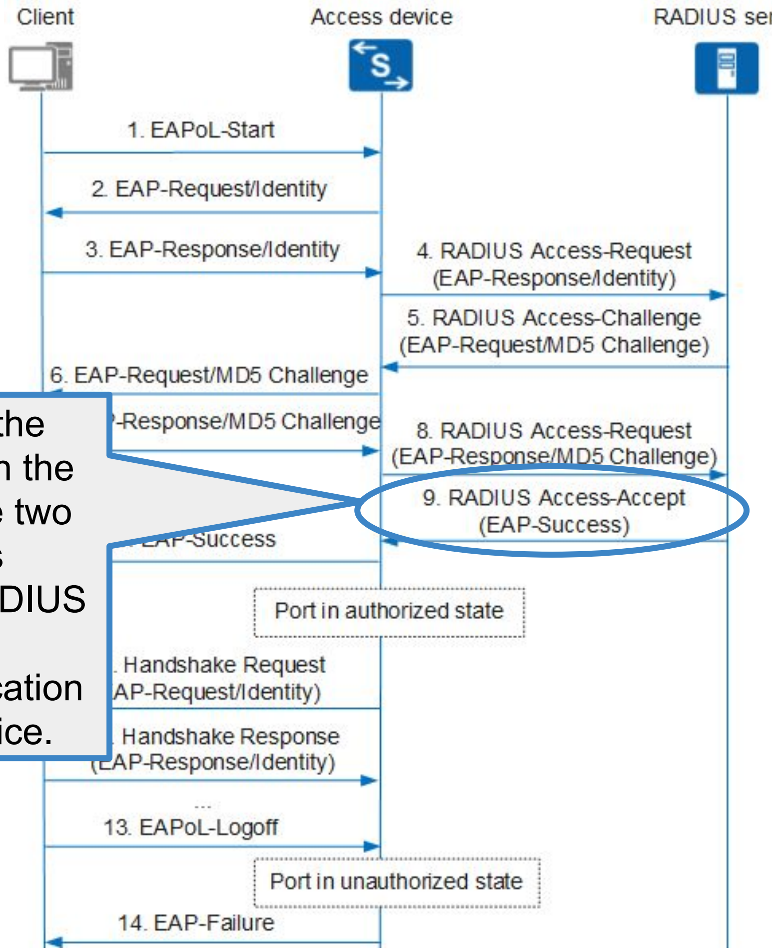
MD5 hash of the EAP-Message id + challenge string + user password

Relay Mode (MD5-challenge)



The access device encapsulates the ***EAP-Response/MD5-Challenge*** packet into a RADIUS Access-Request packet and sends the RADIUS packet to the RADIUS server.

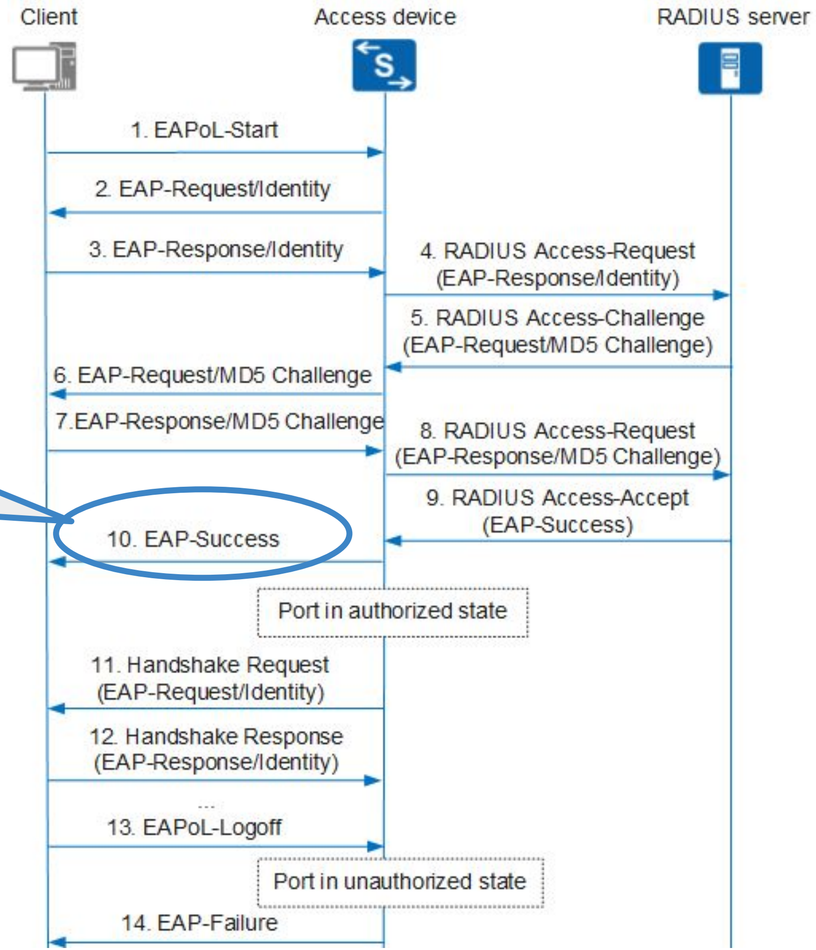
Relay Mode (MD5-challenge)



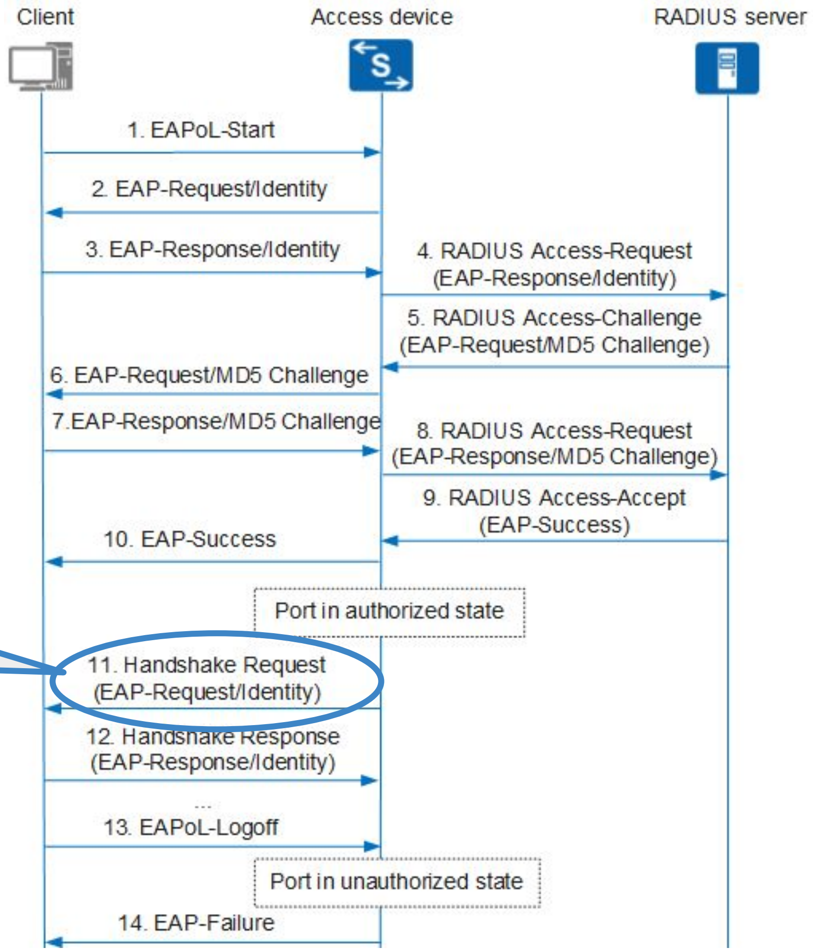
The RADIUS server compares the received encrypted password with the locally encrypted password. If the two passwords match, the user is considered to be valid and the RADIUS server sends a ***RADIUS Access-Accept*** packet (authentication is successful) to the access device.

Relay Mode (MD5-challenge)

After receiving the RADIUS Access-Accept packet, the access device sends an ***EAP-Success*** packet to the client, changes the port state to authorized, and allows the user to access the network through the port.



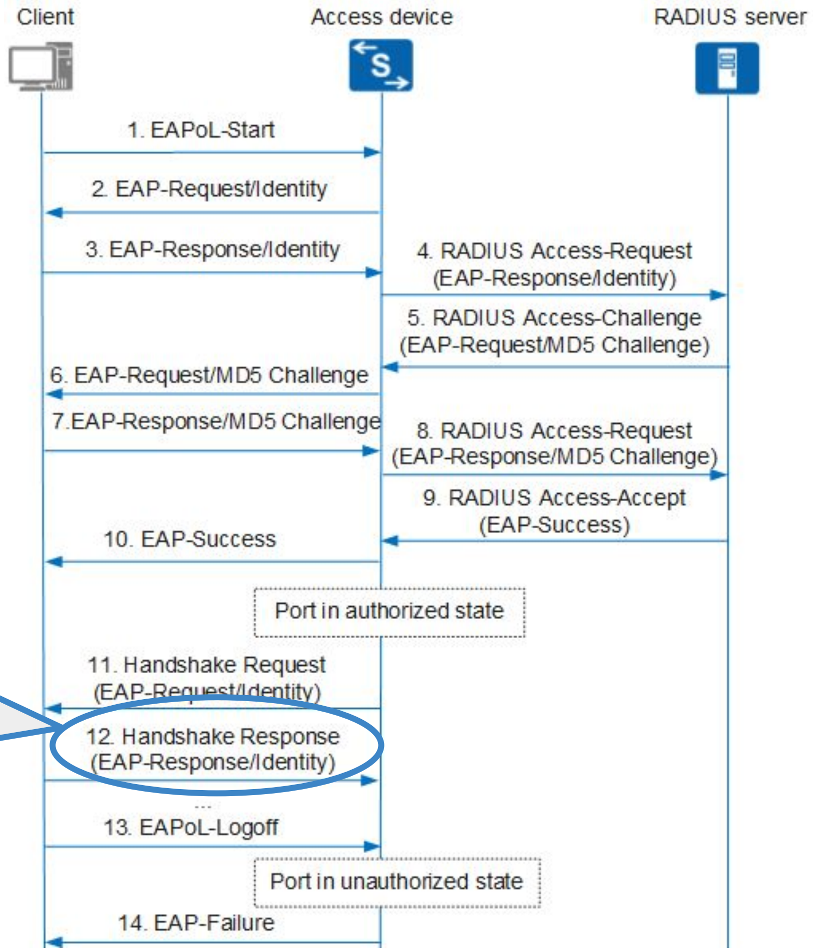
Relay Mode (MD5-challenge)



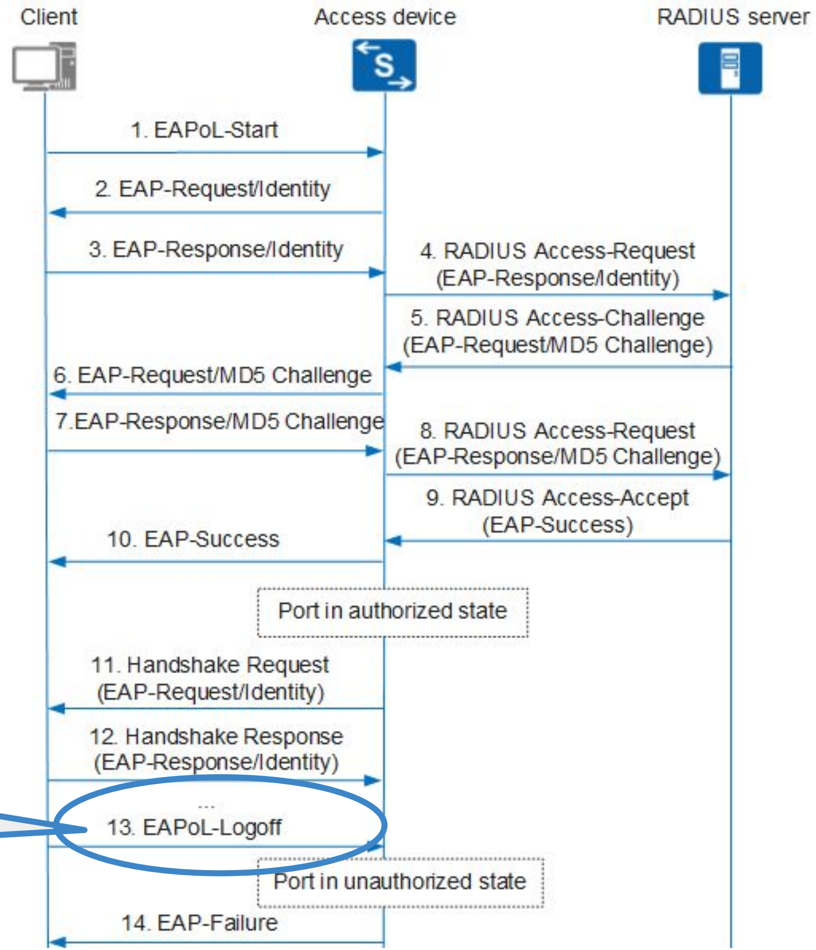
When the user is online, the access device ***periodically sends a handshake packet to the client to monitor the user***.

Relay Mode (MD5-challenge)

After receiving a handshake packet, ***the client sends a response packet to the access device, indicating that the user is still online.*** By default, the access device disconnects the user if it does not receive any response from the client after sending two consecutive handshake packets. The handshake mechanism allows the access device to detect unexpected user disconnections

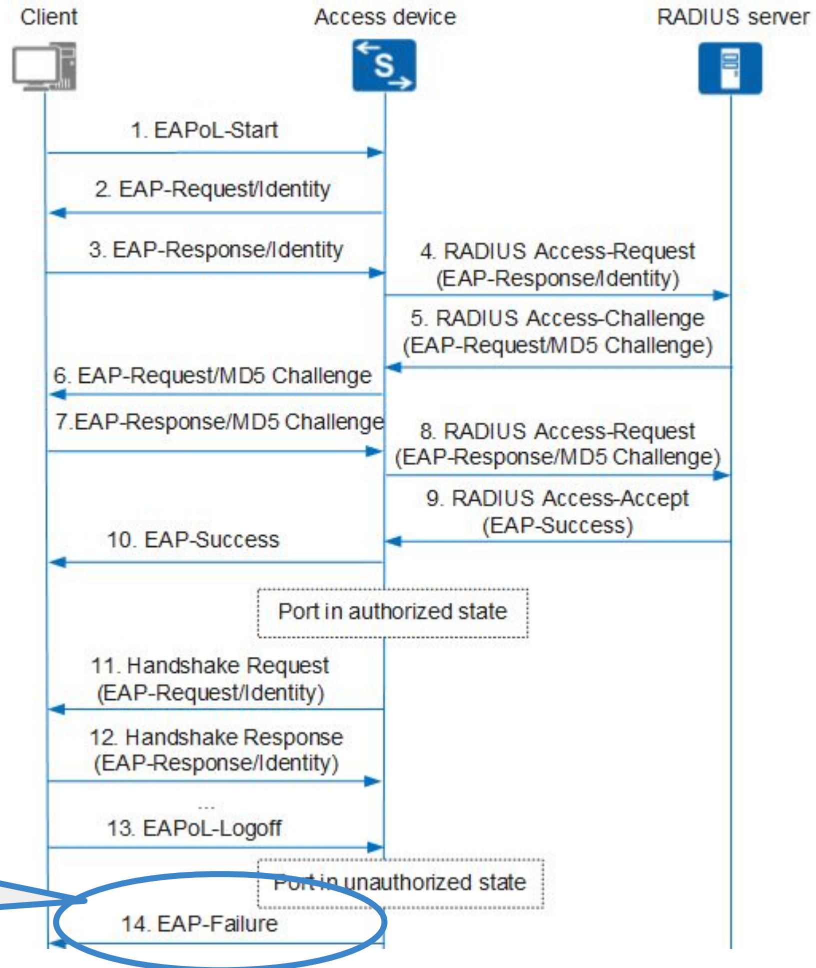


Relay Mode (MD5-challenge)



To go offline, the client sends an ***EAPoL-Logoff*** packet to the access device.

Relay Mode (MD5-challenge)



The access device changes the port state from authorized to unauthorized and sends an ***EAP-Failure*** packet to the client.

a more complex example: EAP-TLS

from RFC 5216

with 802.1x the handshake starts with a EAPOL-start

In the case where the server authenticates to the peer successfully, but the peer fails to authenticate to the server, the conversation will appear as follows:

```
Authenticating Peer     Authenticator
-------------------     -------------
                        <- EAP-Request/
                        Identity
EAP-Response/
Identity (MyID) ->
                        <- EAP-Request/
                        EAP-Type=EAP-TLS
                        (TLS Start)
EAP-Response/
EAP-Type=EAP-TLS
(TLS client_hello)->
                        <- EAP-Request/
                        EAP-Type=EAP-TLS
                        (TLS server_hello,
                          TLS certificate,
                 [TLS server_key_exchange,]
           TLS certificate_request,
             TLS server_hello_done)

EAP-Response/
EAP-Type=EAP-TLS
(TLS certificate,
 TLS client_key_exchange,
 TLS certificate_verify,
 TLS change_cipher_spec,
 TLS finished) ->

                        <- EAP-Request/
                        EAP-Type=EAP-TLS
                        (TLS change_cipher_spec,
                        TLS finished)
EAP-Response/
EAP-Type=EAP-TLS ->
                        <- EAP-Request
                        EAP-Type=EAP-TLS
                        (TLS Alert message)
EAP-Response/
EAP-Type=EAP-TLS ->
                        <- EAP-Failure
                        (User Disconnected)
```

Additional Operations: Re-Authentication

Re-authentication for 802.1X-authenticated Users

- re-authentication if parameters changed
- client sends back user authentication parameters for re-auth
- If the user authentication information on the authentication server remains unchanged, the user keeps online.
  - If the information has been modified, the user is disconnected and needs to be re-authenticated

Re-authentication for Users in Abnormal Authentication State

- users in pre-connection state
- re-authenticate the users based on user entries to access the network quickly
- If a user fails the re-authentication before the user entry aging time expires, the access device deletes the user entry and reclaims the granted network access rights.
- If a user is successfully re-authenticated before the user entry aging time expires, the access device adds a user-authenticated entry and grants corresponding network access rights to the user.

Additional Operations: Log out and Timers

- ***When users go offline but the access device and RADIUS server do not detect the offline events, the following problems may occur:***
  - The RADIUS server still performs accounting for the users, causing incorrect accounting.
  - Unauthorized users may spoof IP addresses and MAC addresses of authorized users to access the network.
  - If there are many offline users, these users are still counted as access users of the device. As a result, other users may fail to access the network.
- The access device needs to detect user logout immediately, delete the user entry, and instruct the RADIUS server to stop accounting.
- A user may log out in the following scenarios: The user proactively logs out on the client, the access device controls user logout, and the RADIUS server logs out the user.
- ***802.1X relies on several timers to control the number of packet retransmission times and timeout interval***.

802.1X Authorization: VLAN

To prevent unauthenticated users from accessing restricted network resources, the restricted network resources and unauthenticated users are allocated to different VLANs. ***After a user is authenticated, the authentication server returns an authorized VLAN to the user.***

The access device then changes the VLAN to which the user belongs to the authorized VLAN, with the interface configuration remaining unchanged. The authorized VLAN takes precedence over the VLAN configured on the interface.

That is, the authorized VLAN takes effect after the authentication succeeds, and the configured VLAN takes effect after the user goes offline. When the RADIUS server assigns an authorized VLAN, the following standard RADIUS attributes must be used together:

- Tunnel-Type: This attribute must be set to VLAN or 13.
- Tunnel-Medium-Type: This attribute must be set to 802 or 6.
- Tunnel-Private-Group-ID: The value is the assigned VLAN ID.

802.1X Authorization: ACL

- After a user is authenticated, the authentication server assigns an ACL to the user. Then, the access device controls the user packets according to the ACL.
  - If the user packets match the permit rule in the ACL, the packets are allowed to pass through.
  - If the user packets match the deny rule in the ACL, the packets are discarded.
- ***The RADIUS server can assign an ACL to a user in either of the following modes:***
  - ***Static ACL assignment***: The RADIUS server uses the standard RADIUS attribute Filter-Id to assign an ACL ID to the user. In this mode, the ACL and corresponding rules are configured on the access device in advance.
  - ***Dynamic ACL assignment***: The RADIUS server uses the RADIUS attribute HW-Data-Filter extended by Huawei to assign an ACL ID and corresponding rules to the user. In this mode, the ACL ID and ACL rules are configured on the RADIUS server.

802.1X Authorization: UCL

A ***User Control List (UCL) is a collection of network terminals*** such as PCs and smartphones. ***The administrator can add users having the same network access requirements to a UCL, and configure a network access policy for the UCL***, greatly reducing the administrator's workload.

The RADIUS server assigns a UCL to a specified user in either of the following modes:

- ❏ Assigns the UCL name through the standard RADIUS attribute Filter-Id.
- ❏ Assigns the UCL ID through the RADIUS attribute HW-UCL-Group extended by Huawei.
- ❏ You must configure the UCL and its network access policy on the access device in advance regardless of the UCL authorization mode used

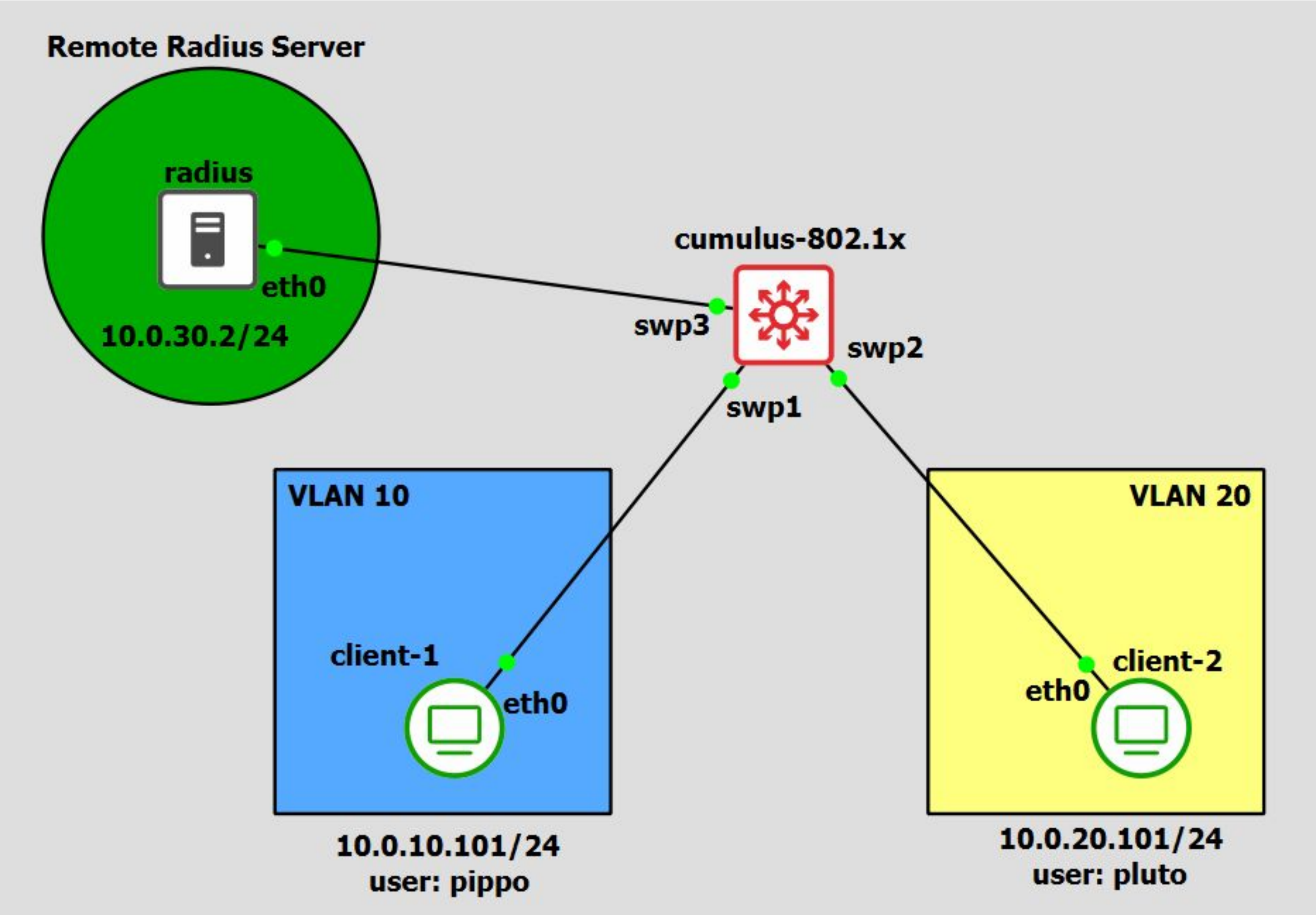
Dot1x vulnerabilities

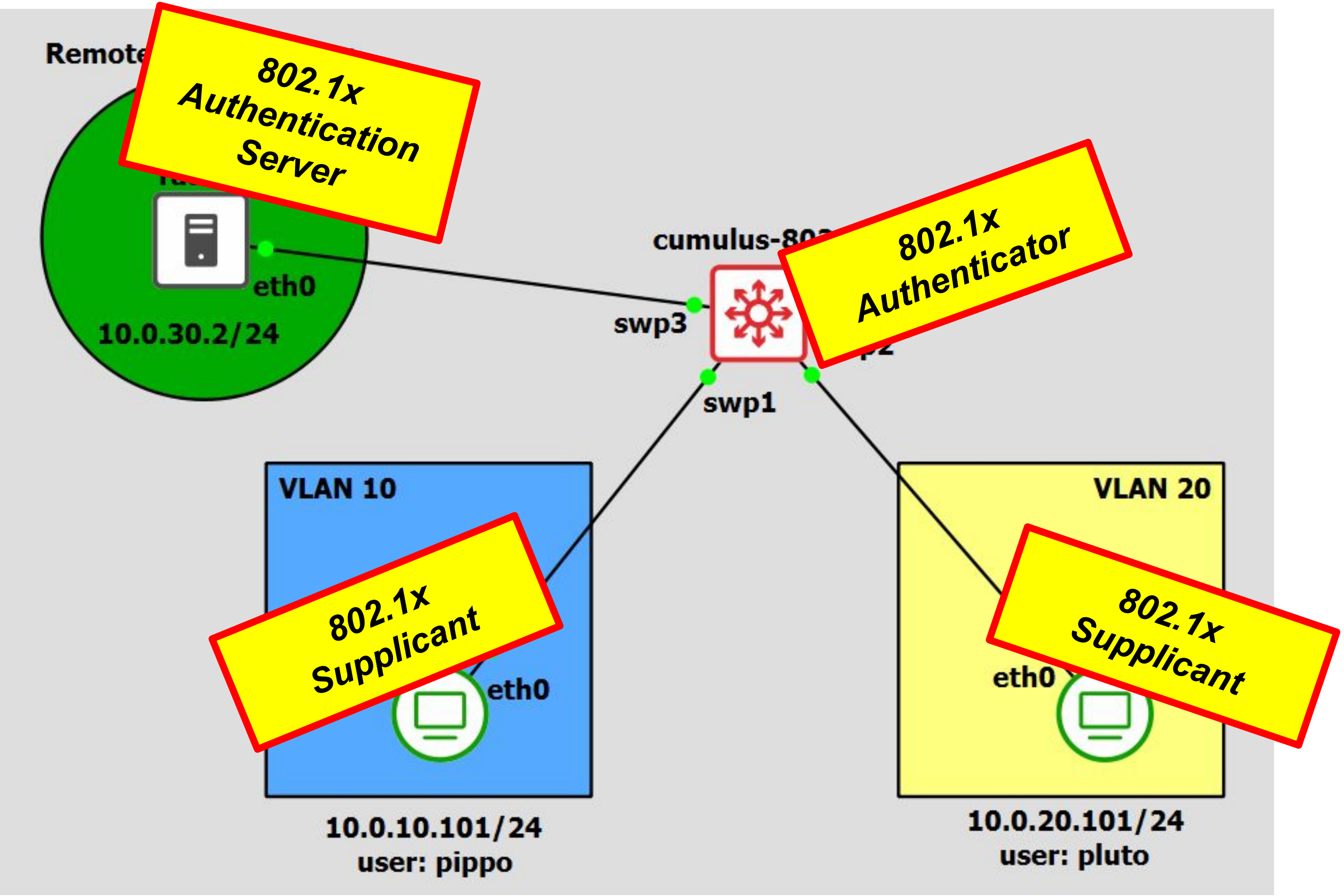
- If an attacker is able to physically intercept packets from a station connected to an authorize port (e.g. attacker and legitimate user connected to the same hub – MitM), the attacker can:
  - spoof the MAC/IP of the authorized user, thus accessing the medium
  - perform a DoS attack by crafting rogue EAPoL-Logoff packets with the victim's MAC address
    - EAPoL-Logoff messages are cleartext, everyone can spoof them
- MACsec and MACsec Key Agreement were added to the standard to overcome this vulnerability

Lab 5: 802.1x Port-Based Authentication and VLAN assignment

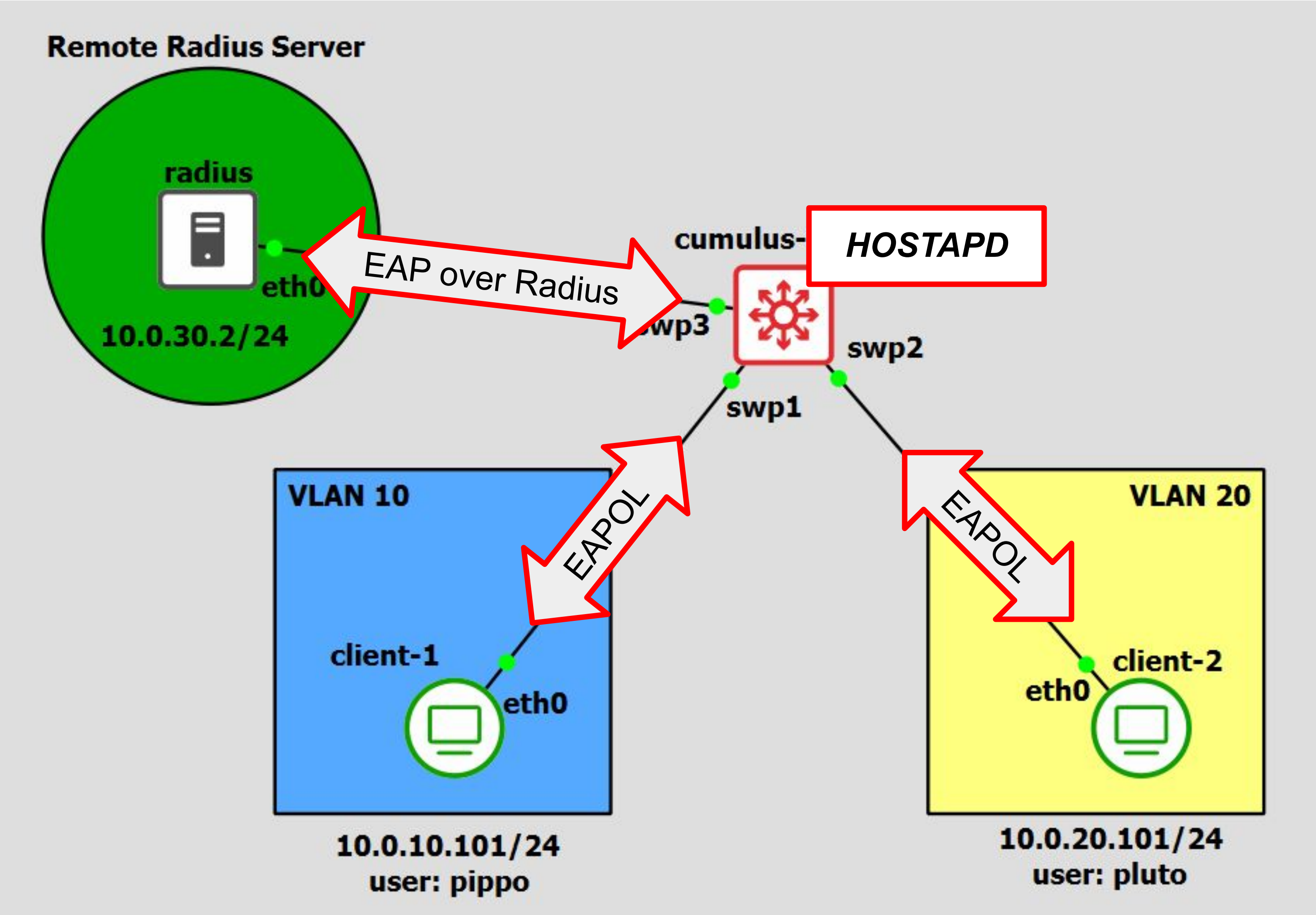
Overview

- In this lab we demonstrate the use of 802.1x with EAP-MD5 to authenticate a user that wants to connect to a VLAN
  - unauthenticated users cannot send traffic (802.1x port authorization)
  - the switch dynamically assign the VLAN to the relative access port according to the authentication server configuration
- ***Lab components***
  - 2 hosts that act as 802.1x supplicants
  - 1 802.1x Authentication Server (Linux freeradius) in VLAN 1
  - 1 cumulus switch that acts as a 802.1x Authenticator + L3 FWD
- **REQUIRES** cumulus linux version <= 4.3
  - from 4.4 on, the dot1x feature have been removed :(


Remote Radius Server
radius
eth0
10.0.30.2/24
cumulus-802.1x
swp3
swp2
swp1
VLAN 10
client-1
eth0
10.0.10.101/24
user: pippo
VLAN 20
client-2
eth0
10.0.20.101/24
user: pluto

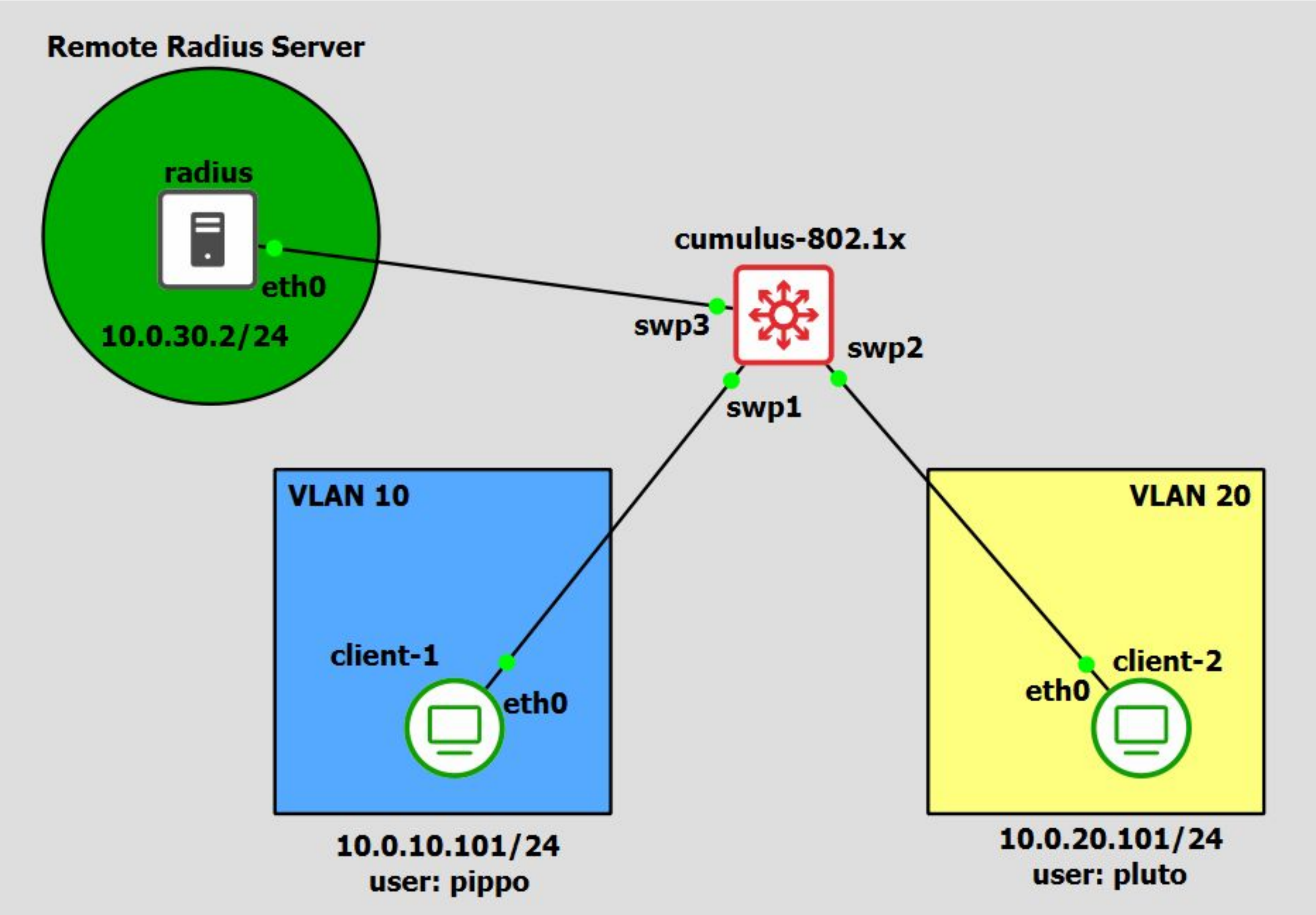


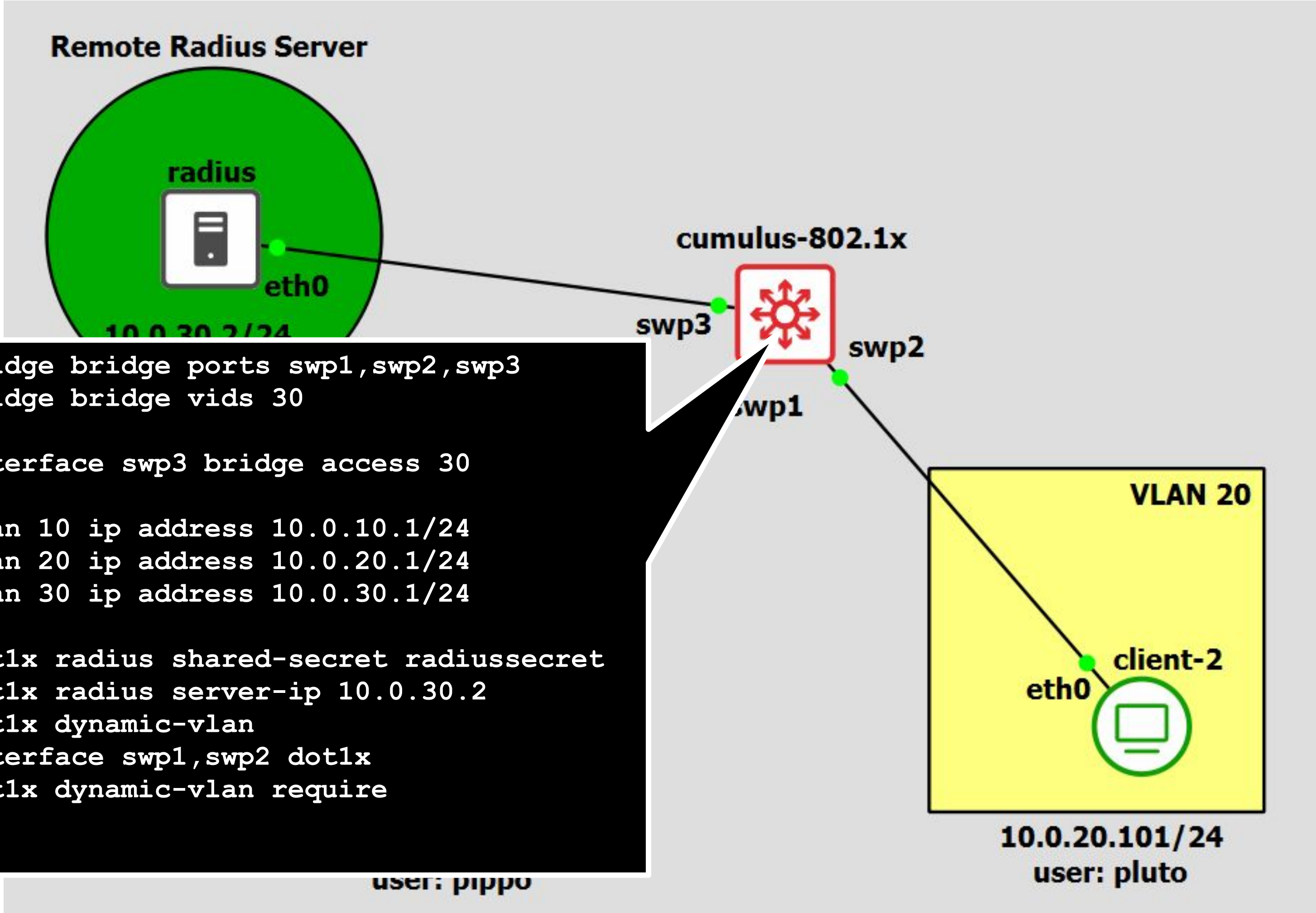
Remote
802.1x Authentication Server
eth0
10.0.30.2/24
cumulus-80
802.1x Authenticator
swp3
swp1
VLAN 10
802.1x Supplicant
eth0
10.0.10.101/24
user: pippo
VLAN 20
802.1x Supplicant
eth0
10.0.20.101/24
user: pluto


Remote Radius Server
radius
eth0
10.0.30.2/24
EAP over Radius
cumulus-
HOSTAPD
swp3
swp2
swp1
VLAN 10
EAPOL
client-1
eth0
10.0.10.101/24
user: pippo
VLAN 20
EAPOL
client-2
eth0
10.0.20.101/24
user: pluto

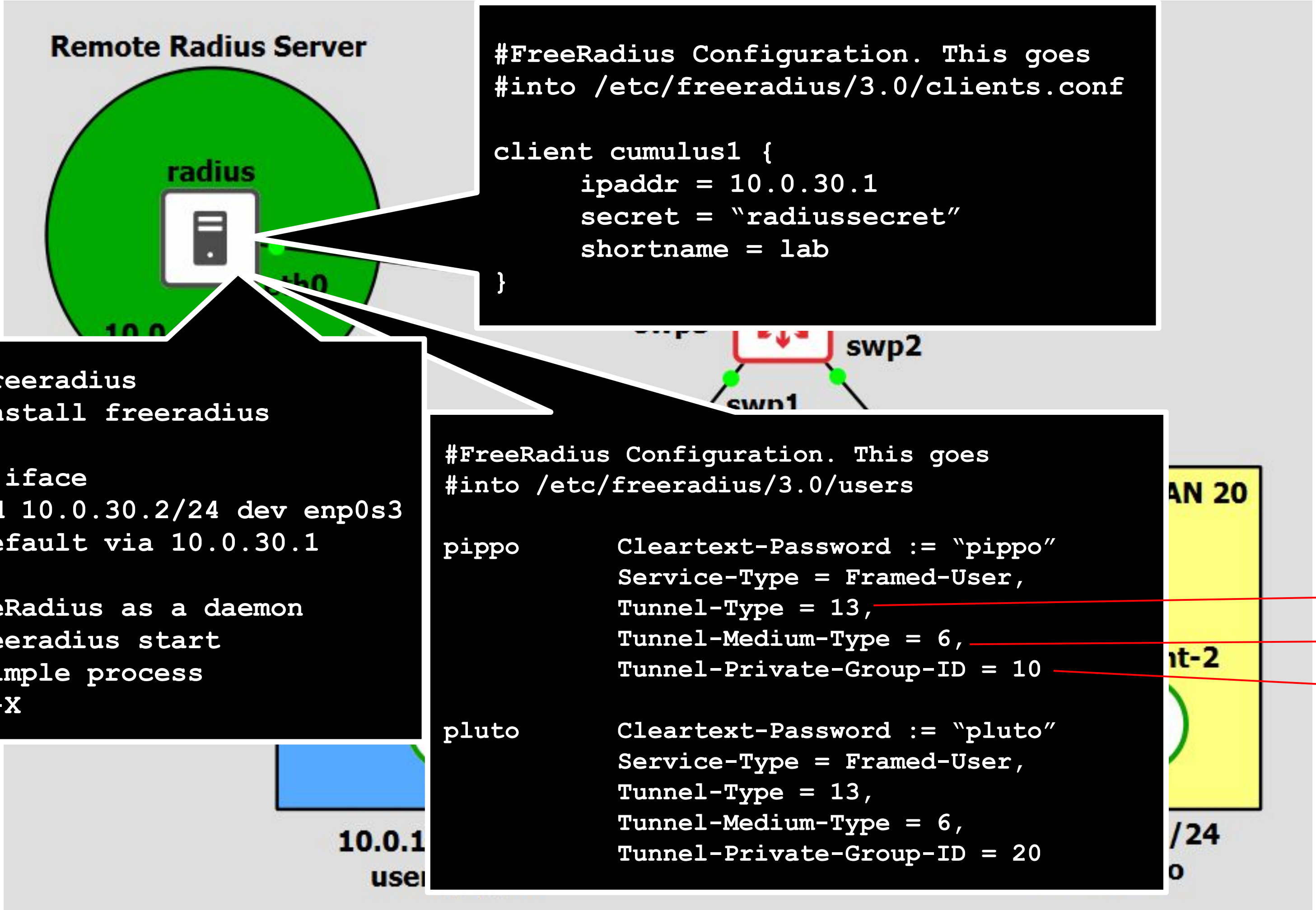
Tasks

1. Configure the L3 switch so that
    a. swp3 statically configured as access port for VLAN 30
    b. swp1, swp2, configured with 802.1x port authorization
        i. for simplicity, the auth method is EAP-MD5
        ii. the Authentication Server is in VLAN 30
    c. swp1, swp2: access port with dynamic VLAN assignment according to the user name
        i. pippo: VLAN 10
        ii. pluto: VLAN 20
        iii. IP forwarding enabled (it requires a vlan interface for each VLAN)
    d. configure VLAN IP interfaces (required to communicate with the RADIUS server in VLAN 30 and for inter-VLAN IP forwarding - if needed)
2. Configure radius
    a. to run freeradius for the above mentioned configuration
3. Configure the 2 clients
    a. LAN access via 802.1x authentication
    b. static IP configuration (for simplicity. Homework: configure dynamic IP configuration with DHCP)


Remote Radius Server
radius
eth0
10.0.30.2/24
cumulus-802.1x
swp3
swp2
swp1
VLAN 10
client-1
eth0
10.0.10.101/24
user: pippo
VLAN 20
client-2
eth0
10.0.20.101/24
user: pluto




```
net add bridge bridge ports swp1,swp2,swp3
net add bridge bridge vids 30

net add interface swp3 bridge access 30

net add vlan 10 ip address 10.0.10.1/24
net add vlan 20 ip address 10.0.20.1/24
net add vlan 30 ip address 10.0.30.1/24

net add dot1x radius shared-secret radiussecret
net add dot1x radius server-ip 10.0.30.2
net add dot1x dynamic-vlan
net add interface swp1,swp2 dot1x
net add dot1x dynamic-vlan require

net commit
```

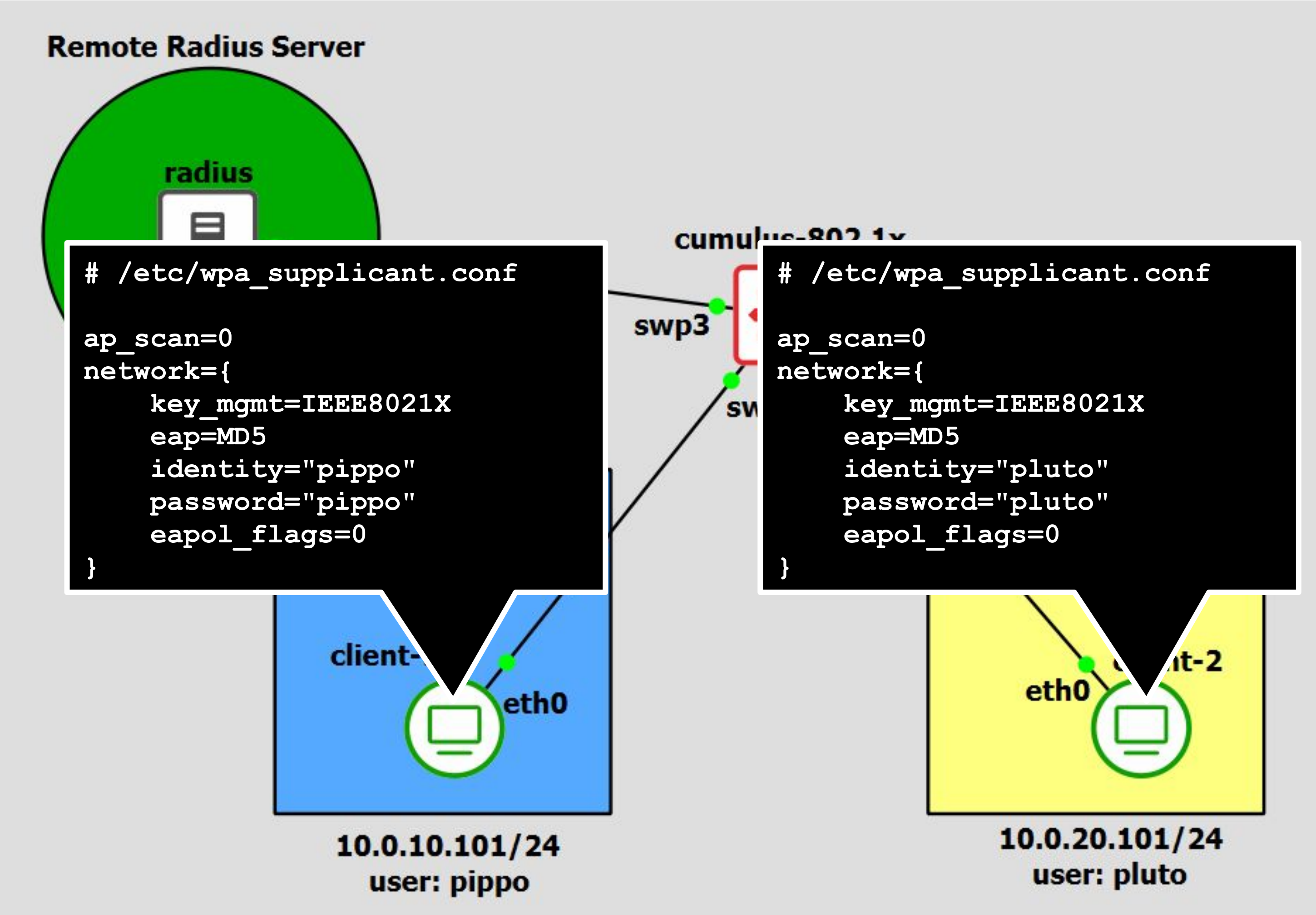


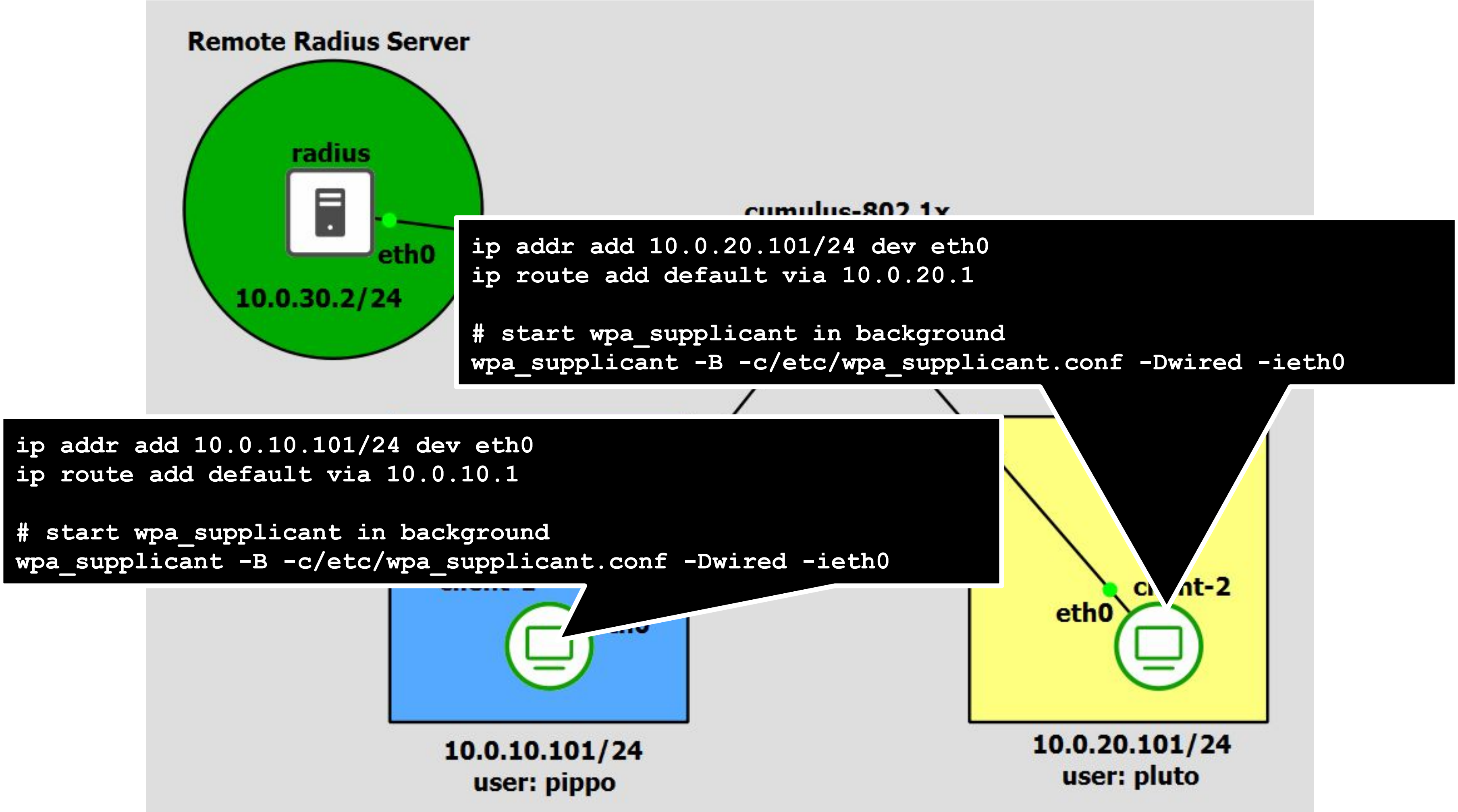

```
#FreeRadius Configuration. This goes
#into /etc/freeradius/3.0/clients.conf

client cumulus1 {
        ipaddr = 10.0.30.1
        secret = "radiussecret"
        shortname = lab
}
```

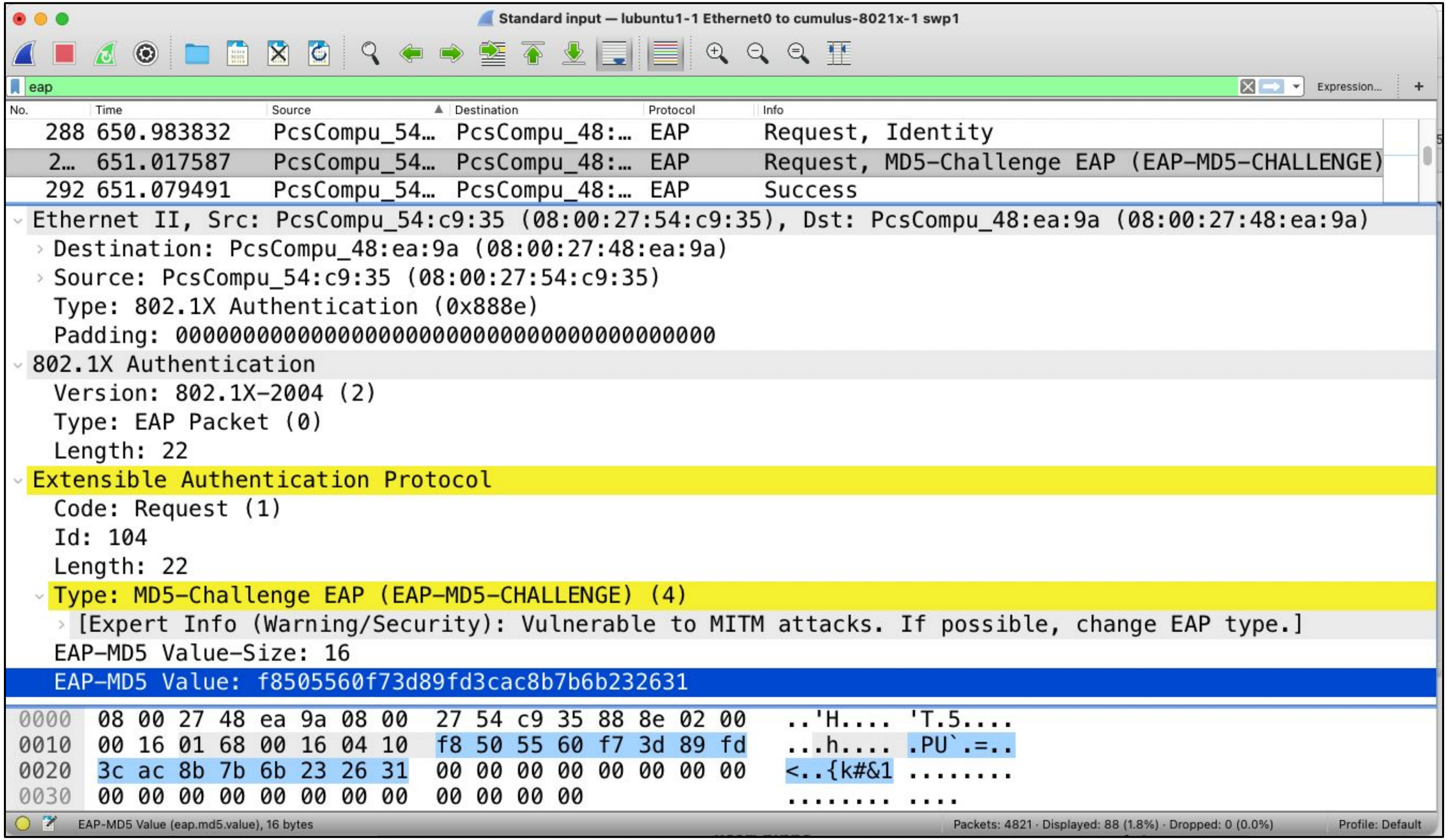
```
#install freeradius
sudo apt install freeradius

#configure iface
ip addr add 10.0.30.2/24 dev enp0s3
ip r add default via 10.0.30.1

#start FreeRadius as a daemon
service freeradius start
#or as a simple process
freradius -X
```

```
#FreeRadius Configuration. This goes
#into /etc/freeradius/3.0/users

pippo       Cleartext-Password := "pippo"
            Service-Type = Framed-User,
            Tunnel-Type = 13,
            Tunnel-Medium-Type = 6,
            Tunnel-Private-Group-ID = 10

pluto       Cleartext-Password := "pluto"
            Service-Type = Framed-User,
            Tunnel-Type = 13,
            Tunnel-Medium-Type = 6,
            Tunnel-Private-Group-ID = 20
```

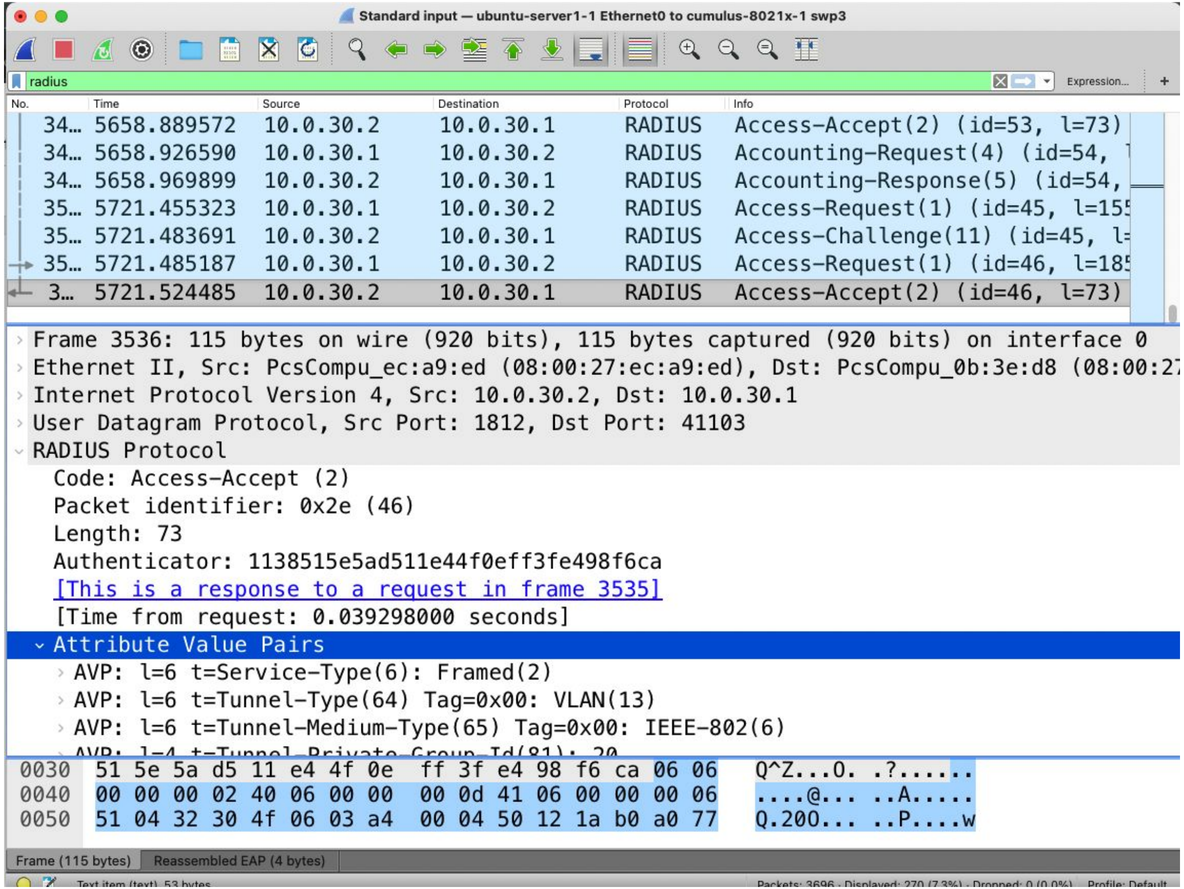
VLAN
IEEE-802
VLAN id


Remote Radius Server
radius
swp3
# /etc/wpa_supplicant.conf
ap_scan=0
network={
key_mgmt=IEEE8021X
eap=MD5
identity="pippo"
password="pippo"
eapol_flags=0
}
# /etc/wpa_supplicant.conf
ap_scan=0
network={
key_mgmt=IEEE8021X
eap=MD5
identity="pluto"
password="pluto"
eapol_flags=0
}
eth0
eth0
10.0.10.101/24
user: pippo
10.0.20.101/24
user: pluto


Remote Radius Server
radius
eth0
10.0.30.2/24
cumulus-802.1x
ip addr add 10.0.20.101/24 dev eth0
ip route add default via 10.0.20.1
start wpa_supplicant in background
wpa_supplicant -B -c/etc/wpa_supplicant.conf -Dwired -ieth0
ip addr add 10.0.10.101/24 dev eth0
ip route add default via 10.0.10.1
start wpa_supplicant in background
wpa_supplicant -B -c/etc/wpa_supplicant.conf -Dwired -ieth0
eth0
10.0.10.101/24
user: pippo
10.0.20.101/24
user: pluto



802.1x (EAPoL) exchange between the supplicant (the client) and the authenticator (the switch)



RADIUS (EAPoR) exchange and the authenticator (the switch) and the authentication server (RADIUS)

```
cumulus@cumulus-8021x:mgmt:~$ net show dot1x interface swp1 detail
s
Interface  MAC Address        Attribute                       Value
---------  -----------------  ------------------------------  ---------------------------
swp1       08:00:27:48:ea:9a  Status Flags                    [DYNAMIC_VLAN][AUTHORIZED]
                              Username                        pippo
                              Authentication Type             MD5
                              VLAN                            10
                              Dynamic ACL Filename
                              Session Time (seconds)          1898
                              EAPOL Frames RX                 3
                              EAPOL Frames TX                 3
                              EAPOL Start Frames RX           1
                              EAPOL Logoff Frames RX          0
                              EAPOL Response ID Frames RX     1
                              EAPOL Response Frames RX        2
                              EAPOL Request ID Frames TX      1
                              EAPOL Request Frames TX         2
                              EAPOL Invalid Frames RX         0
                              EAPOL Length Error Frames Rx    0
                              EAPOL Frame Version             1
                              EAPOL Auth Last Frame Source    08:00:27:48:ea:9a
                              EAPOL Auth Backend Responses    2
                              RADIUS Auth Session ID          3E4A87DC53B05DE7
cumulus@cumulus-8021x:mgmt:~$
```

```
cumulus@cumulus-8021x:mgmt:~$ net show dot1x interface summary

Interface  MAC Address        Username  State       Authentication Type  MAB  VLAN  DACL Active
---------  -----------------  --------  ----------  -------------------  ---  ----  -----------
swp1       08:00:27:48:ea:9a  pippo     AUTHORIZED  MD5                  NO     10  NO
swp2       08:00:27:48:ea:9b  pluto     AUTHORIZED  MD5                  NO     20  NO
```

802.1X-2010 MACsec Key Agreement

802.1X-2010 extensions – MACsec Key Agreement

- 802.1X-2010 revision *supersedes* 802.1X-2004 by adding MACsec Key Agreement (MKA) protocol
  - used to determine MACsec Secure Association Keys (SAK) between stations with the same Connectivity Association Key (CAK)
- **CAK:** pre-shared key used as source keying material for message integrity checking and SAK distribution – initial secret for the stations in the LAN
  - can be configured manually (pre-shared key) – ***static CAK mode***
  - or with 802.1X/EAP authentication methods – ***dynamic CAK mode***
    - **ICK:** integrity check key – derived from CAK and used for integrity checking
    - **KEK:** key encryption key – derived from CAK and used to encrypt SAK distribution
- **SAK:** key used for unidirectional secure channels (like the two keys we statically configured in the MACsec Lab for TX and RX)
  - generated by a **Key Server** selected with an election mechanism

Static CAK mode

- In static CAK mode the CAK is *pre-shared* among all the MKA stations in the LAN
    - Then CAK is used to *protect* the control plane communication
    - and SAKs are randomly derived to encrypt the MACsec data exchange
- Once the stations successfully exchange the pre-shared keys, the MKA protocol can be enabled on the interfaces

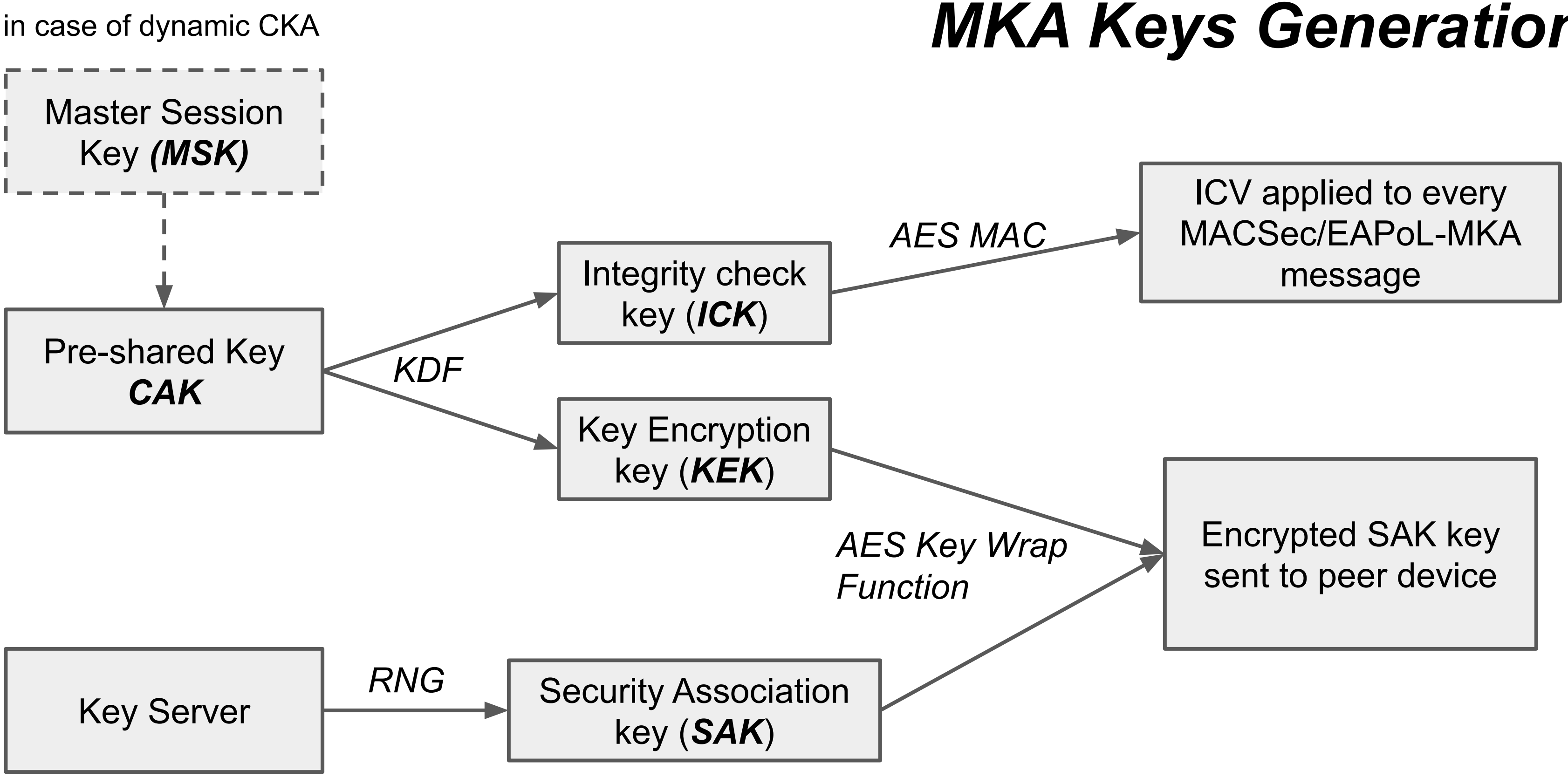
- Usually this mode is used in switch-to-switch links or switch-to-router links

Dynamic CAK mode

- In dynamic CAK mode, the stations derive the CAK as part of the 802.1x authentication process
- Each authenticated station, receivers by the (radius) authentication server a Master Session Key (MSK)
  - then each station derives the CAK (and CAK name) from the received MSK
- Then the SAKs are randomly generated in the same way

- Usually employed in host-to-switch links
  - if in switch-to-switch links, nodes must act both as authenticators and supplicants, to authenticate each other

MKA Keys Generation



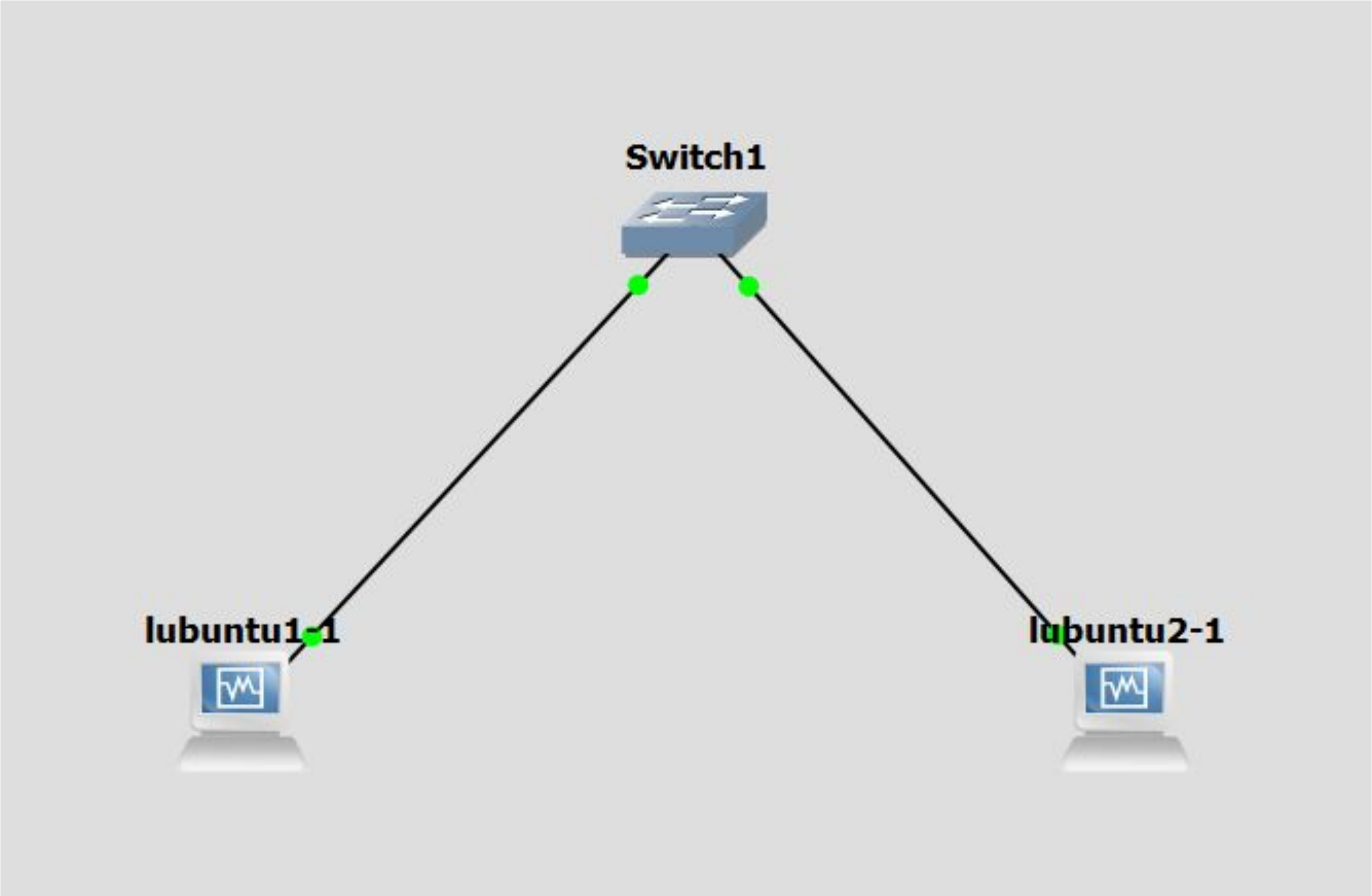
RNG: Random Number Generator
KDF: Key Derivation Function

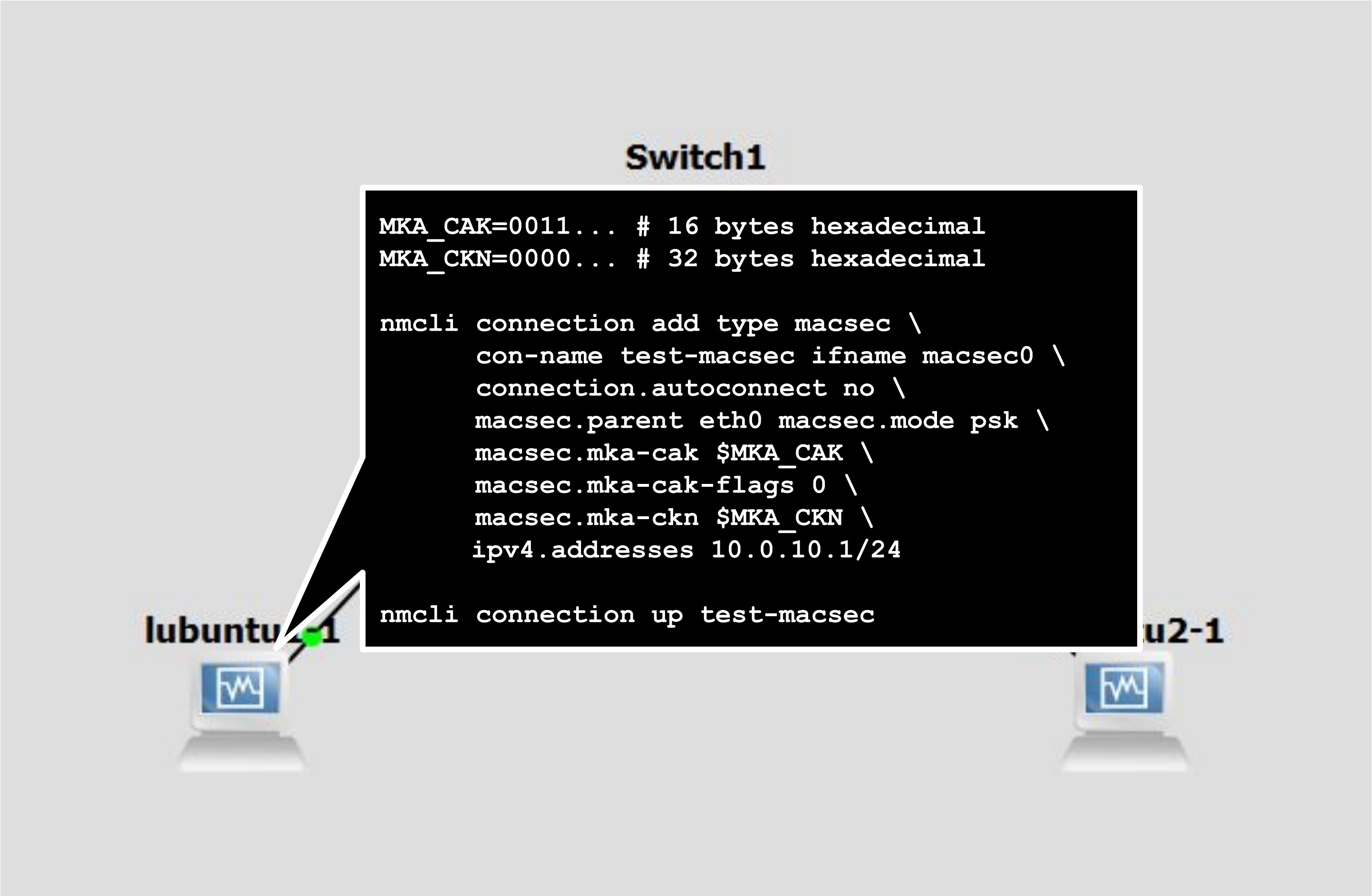
MKA protocol at a glance

One station must be elected as the ***Key Server***, responsible for distributing SAKs (every station is configured to be a possible Key Server for reliability)

- Every station in the LAN broadcasts “heartbeat” messages containing:
  - MACSec capabilities
  - Key Server priority (generally the switch is configured with the highest priority)
  - Anti-replay information (a list of “live” or “potentially live” stations)
- A simple election process is used to elect the Key Server based on priority
  - once all stations agreed on the list of “live” stations, the Key Server is elected
- The Key Server distributes the Secure Association Keys
- Secure Channel is established and encrypted communication can start
- If a station doesn’t send keepalive messages after a predetermined timeout, the security association is canceled

Simple MKA lab with Linux


Switch1
lubuntu1-1
lubuntu2-1




```
MKA_CAK=0011... # 16 bytes hexadecimal
MKA_CKN=0000... # 32 bytes hexadecimal

nmcli connection add type macsec \
      con-name test-macsec ifname macsec0 \
      connection.autoconnect no \
      macsec.parent eth0 macsec.mode psk \
      macsec.mka-cak $MKA_CAK \
      macsec.mka-cak-flags 0 \
      macsec.mka-ckn $MKA_CKN \
      ipv4.addresses 10.0.10.1/24

nmcli connection up test-macsec
```


```
MKA_CAK=0011... # 16 bytes hexadecimal
MKA_CKN=0000... # 32 bytes hexadecimal

nmcli connection add type macsec \
      con-name test-macsec ifname macsec0 \
      connection.autoconnect no \
      macsec.parent eth0 macsec.mode psk \
      macsec.mka-cak $MKA_CAK \
      macsec.mka-cak-flags 0 \
      macsec.mka-ckn $MKA_CKN \
      ipv4.addresses 10.0.10.2/24

nmcli connection up test-macsec
```

EAPOL-MKA Key Server announcement

Applica un filtro di visualizzazione ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	PcsCompu_7a…	Nearest-non…	EAPOL-MKA	98	Key Server
2	2.001878	PcsCompu_7a…	Nearest-non…	EAPOL-MKA	98	Key Server
3	2.974594	PcsCompu_21…	Nearest-non…	EAPOL-MKA	118	Key Server, Potential Peer List

```
> Ethernet II, Src: PcsCompu_7a:ca:24 (08:00:27:7a:ca:24), Dst: Nearest-non-TPMR-bridge (01:80:c2:00:00:03)
v 802.1X Authentication
    Version: 802.1X-2010 (3)
    Type: MKA (5)
    Length: 80
v MACsec Key Agreement
  v Basic Parameter set
      MKA Version Identifier: 1
      Key Server Priority: 255
      1... .... = Key Server: True
      .1.. .... = MACsec Desired: True
      ..10 .... = MACsec Capability: MACsec Integrity with no confidentiality offset (2)
      .... 0000 0011 1100 = Parameter set body length: 60
      SCI: 0800277aca240001
      Actor Member Identifier: fec743b0cfd003e09936f95a
      Actor Message Number: 00000002
      Algorithm Agility: IEEE Std 802.1X-2010 (0x0080c201)
      CAK Name: 0000000000000000000000000000000000000000000000000000000000000000
    Integrity Check Value: 2def27e6dd8079cb3254c7be2462f61c
```

EAPOL-MKA election process

No.	Time	Source	Destination	Protocol	Length	Info
2	2.001878	PcsCompu_7a…	Nearest-non…	EAPOL-MKA	98	Key Server
3	2.974594	PcsCompu_21…	Nearest-non…	EAPOL-MKA	118	Key Server, Potential Peer List
4	4.004011	PcsCompu_7a…	Nearest-non…	EAPOL-MKA	118	Live Peer List

```
> Frame 4: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface -, id 0
> Ethernet II, Src: PcsCompu_7a:ca:24 (08:00:27:7a:ca:24), Dst: Nearest-non-TPMR-bridge (01:80:c2:00:00:03)
> 802.1X Authentication
v MACsec Key Agreement
  > Basic Parameter set
  v Live Peer List Parameter set
      Parameter set type: Live Peer List (1)
      .... 0000 0001 0000 = Parameter set body length: 16
      Peer Member Identifier: e25022b4972e82cc01879ee5
      Peer Message Number: 00000001
    Integrity Check Value: e4697816b9016d3ffdea7e990078cf1b
```

EAPOL-MKA SAK distribution

No.	Time	Source	Destination	Protocol	Length	Info
4	4.004011	PcsCompu_7a…	Nearest-non…	EAPOL-MKA	118	Live Peer List
5	4.975305	PcsCompu_21…	Nearest-non…	EAPOL-MKA	150	Key Server, Live Peer List, Distributed SAK
6	4.975305	PcsCompu_21…	Nearest-non…	EAPOL-MKA	194	Key Server, Live Peer List, MACsec SAK Use,

```
> Frame 6: 194 bytes on wire (1552 bits), 194 bytes captured (1552 bits) on interface -, id 0
> Ethernet II, Src: PcsCompu_21:aa:f8 (08:00:27:21:aa:f8), Dst: Nearest-non-TPMR-bridge (01:80:c2:00:00:03)
> 802.1X Authentication
v MACsec Key Agreement
  > Basic Parameter set
  > Live Peer List Parameter set
  > MACsec SAK Use parameter set
  v Distributed SAK parameter set
      Parameter set type: Distributed SAK (4)
      00.. .... = Distributed AN: 0
      ..01 .... = Confidentiality Offset: No confidentiality offset (1)
      .... 0000 0001 1100 = Parameter set body length: 28
      Key Number: 00000001
      AES Key Wrap of SAK: f71f01c952a83823d8433defbc3b184d3c5725e1143cbfff
    Integrity Check Value: 98db6ee5cc7bdb9de598a4078a0417a6
```

Suggested Homework

- ❑ Add another lubuntu VM to the LAN, participating to the MACsec Key Agreement

Recap: Ethernet insecurity and countermeasures

Ethernet Security Recap

- ***Different default insecure behaviour***
  - authentication: unauthorized joins, port stealing, MAC flooding
  - confidentiality: no encryption, simple hijacking
  - message integrity: DHCP spoofing, STP spoofing, ARP spoofing
- ***Countermeasures***
  - Physical protection
  - Port security
  - L2 ACLs
  - Port Authentication (802.1x)
  - LAN segmentation (VLANs)
  - VLAN segmentation (Private VLANs)
  - L2 encryption/integrity/anti replay (MACsec)
  - Security protocols at upper layers (S-ARP, IPv6 SeND)