Grandstream Networks, Inc.

802.1x Authentication Guide

Table of Content

Table of figures

SUPPORTED DEVICES

Following table shows Grandstream products supporting 802.1x feature:

Model	Supported	Firmware
GXP16XX Series		
GXP1610	Yes	1.0.3.28 or higher
GXP1620/25	Yes	1.0.3.28 or higher
GXP1628	Yes	1.0.3.28 or higher
GXP1630	Yes	1.0.3.28 or higher
GXP21XX Series		
GXP2130	Yes	1.0.7.25 or higher
GXP2140	Yes	1.0.7.25 or higher
GXP2160	Yes	1.0.7.25 or higher
GXP2135	Yes	1.0.7.25 or higher
GXP2170	Yes	1.0.7.25 or higher
GXV32XX Series		
GXV3240	Yes	1.0.3.92 or higher
GXV3275	Yes	1.0.3.92 or higher

INTRODUCTION

IEEE 802.1x is a standard for port-based Network Access Control (PNAC), designed to provide an authentication mechanism for network devices to connect to LAN or WAN. The IEEE 802.1x protocol includes the encapsulation of the Extensible Authentication Protocol (EAP) over LAN (known as “EAPOL” or “EAP over LAN”) for messages exchange during the authentication process.

802.1x is implemented to accommodate the following:

- Authentication based on Network Access Identifier and credentials.
- Centralized authentication, authorization and accounting.
- Public Network Security.
- Distribution of dynamic encryption keys.

This guide will outline the use and configuration of 802.1x authentication on Grandstream IP phones.

802.1X AUTHENTICATION PROCESS

802.1x Elements

The main elements interacting in 802.1x process are:

- **Supplicant:** PC, Laptop, IP phones and any other device aiming to connect to the network.
- **Authenticator:** Network switches/Wireless access points providing first connection level to supplicants.
- **Authentication server:** Server or device (Radius server, AD etc…) that can hold authentication credentials for all users and verify provided information exchanged by end points during authentication process.

Authentication Process

Please refer to following process describing how IEEE 802.1x operates:

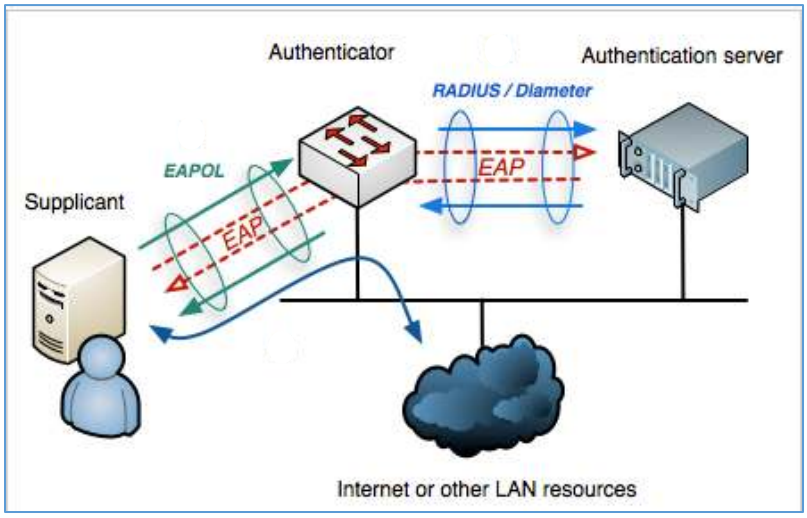


Figure 1: 802.1x Authentication Process

1. The client/supplicant sends an EAP-start message and a series of messages are exchanged to authenticate the client.
2. The access point forwards the EAP-request identity message.
3. The client sends an EAP-response packet that contains his identity to the authentication server.
4. The server uses a specific authentication algorithm to verify the client's identity.
5. The authentication server sends either an accept message or a reject message to the access point.
6. The access point sends an EAP-success packet or reject packet to the client.
7. When authentication server accepts the client, the access point transits the client's port to an authorized state and forwards additional traffic.

Additional information:

a) All messages between supplicant and authenticator are delivered in **EAPOL (Extensible Authentication Protocol Over LAN)** form.

b) Authenticator converts messages to RADIUS messages and send them to RADIUS server (Authentication Server)

c) Authentication Server negotiates the type of EAP authentication that is acceptable to both the supplicant and itself and starts communicating with the supplicant via EAP messages to carry out the authentication process. Some authenticators may not support all types of EAP and hence would act as an EAP pass through where the supplicants directly communicate with Authentication Servers to complete the authentication process.

Note:
Before the authentication happens, the authenticator sets the network port to the **Uncontrolled State** where only EAP / EAPOL messages are allowed to pass through between the supplicant and the authentication server. All other traffic remains blocked from that network port. But after the authentication, the network port is set to **Controlled/Authorized State** to grant network access according to the NAC policies.

Authentication Flowchart

Please refer to following diagram describing and summarizing an implementation of 802.1x authentication process:

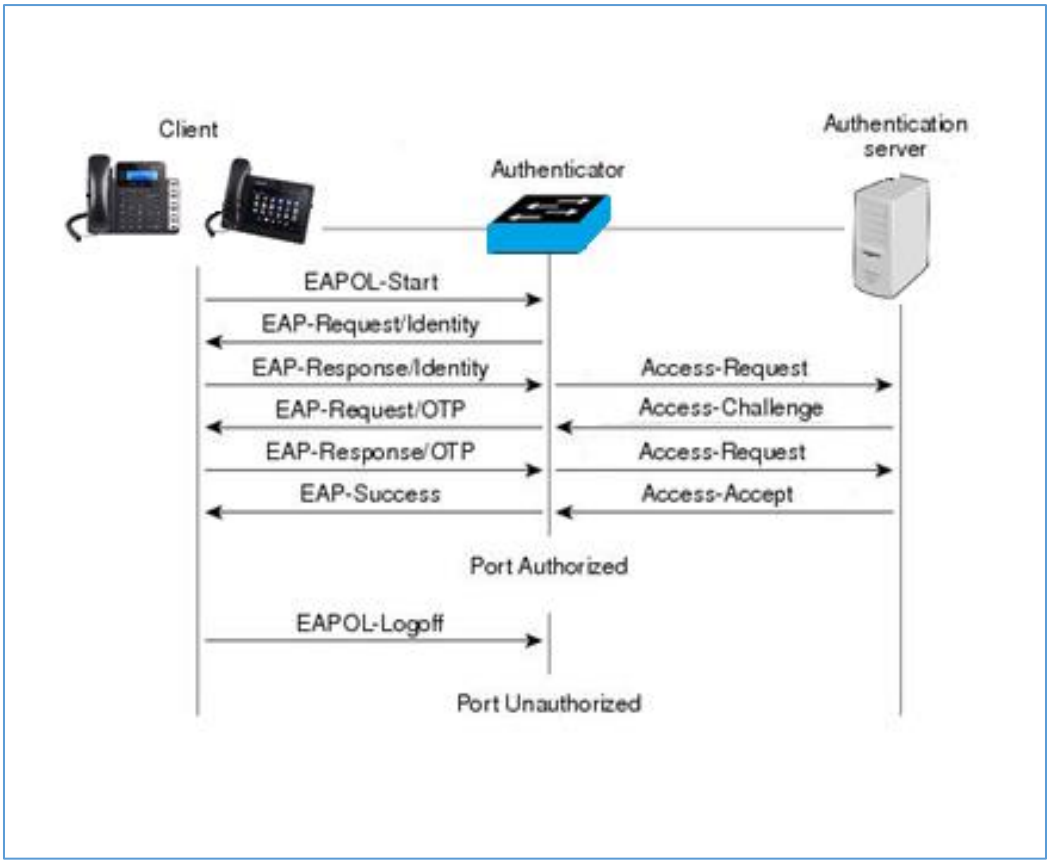


Figure 2: 802.1x Authentication Flowchart

EAP Methods for 802.1x Authentication

The following are the main types of EAP protocol for the 802.1x authentication supported:

- **EAP-MD5 (EAP-Message Digest 5):** Basic authentication method using Username/Password combination to verify authentication credentials and offering basic protection for the messages exchanged. This type offers lowest security level and can be used in wired networks only requiring basic security.
- **EAP-TLS (EAP-Transport Level Security):** Client and Server authentication need to have pre-installed certificates to be authenticated, since those certificates are required and TLS tunnel is created between the authentication server and client. This method is more secure and can be used for wired and wireless networks.
- **EAP-PEAPv0/MSCHAPv2:** The most common method form of PEAP; MSCHAP (Microsoft Challenge Handshake Authentication Protocol) allows authentication to databases supporting MS-CHAPv2 format, including Microsoft NT and Microsoft Active Directory and using a CA certificate at each client to authenticate with the server. It's a mutual authentication method that supports password-based user or computer authentication. During the authentication process, both the server and client must prove that they have knowledge of the user's password in order for authentication to succeed.

802.1X AUTHENTICATION CONFIGURATION

The configuration can be done either using the webGUI or via phone keypad.

In this guide, we will GXV32xx series as example to configure the 802.1x authentication.

Via Web User Interface

To enable and configure 802.1x authentication using the web user interface on GXV32xx, please refer to following steps:

1. Access to web GUI of your device.
2. Navigate to **Maintenance** -> **Network Settings** -> **802.1x Mode**.
3. Choose from drop down list the 802.1x method desired (EAP_MD5, EAP_TLS or EAP-PEAP)

- If **EAP-MD5** is selected:

  a. Enter the user name in **802.1x Identity** field for authentication.
  b. Enter the password in **802.1x Secret** field for authentication.

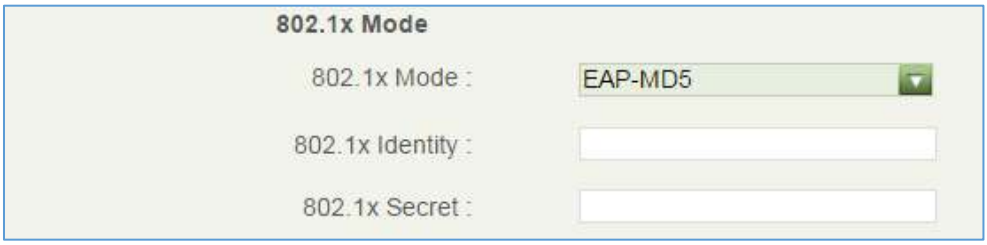


Figure 3: 802.1x MD5

- If **EAP-TLS** is selected:

  a. Enter the user name in **802.1x Identity** field for authentication.
  b. Enter the password in **802.1x Private Key Password** field for authentication.
  c. Click **Upload** button to browse and load **802.1x CA Certificate** (*.pem, *.cer, *.crt or *.der) from your local system.
  d. Click **Upload** button to browse and load the **802.1x Client Certificate** (*.pem, or *.cer) from your local system.
  e. Click **Upload** button to browse and load the **802.1x Client Certificate** (*.pem, or *.cer) from your local system.

Figure 4: 802.1x TLS

- If **EAP-PEAP** is selected:

  a. Enter the user name in 802.1x Identity field for authentication.
  b. Enter the password in **802.1x Secret** field for authentication.
  c. Click **Upload** button to browse and load **802.1x CA Certificate** (*.pem, *.cer, *.crt or *.der) from your local system.

Figure 5: 802.1x PEAP

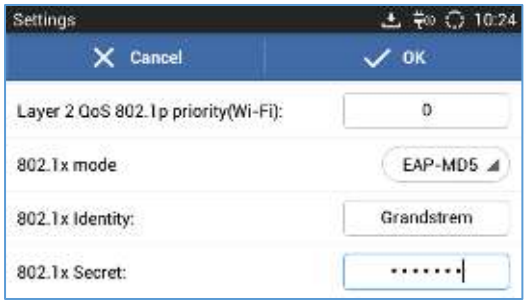
4. Press **Save** and **Apply** buttons and reboot your device to apply the new settings.

Via Phone Keypad

To enable and configure the 802.1x authentication using the keypad menu on GXV32xx, please refer to following steps:

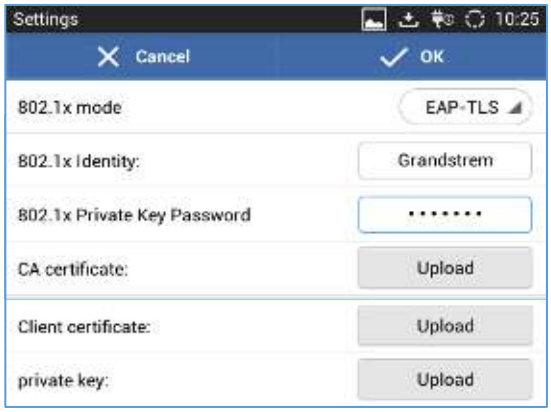
1. Press **Menu** button and navigate to **Settings**.
2. Access to **Wireless & Network** settings and navigate to **Additional network settings**
3. Access to **802.1x mode** and select the method desired from the drop down list.

- If **EAP-MD5** is selected:



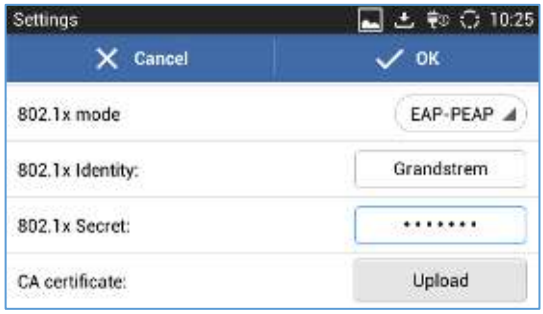
a. Enter the user name in **Identity** field for authentication.
b. Enter the password in **MD5 Password** field for authentication.

- If **EAP-TLS** is selected:



a. Enter the user name in **802.1x Identity** field for authentication.
b. Enter the password in **802.1x Private Key Password** field for authentication.
c. Click **Upload** button to browse and load **802.1x CA Certificate** (*.pem, *.cer, *.crt or *.der) from your local system.
d. Click **Upload** button to browse and load the **802.1x Client Certificate** (*.pem, or *.cer) from your local system.
e. Click **Upload** button to browse and load the **802.1x Client Certificate** (*.pem, or *.cer) from your local system.

- If **EAP-PEAP** is selected:



a. Enter the user name in **802.1x Identity** field for authentication.
b. Enter the password in **802.1x Secret** field for authentication.
c. Click **Upload** button to browse and load **802.1x CA Certificate** (*.pem, *.cer, *.crt or *.der) from your local system

4. Press **OK** for saving and reboot your device to apply the new settings.

802.1X AUTHENTICATION FLOW

Once 802.1x settings are configured on the phone either via webGUI or via phone Keypad and device rebooted. Authentication process is divided into two stages:

Prior to Authentication

The only messages that will be accepted from the client are the EAP messages, which will be forwarded to the authentication Server. The authenticator will block access to the network for the phone, it will try to establish a security negotiation with the IP phone and create an 802.1X session. The IP phone provides its authentication information for the authenticator, then the authenticator forwards the information to the authentication server.

Authentication Process

After 802.1x client is powered on, it will transmit the EAP message to the authenticator. It will forward then the client's request to the authentication server without changing its contents. The server will verify the user credentials and transmit back its response to the authenticator, which will determine whether the port remains in blocked mode or will grant access to the client, if the server response is "Access granted" then client port state will have access to the network. If the authentication fails, the authenticator port will remain blocked, and in some cases the port will be disabled (depends on vendor implementation).

Authentication Process Using EAP-MD5 Protocol

Authentication Process

The following figure shows a successful 802.1x authentication process using EAP-MD5 protocol (RADIUS is used as authentication server).

Figure 6: 802.1x Authentication Using MD5

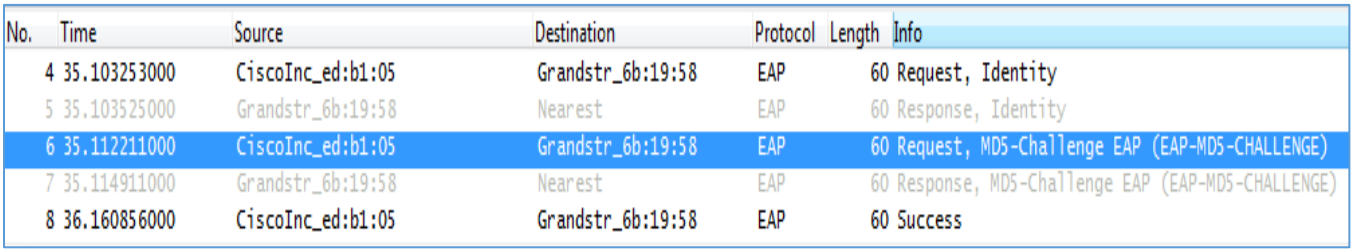
1. The client starts by sending an “EAPOL-Start” packet to the switch.
2. The switch reply to the client with an “EAP-Request/Identity” packet.
3. The client sends back an “EAP-Response/Identity” packet to the switch.
4. The switch strips the Ethernet header and encapsulates the remaining EAP frame in the RADIUS format, and then sends it to the RADIUS server.
5. The RADIUS server recognizes the packet as an EAP-MD5 type and sends back a Challenge message to switch.
6. The switch strips the authentication server’s frame header, encapsulates the remaining EAP frame into the EAPOL format, and sends it to the client.
7. The client responds to the Challenge message.
8. The switch passes the response to the RADIUS server.
9. The RADIUS server validates the authentication information and sends an authentication success message.
10. The switch passes the successful message to the client.

Once the phone is authenticated successfully, the switch provides network access permissions. If the phone does not provide proper identification, RADIUS server will reply with a rejection message. The switch relies

the message to the phone and blocks access to the LAN. When the phone is disabled or reset, it will send an EAPOL-Logoff message, which prompts the switch to block access to the LAN Success message.

Flow Example

The following figure shows a trace of EAP-MD5 process.



No.	Time	Source	Destination	Protocol	Length	Info
4	35.103253000	CiscoInc_ed:b1:05	Grandstr_6b:19:58	EAP	60	Request, Identity
5	35.103525000	Grandstr_6b:19:58	Nearest	EAP	60	Response, Identity
6	35.112211000	CiscoInc_ed:b1:05	Grandstr_6b:19:58	EAP	60	Request, MD5-Challenge EAP (EAP-MD5-CHALLENGE)
7	35.114911000	Grandstr_6b:19:58	Nearest	EAP	60	Response, MD5-Challenge EAP (EAP-MD5-CHALLENGE)
8	36.160856000	CiscoInc_ed:b1:05	Grandstr_6b:19:58	EAP	60	Success

Figure 7: EAP MD5 Challenge

Authentication Process Using EAP-TLS Protocol

Authentication Process

The following figure shows a successful 802.1x authentication process using EAP-MD5 protocol (RADIUS is used as authentication server).

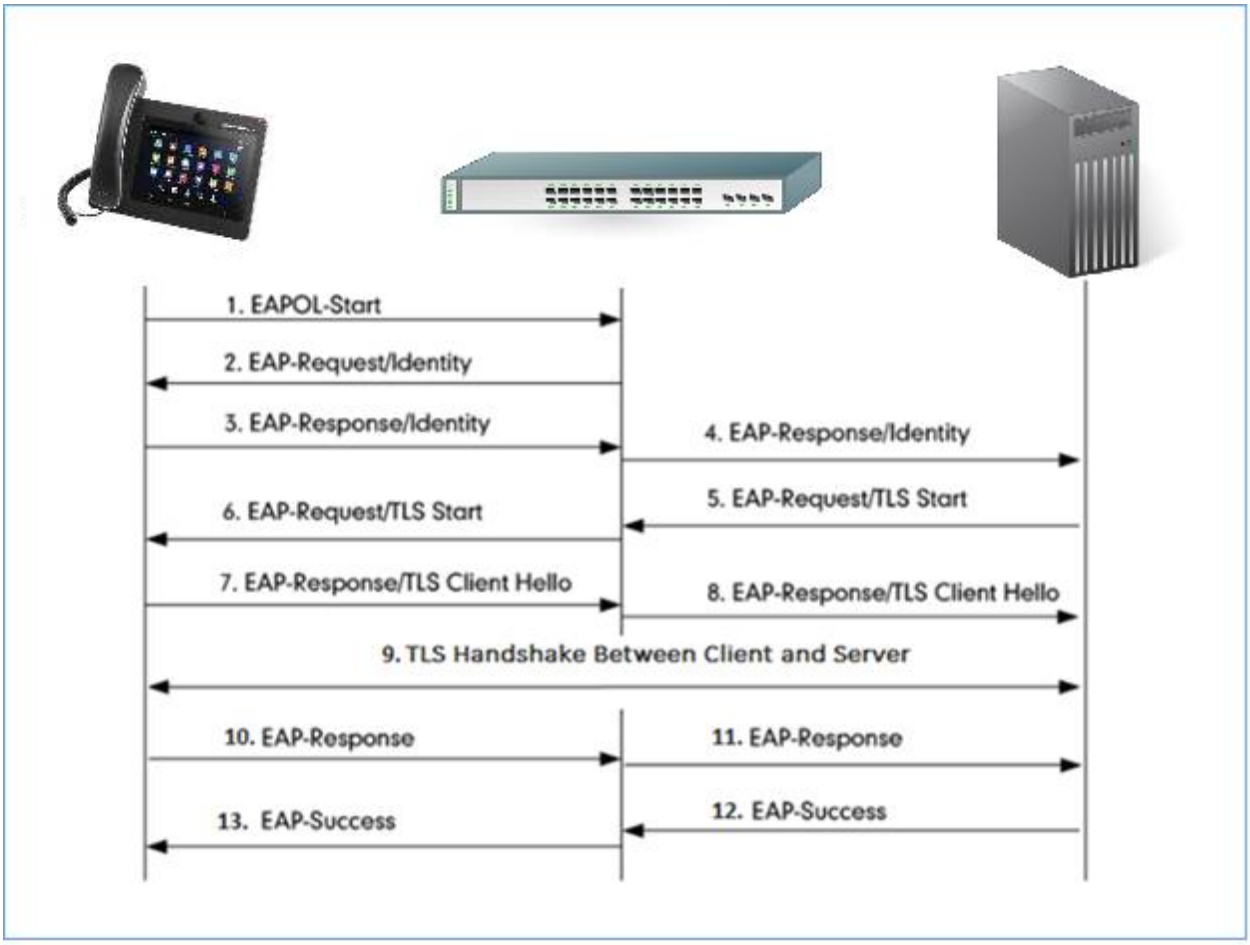


Figure 8: 802.1x Authentication Using TLS

1. The phone sends an “EAPO-Start” packet to the switch.
2. The switch reply with an “EAP-Request/Identity” packet.
3. The phone sends back an "EAP-Response/Identity" packet to the switch.
4. The switch strips the Ethernet header and encapsulates the remaining EAP frame in the RADIUS format, and then sends it to the server.
5. The authentication server recognizes the packet as an EAP-TLS type and sends an “EAP-Request” packet with a TLS start message to the switch.
6. The switch strips the authentication server’s frame header, encapsulates the remaining EAP frame in the EAPOL format, and then sends it to the phone.
7. The phone responds with an “EAP-Respond” packet containing a TLS client hello handshake message to the switch. The client hello message includes the TLS version supported by the phone, a session ID, a random number and a set of cipher suites.
8. The switch passes the response to the authentication server.
9. TLS handshake between the phone and RADIUS server (phone and server exchange key and cipher).
10. The phone responds with an “EAP-Response” packet to the switch.

11. The switch passes the response to the server.
12. The server responds with a success message indicating that the phone and the RADIUS server have successfully authenticated each other.
13. The switch passes the message to the phone.

After the phone's successful authentication, the switch provides network access permissions. If the phone does not provide proper identification, the RADIUS server responds with a rejection message. The switch relies the message to the phone and blocks access to the LAN. When the phone is disabled or reset, it will send an EAPOL-Logoff message, which tell the switch to block access to the LAN.

Flow Example

Bellow trace example of EAP-TLS authentication.

No.	Time	Source	Destination	Protocol	Length	Info
9	35.748566000	CiscoInc_ed:b1:06	Grandstr_6b:19:58	EAP	60	Request, Identity
10	35.748876000	Grandstr_6b:19:58	Nearest	EAP	60	Response, Identity
11	35.755039000	CiscoInc_ed:b1:06	Grandstr_6b:19:58	EAP	60	Request, TLS EAP (EAP-TLS)
12	35.756639000	Grandstr_6b:19:58	Nearest	TLSv1	222	Client Hello
13	35.795328000	CiscoInc_ed:b1:06	Grandstr_6b:19:58	TLSv1	1042	Server Hello, Certificate, Server Key Exchange, Certificate Request
14	35.798797000	Grandstr_6b:19:58	Nearest	EAP	60	Response, TLS EAP (EAP-TLS)
15	35.812699000	CiscoInc_ed:b1:06	Grandstr_6b:19:58	TLSv1	1042	Server Hello, Certificate, Server Key Exchange, Certificate Request
16	35.813164000	Grandstr_6b:19:58	Nearest	EAP	60	Response, TLS EAP (EAP-TLS)
17	35.821524000	CiscoInc_ed:b1:06	Grandstr_6b:19:58	TLSv1	743	Server Hello, Certificate, Server Key Exchange, Certificate Request
18	35.966417000	Grandstr_6b:19:58	Nearest	TLSv1	1426	Certificate, Client Key Exchange, Certificate Verify, Change Cipher
19	35.980125000	CiscoInc_ed:b1:06	Grandstr_6b:19:58	EAP	60	Request, TLS EAP (EAP-TLS)
20	35.980665000	Grandstr_6b:19:58	Nearest	TLSv1	1104	Certificate, Client Key Exchange, Certificate Verify, Change Cipher
21	36.001255000	CiscoInc_ed:b1:06	Grandstr_6b:19:58	TLSv1	87	Change Cipher Spec, Encrypted Handshake Message
22	36.003449000	Grandstr_6b:19:58	Nearest	EAP	60	Response, TLS EAP (EAP-TLS)
23	37.049720000	CiscoInc_ed:b1:06	Grandstr_6b:19:58	EAP	60	Success

Figure 9: EAP TLS Challenge

Authentication Process Using EAP-PEAP (MSCHAPv2) Protocol

Authentication Process

The following figure shows a successful 802.1x authentication process using EAP-PEAPv0/MSCHAPv2 protocol (RADIUS is used as authentication server).

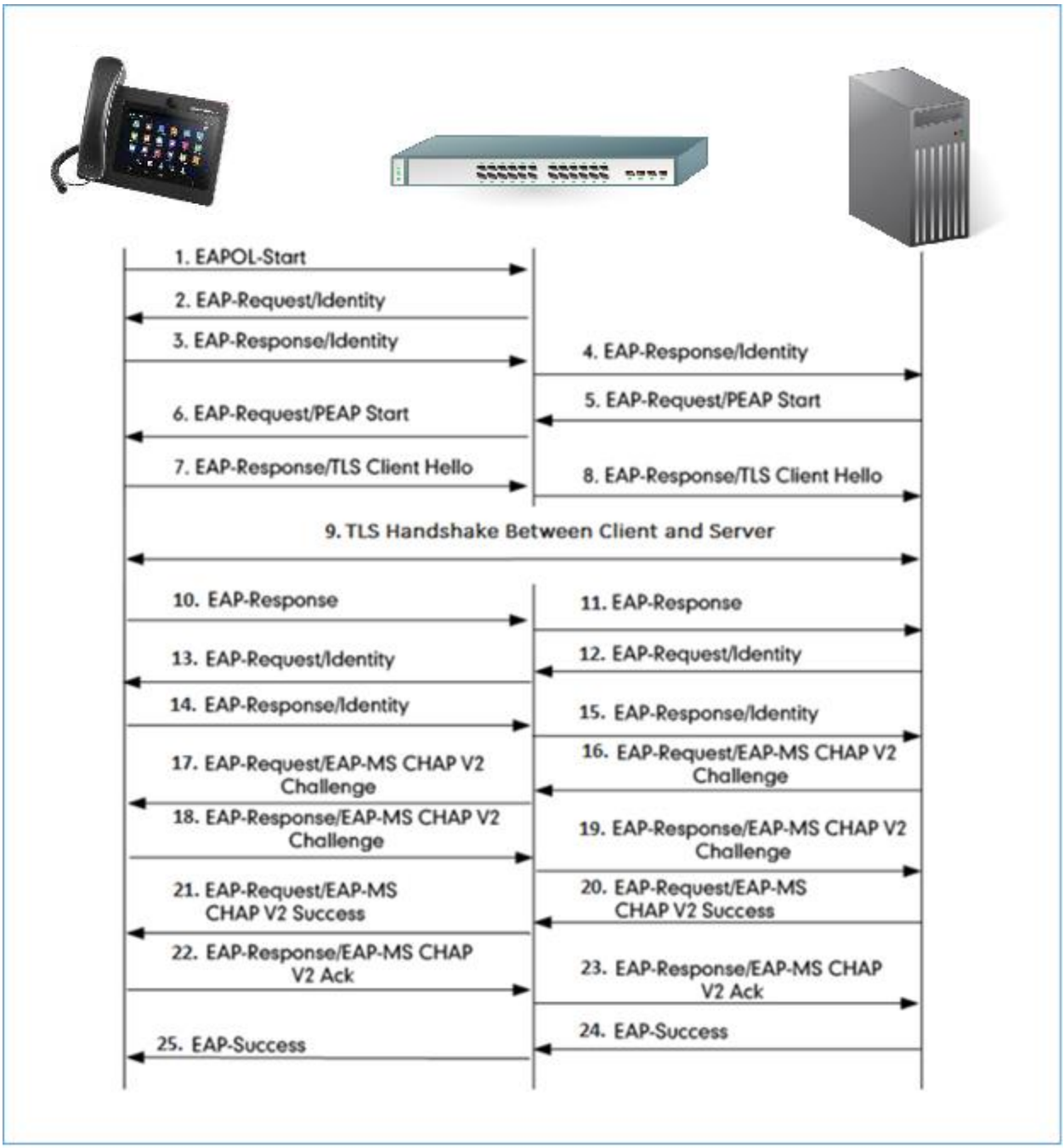


Figure 10: 802.1x Authentication Using PEAPv0/MSCHAPv2

1. The phone sends an “EAPO-Start” packet to the switch.
2. The switch reply with an “EAP-Request/Identity” packet.
3. The phone sends back an "EAP-Response/Identity" packet to the switch.
4. The switch strips the Ethernet header and encapsulates the remaining EAP frame in the RADIUS format, and then sends it to the server.
5. The authentication server recognizes the packet as an EAP-TLS type and sends an “EAP-Request” packet with a TLS start message to the switch.
6. The switch strips the authentication server’s frame header, encapsulates the remaining EAP frame in the EAPOL format, and then sends it to the phone.
7. The phone responds with an “EAP-Respond” packet containing a TLS client hello handshake message to the switch. The client hello message includes the TLS version supported by the phone, a session ID, a random number and a set of cipher suites.
8. The switch passes the response to the authentication server.
9. TLS handshake between the phone and RADIUS server (phone and server exchange key and cipher).
10. The phone responds with an “EAP-Response” packet.
11. The switch passes the response to the server.
12. The RADIUS server sends an “EAP-Request/Identity” packet.
13. The switch relies the request to the phone.
14. The phone replies with an “EAP-Response/Identity” packet.
15. The switch passes the response to the RADIUS server.
16. The RADIUS server sends an “EAP-Request” packet that includes a MSCHAPv2 challenge message.
17. The switch passes the request to the phone.
18. The phone sends back a challenge message to the switch.
19. The switch relies the message to the server.
20. The RADIUS server sends a success message indicating that the phone provided the proper identity.
21. The switch relies the message to the phone.
22. The phone responds with an ACK message to the switch.
23. The switch relies the respond message to the server.
24. The RADIUS server sends a successful message to the switch.
25. The switch passes the message to the phone.

After the phone’s successful authentication, the switch provides network access permissions. If the phone does not provide proper identification, the RADIUS server responds with a rejection message. The switch relies the message to the phone and blocks access to the LAN. When the phone is disabled or reset, it will send an EAPOL-Logoff message, which tell the switch to block access to the LAN.

Flow Example

Bellow trace example of EAP-PEAPv0/MSCHAPv2 authentication.

Please note that the trace contains two phases, the first one is similar to the EAP TLS challenge, after the "EAP-Response" in step 10, the server sends another "EAP-Request/Identity" protected by the TLS ciphersuite negotiated in phase 1 to exchange the phone user and password.

No.	Time	Source	Destination	Protocol	Length	Info
12	35.665216000	CiscoInc_ed:b1:04	Grandstr_6b:19:58	EAP	60	Request, Identity
13	35.665575000	Grandstr_6b:19:58	Nearest	EAP	60	Response, Identity
14	35.671597000	CiscoInc_ed:b1:04	Grandstr_6b:19:58	EAP	60	Request, Protected EAP (EAP-PEAP)
15	35.672614000	Grandstr_6b:19:58	Nearest	TLSv1	226	Client Hello
16	35.706501000	CiscoInc_ed:b1:04	Grandstr_6b:19:58	TLSv1	1042	Server Hello, Certificate, Server Key Exchange, Server Hello Done
17	35.708283000	Grandstr_6b:19:58	Nearest	EAP	60	Response, Protected EAP (EAP-PEAP)
18	35.713686000	CiscoInc_ed:b1:04	Grandstr_6b:19:58	TLSv1	1038	Server Hello, Certificate, Server Key Exchange, Server Hello Done
19	35.714209000	Grandstr_6b:19:58	Nearest	EAP	60	Response, Protected EAP (EAP-PEAP)
20	35.721014000	CiscoInc_ed:b1:04	Grandstr_6b:19:58	TLSv1	603	Server Hello, Certificate, Server Key Exchange, Server Hello Done
21	35.804305000	Grandstr_6b:19:58	Nearest	TLSv1	226	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Messag
22	35.818551000	CiscoInc_ed:b1:04	Grandstr_6b:19:58	TLSv1	83	Change Cipher Spec, Encrypted Handshake Message
23	35.820230000	Grandstr_6b:19:58	Nearest	EAP	60	Response, Protected EAP (EAP-PEAP)
24	35.824942000	CiscoInc_ed:b1:04	Grandstr_6b:19:58	TLSv1	61	Application Data
25	35.827767000	Grandstr_6b:19:58	Nearest	TLSv1	98	Application Data, Application Data
26	35.833959000	CiscoInc_ed:b1:04	Grandstr_6b:19:58	TLSv1	77	Application Data
27	35.840927000	Grandstr_6b:19:58	Nearest	TLSv1	146	Application Data, Application Data
28	35.854437000	CiscoInc_ed:b1:04	Grandstr_6b:19:58	TLSv1	109	Application Data
29	35.857707000	Grandstr_6b:19:58	Nearest	TLSv1	98	Application Data, Application Data
30	35.871943000	CiscoInc_ed:b1:04	Grandstr_6b:19:58	TLSv1	61	Application Data
31	35.872632000	Grandstr_6b:19:58	Nearest	TLSv1	98	Application Data, Application Data
32	36.915426000	CiscoInc_ed:b1:04	Grandstr_6b:19:58	EAP	60	Success

Figure 11: EAP PEAP Challenge