



Grandstream Networks, Inc.

GXV33XX IP Video Phones with Android™

802.1x Authentication Guide



Table of Contents

SUPPORTED DEVICES	4
INTRODUCTION.....	5
802.1X AUTHENTICATION PROCESS.....	6
802.1x Elements.....	6
Authentication Process	6
Authentication Flowchart.....	7
EAP Methods for 802.1x Authentication	8
802.1X AUTHENTICATION CONFIGURATION.....	9
Via Web User Interface.....	9
Via Phone Keypad.....	10
802.1X AUTHENTICATION FLOW.....	13
Authentication Process Using EAP-MD5 Protocol	13
<i>Authentication Process</i>	13
<i>Flow Example</i>	14
Authentication Process Using EAP-TLS Protocol	15
<i>Authentication Process</i>	15
<i>Flow Example</i>	16
Authentication Process Using EAP-PEAP Protocol	17
<i>Authentication Process</i>	17
<i>Flow Example</i>	19



Table of Figures

Figure 1: 802.1x Authentication Process	6
Figure 2: 802.1x Authentication Flowchart	7
Figure 3: 802.1x MD5 Mode.....	9
Figure 4: 802.1x TLS	9
Figure 5: 802.1x PEAP	10
Figure 6: 802.1X Keypad Menu Configuration.....	10
Figure 7: 802.1x mode (EAP-MD5).....	11
Figure 8: 802.1x mode (EAP-TLS).....	11
Figure 9: 802.1x mode (PEAP)	11
Figure 10: Wi-Fi 802.1X LCD Menu settings	12
Figure 11: Certificate Management	12
Figure 12: 802.1x Authentication Using MD5	13
Figure 13: EAP MD5 Challenge	14
Figure 14: 802.1x Authentication Using TLS	15
Figure 15: EAP TLS Challenge	16
Figure 16: 802.1x Authentication Using PEAPv0/MSCHAPv2	17
Figure 17: EAP PEAP Challenge	19

SUPPORTED DEVICES

Following table shows Grandstream products supporting 802.1x feature:

Model	Supported	Firmware
<i>IP Multimedia Phone for Android™ GXV33XX Series</i>		
GXV3370	Yes	1.0.3.9 or higher
GXV3380		
GXV3350		

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 Video 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:

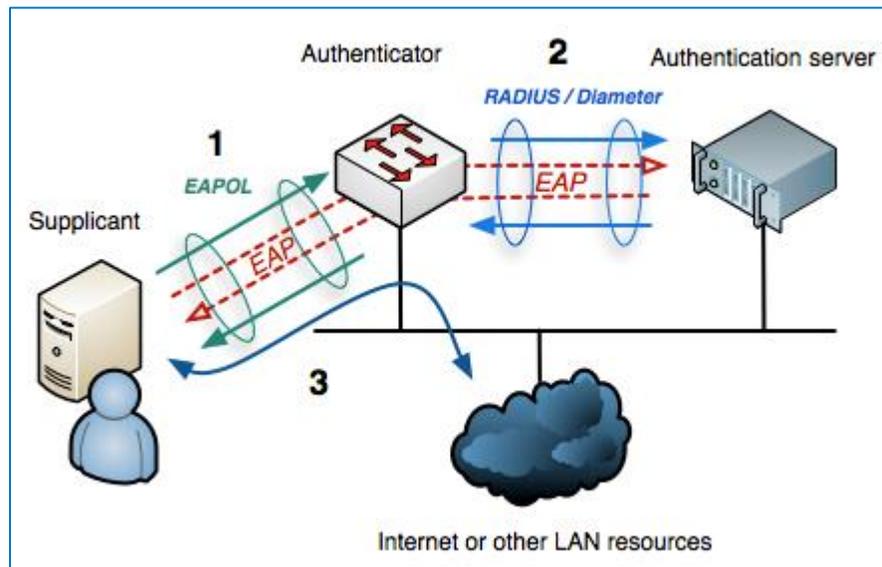


Figure 1: 802.1x Authentication Process

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 an EAP-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:

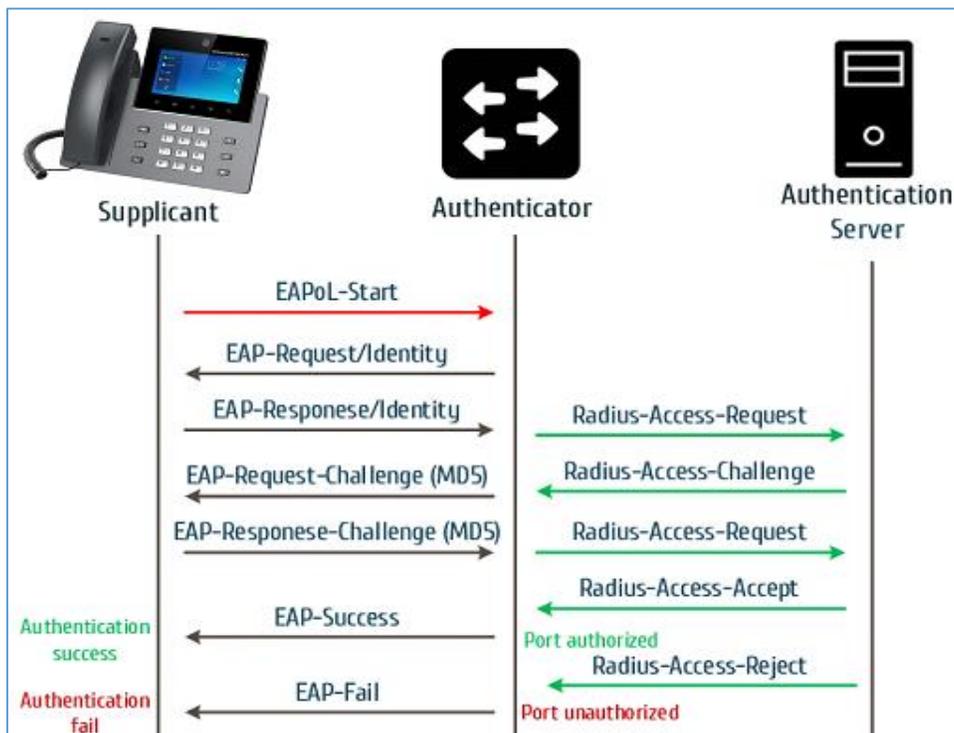


Figure 2: 802.1x Authentication Flowchart



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 Web GUI or via phone keypad:

Via Web User Interface

To enable and configure 802.1x authentication using the web user interface on GXV33xx, please refer to following steps:

1. Access to Web GUI of your device.
2. Navigate to **GXV33XX GUI** → **Network Settings** → **Ethernet 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:

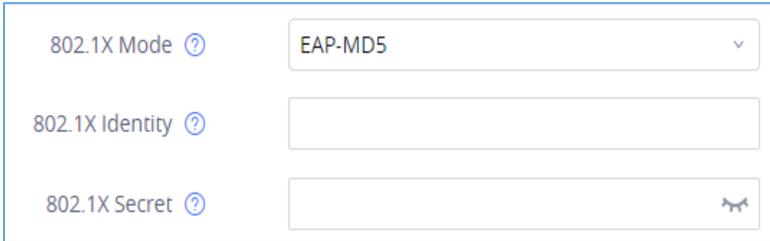


Figure 3: 802.1x MD5 Mode

- a. Enter the username in **802.1x Identity** field for authentication.
- b. Enter the password in **802.1x Secret** field for authentication.

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

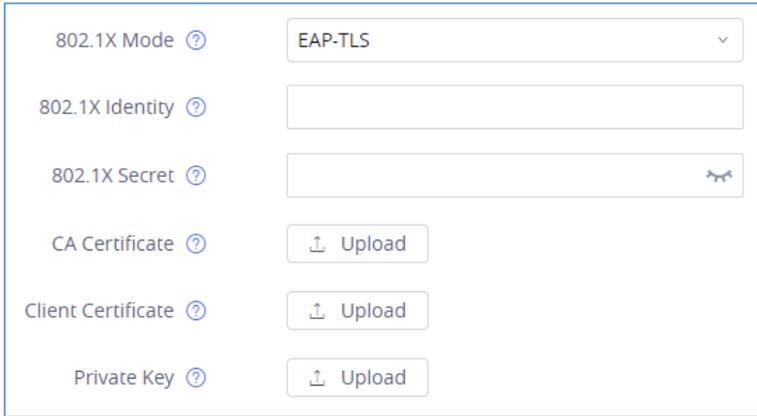
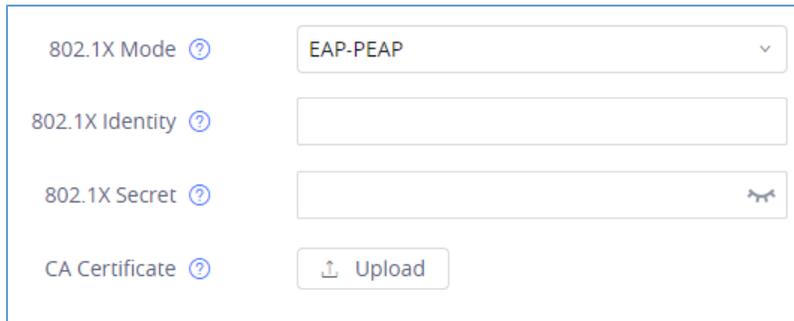


Figure 4: 802.1x TLS

- a. Enter the username 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**, **Client Certificate** and **Private Key**.



- If EAP-PEAP is selected:



The screenshot shows a configuration interface for 802.1X PEAP. It includes four fields: '802.1X Mode' with a dropdown menu set to 'EAP-PEAP', '802.1X Identity' with an empty text input field, '802.1X Secret' with an empty password input field and a toggle icon, and 'CA Certificate' with an 'Upload' button.

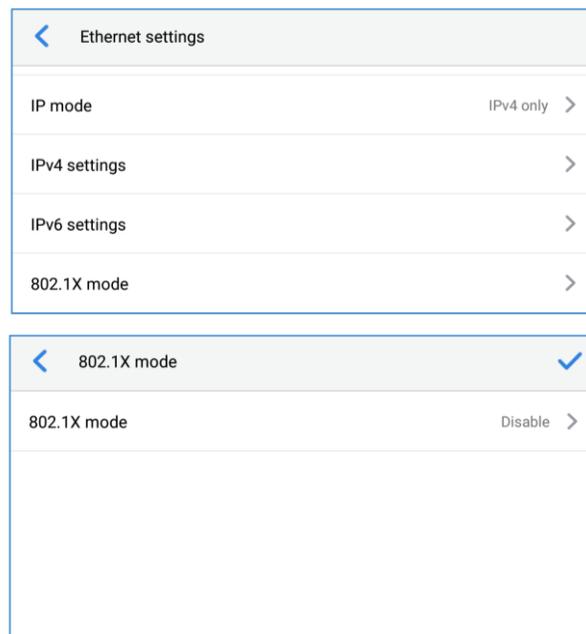
Figure 5: 802.1x PEAP

- Enter the username in 802.1x Identity field for authentication.
 - Enter the password in **802.1x Secret** field for authentication.
 - Click **Upload** button to browse and load **802.1x CA Certificate** (*.pem, *.cer, *.crt or *.der) from your local system.
- 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 GXV33xx, please refer to following steps:

- Press **Menu** button and navigate to **Settings**.
- Access to **Network** → **Ethernet settings** and navigate to **Additional network settings**
- Select **802.1X mode**



The screenshot shows two screens from a keypad menu. The top screen is titled 'Ethernet settings' and lists 'IP mode' (IPv4 only), 'IPv4 settings', 'IPv6 settings', and '802.1X mode'. The bottom screen is titled '802.1X mode' and shows '802.1X mode' set to 'Disable' with a checkmark in the top right corner.

Figure 6: 802.1X Keypad Menu Configuration



- If EAP-MD5 is selected:



Figure 7: 802.1x mode (EAP-MD5)

- Enter the username in **Identity** field for authentication.
- Enter the password in **MD5 Password** field for authentication.

- If EAP-TLS is selected:



Figure 8: 802.1x mode (EAP-TLS)

- Enter the username in **802.1x Identity** field for authentication.
- Enter the **password** in **802.1x Private Key Password** field for authentication.
- Both, the **802.1x CA Certificate** and **802.1x Client Certificate** (*.pem, or *.cer) need to be provisioned or uploaded from the Web GUI.

- If EAP-PEAP is selected:



Figure 9: 802.1x mode (PEAP)

- Enter the username in **802.1x Identity** field for authentication.
- Enter the password in **802.1x Secret** field for authentication.
- Upload** the **802.1x CA Certificate** (*.pem, *.cer, *.crt or *.der) from the Web GUI

Notes:

- The GXV33XX supports 802.1X EAP authentication for both Ethernet and Wi-Fi connection. To access 802.1x Wi-Fi settings, follow the steps below:
 - go to **GXV Menu → Network → Wi-Fi → Add Network**
 - Enter the SSID name, then select “802.1X EAP” under security parameter



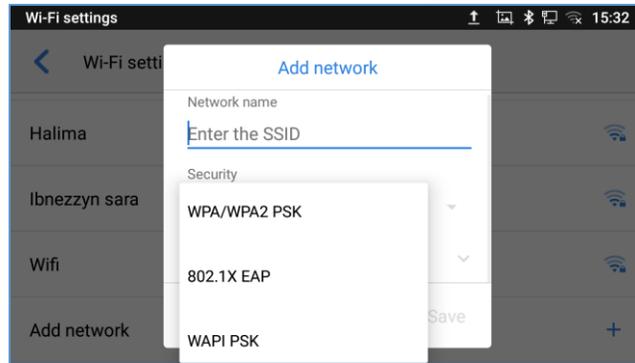


Figure 10: Wi-Fi 802.1X LCD Menu settings

- Available EAP methods: PWD (for EAP-MD5); TLS; TTLS and PEAP
- The system will use CA/Client certificates and Key uploaded under **Web GUI → System Settings → Security Settings → Certificate Management**

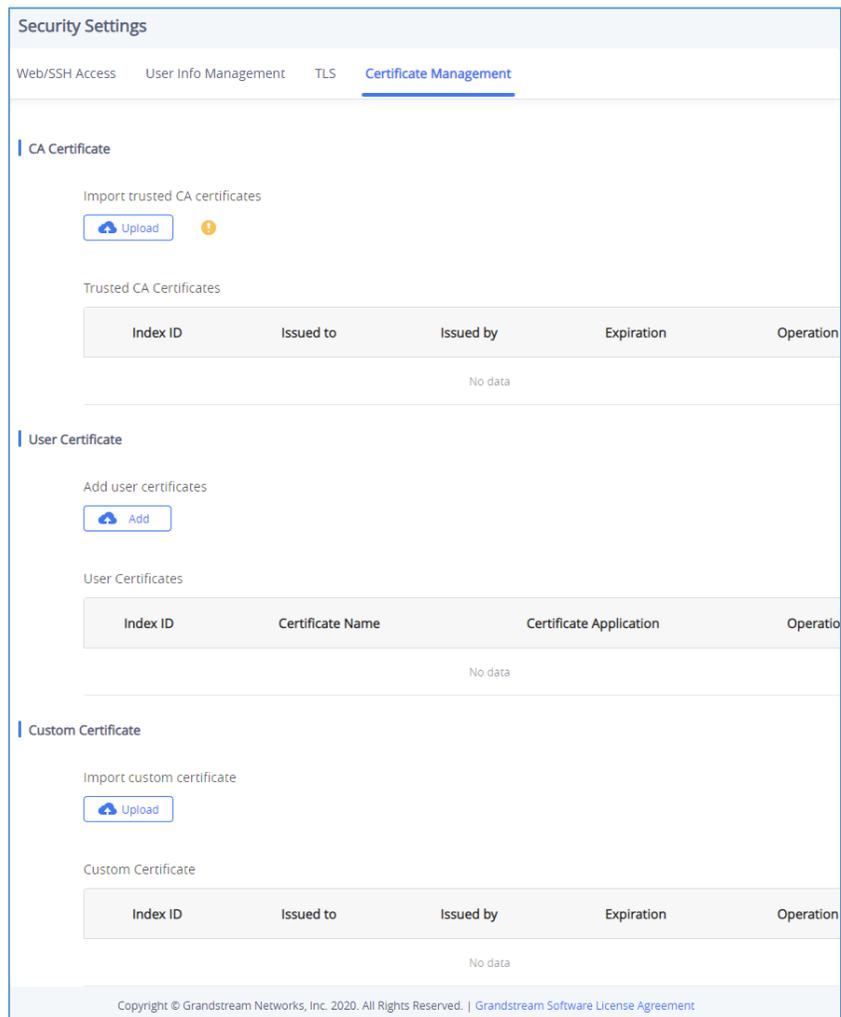


Figure 11: Certificate Management



802.1X AUTHENTICATION FLOW

Once 802.1x settings are configured on the phone either via Web GUI or via phone Keypad and after the completes the reboot. The Authentication process begins, it 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).

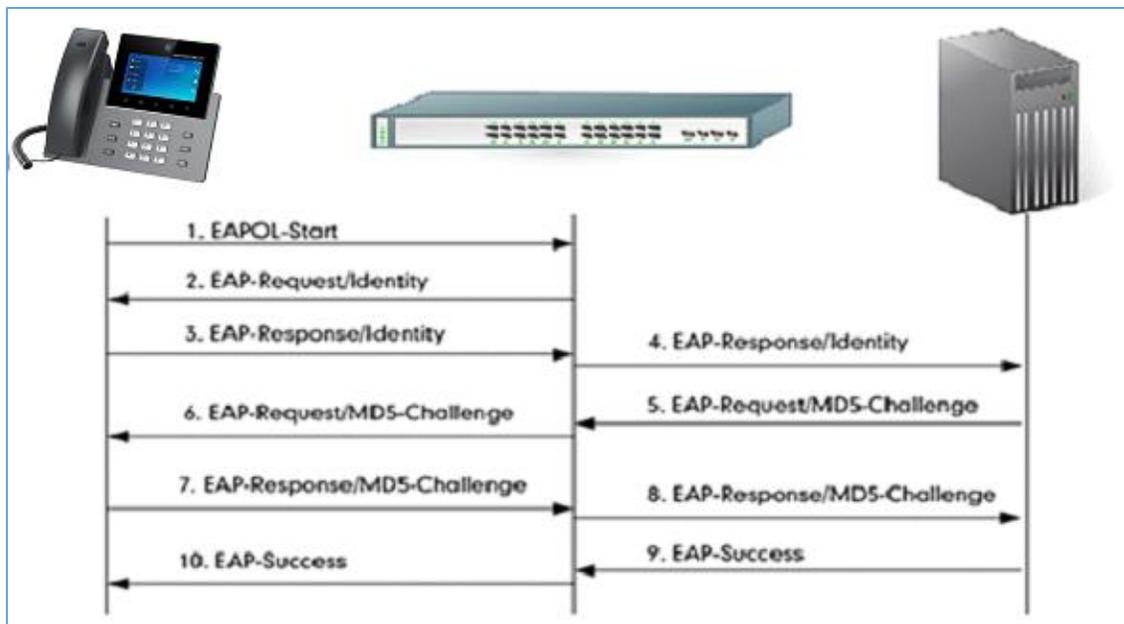


Figure 12: 802.1x Authentication Using MD5



1. The client starts by sending an “EAPOL-Start” packet to the switch.
2. The switch replies 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 relays 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.

Source	Destination	Protocol	Length	Info
CiscoInc_ed:b1:05	Grandstr_6b:19:58	EAP	60	Request, Identity
Grandstr_6b:19:58	Nearest	EAP	60	Response, Identity
CiscoInc_ed:b1:05	Grandstr_6b:19:58	EAP	60	Request, MD5-Challenge EAP (EAP-MD5-CHALLENGE)
Grandstr_6b:19:58	Nearest	EAP	60	Response, MD5-Challenge EAP (EAP-MD5-CHALLENGE)
CiscoInc_ed:b1:05	Grandstr_6b:19:58	EAP	60	Success

Figure 13: 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).

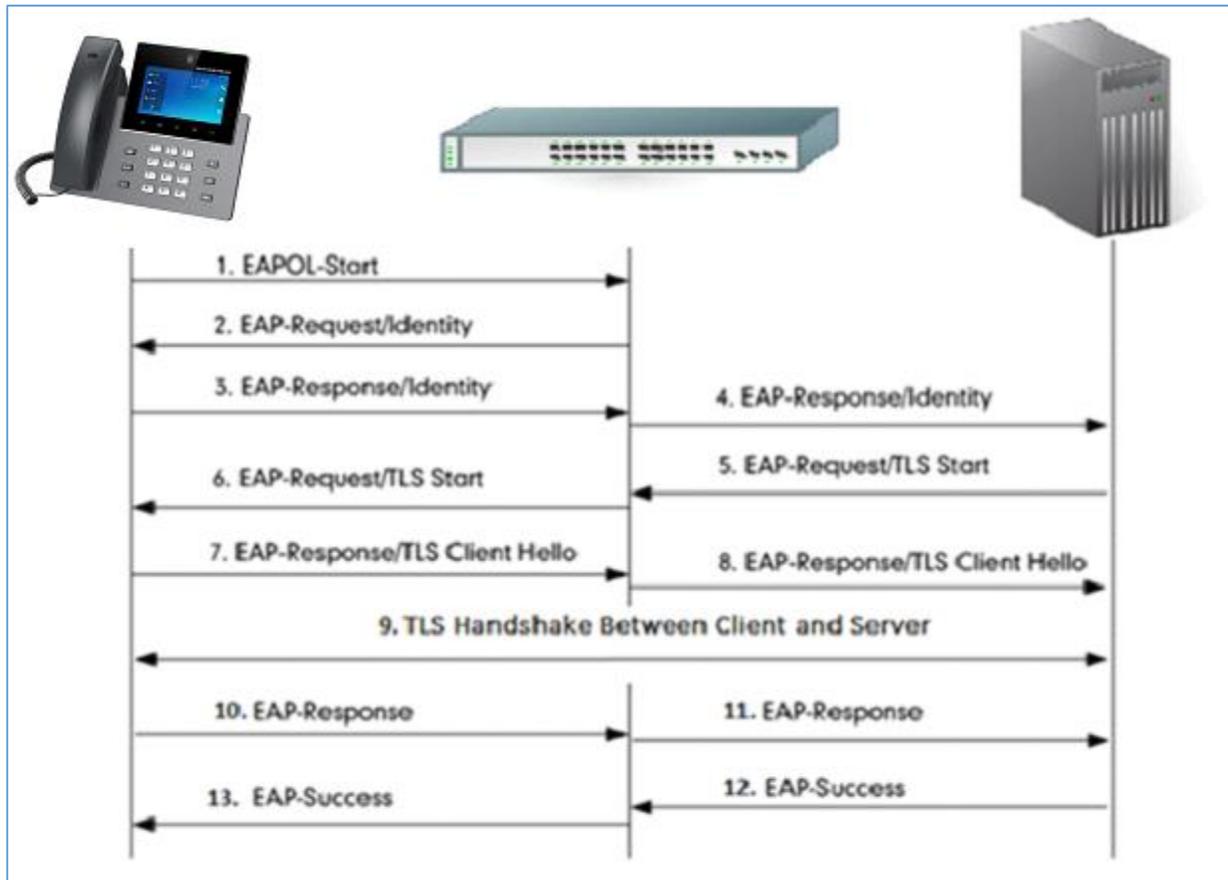


Figure 14: 802.1x Authentication Using TLS

1. The phone sends an "EAPOL-Start" packet to the switch.
2. The switch replies 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-Response" 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 relays 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 tells the switch to block access to the LAN.

Flow Example

Bellow trace example of EAP-TLS authentication.

Source	Destination	Protocol	Length	Info
CiscoInc_ed:b1:06	Grandstr_6b:19:58	EAP	60	Request, Identity
Grandstr_6b:19:58	Nearest	EAP	60	Response, Identity
CiscoInc_ed:b1:06	Grandstr_6b:19:58	EAP	60	Request, TLS EAP (EAP-TLS)
Grandstr_6b:19:58	Nearest	TLSv1	222	Client Hello
CiscoInc_ed:b1:06	Grandstr_6b:19:58	TLSv1	1042	Server Hello, Certificate, Server Key Exchange, Certificate Request
Grandstr_6b:19:58	Nearest	EAP	60	Response, TLS EAP (EAP-TLS)
CiscoInc_ed:b1:06	Grandstr_6b:19:58	TLSv1	1042	Server Hello, Certificate, Server Key Exchange, Certificate Request
Grandstr_6b:19:58	Nearest	EAP	60	Response, TLS EAP (EAP-TLS)
CiscoInc_ed:b1:06	Grandstr_6b:19:58	TLSv1	743	Server Hello, Certificate, Server Key Exchange, Certificate Request
Grandstr_6b:19:58	Nearest	TLSv1	1426	Certificate, Client Key Exchange, Certificate Verify, Change Cipher
CiscoInc_ed:b1:06	Grandstr_6b:19:58	EAP	60	Request, TLS EAP (EAP-TLS)
Grandstr_6b:19:58	Nearest	TLSv1	1104	Certificate, Client Key Exchange, Certificate Verify, Change Cipher
CiscoInc_ed:b1:06	Grandstr_6b:19:58	TLSv1	87	Change Cipher Spec, Encrypted Handshake Message
Grandstr_6b:19:58	Nearest	EAP	60	Response, TLS EAP (EAP-TLS)
CiscoInc_ed:b1:06	Grandstr_6b:19:58	EAP	60	Success

Figure 15: EAP TLS Challenge



Authentication Process Using EAP-PEAP Protocol

Authentication Process

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

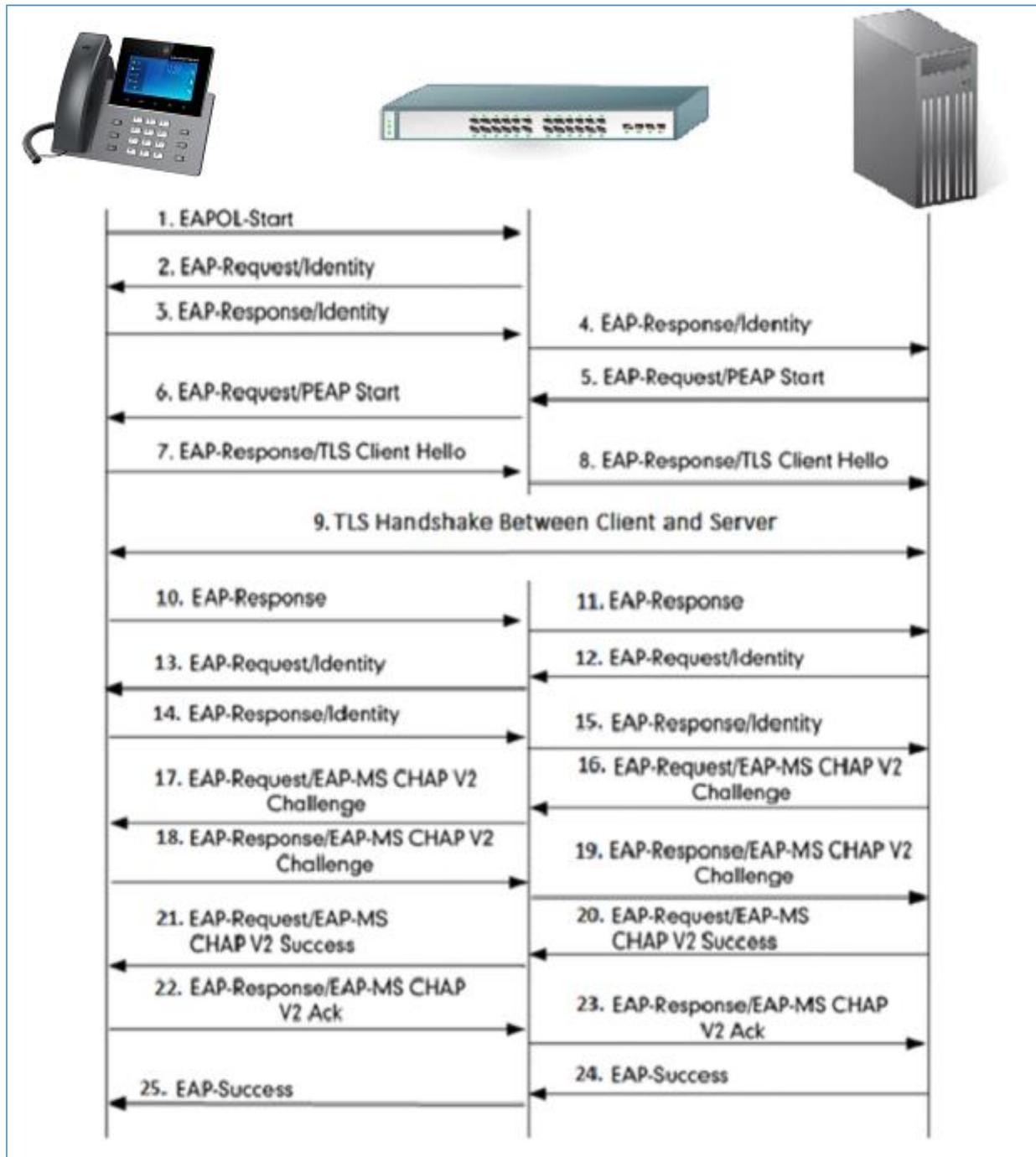


Figure 16: 802.1x Authentication Using PEAPv0/MSCHAPv2



1. The phone sends an “EAPo-Start” packet to the switch.
2. The switch replies 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 relays 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 relays the message to the server.
20. The RADIUS server sends a success message indicating that the phone provided the proper identity.
21. The switch relays the message to the phone.
22. The phone responds with an ACK message to the switch.
23. The switch relays 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 relays 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 to block access to the LAN on the switch.

Flow Example

Below 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 cipher suite negotiated in phase 1 to exchange the phone user and password.

Source	Destination	Protocol	Length	Info
CiscoInc_ed:b1:04	Grandstr_6b:19:58	EAP	60	Request, Identity
Grandstr_6b:19:58	Nearest	EAP	60	Response, Identity
CiscoInc_ed:b1:04	Grandstr_6b:19:58	EAP	60	Request, Protected EAP (EAP-PEAP)
Grandstr_6b:19:58	Nearest	TLSv1	226	Client Hello
CiscoInc_ed:b1:04	Grandstr_6b:19:58	TLSv1	1042	Server Hello, Certificate, Server Key Exchange, Server Hello Done
Grandstr_6b:19:58	Nearest	EAP	60	Response, Protected EAP (EAP-PEAP)
CiscoInc_ed:b1:04	Grandstr_6b:19:58	TLSv1	1038	Server Hello, Certificate, Server Key Exchange, Server Hello Done
Grandstr_6b:19:58	Nearest	EAP	60	Response, Protected EAP (EAP-PEAP)
CiscoInc_ed:b1:04	Grandstr_6b:19:58	TLSv1	603	Server Hello, Certificate, Server Key Exchange, Server Hello Done
Grandstr_6b:19:58	Nearest	TLSv1	226	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
CiscoInc_ed:b1:04	Grandstr_6b:19:58	TLSv1	83	Change Cipher Spec, Encrypted Handshake Message
Grandstr_6b:19:58	Nearest	EAP	60	Response, Protected EAP (EAP-PEAP)
CiscoInc_ed:b1:04	Grandstr_6b:19:58	TLSv1	61	Application Data
Grandstr_6b:19:58	Nearest	TLSv1	98	Application Data, Application Data
CiscoInc_ed:b1:04	Grandstr_6b:19:58	TLSv1	77	Application Data
Grandstr_6b:19:58	Nearest	TLSv1	146	Application Data, Application Data
CiscoInc_ed:b1:04	Grandstr_6b:19:58	TLSv1	109	Application Data
Grandstr_6b:19:58	Nearest	TLSv1	98	Application Data, Application Data
CiscoInc_ed:b1:04	Grandstr_6b:19:58	TLSv1	61	Application Data
Grandstr_6b:19:58	Nearest	TLSv1	98	Application Data, Application Data
CiscoInc_ed:b1:04	Grandstr_6b:19:58	EAP	60	Success

Figure 17: EAP PEAP Challenge

