



Yealink Technical White Paper

802.1X Authentication



Version 15.20 | Jan.2019

Table of Contents

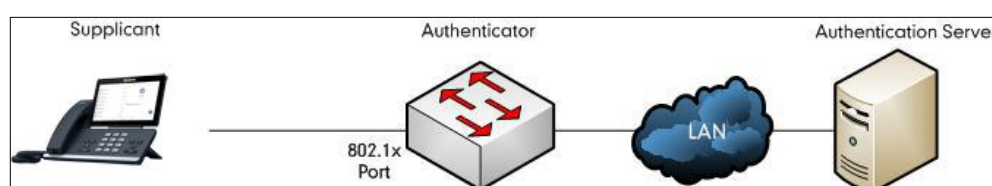
About 802.1X.....	3
Yealink Phones Compatible with 802.1X	3
Configuring 802.1X Settings.....	4
Configuring 802.1X using Configuration Files.....	4
Configuring 802.1X via Web User Interface	7
Configuring 802.1X via Phone User Interface.....	11
802.1X Authentication Process	11
Troubleshooting	14
Why doesn't the phone pass 802.1X authentication?.....	14
Appendix A: Glossary.....	15
Appendix B: 802.1X Authentication Process	16
A Successful Authentication Using EAP-MD5 Protocol.....	16
A Successful Authentication Using EAP-TLS Protocol.....	17
A Successful Authentication Using EAP-PEAP/MSCHAPv2 Protocol	19
A Successful Authentication Using EAP-TTLS/EAP-MSCHAPv2 Protocol.....	21
A Successful Authentication Using EAP-PEAP/GTC Protocol.....	21
A Successful Authentication Using EAP-TTLS/EAP-GTC Protocol	21
A Successful Authentication Using EAP-FAST Protocol.....	21

About 802.1X

The IEEE 802.1X standard defines a Port-based Network Access Control (PNAC) and authentication protocol that restricts unauthorized clients from connecting to a LAN. The IEEE 802.1X defines the encapsulation of the Extensible Authentication Protocol (EAP) defined in RFC3748 which is known as "EAP over LAN" or EAPOL.

802.1X authentication involves three parties: a supplicant, an authenticator and an authentication server. The supplicant is a client device (such as a Teams phone) that wishes to attach to the network. The authenticator is a network device, such as an Ethernet switch. And the authentication server is typically a host running software supporting the RADIUS and EAP protocols.

The authenticator acts like a security guard to a protected network. The supplicant is not allowed access through the authenticator to the protected side of the network until the supplicant's identity has been validated and authorized. An analogy to this is like providing a valid visa at the airport's arrival immigration before being allowed to enter the country. With 802.1X port-based authentication, the supplicant provides credentials, such as user name, password or digital certificate for the authenticator, and the authenticator forwards the credentials to the authentication server for verification. If the authentication server determines the credentials are valid, the supplicant is allowed to access resources located on the protected side of the network.



Yealink Phones Compatible with 802.1X

802.1X is the most widely accepted form of port-based network access control in use and is available on Yealink Teams IP phones.

Yealink Teams IP phones support 802.1X authentication based on EAP-MD5, EAP-TLS, EAP-PEAP/MSCHAPv2, EAP-TTLS/EAP-MSCHAPv2, EAP-PEAP/GTC, EAP-TTLS/EAP-GTC and EAP-FAST protocols.

Yealink Teams IP phones support 802.1X as a supplicant, both Pass-thru Mode and Pass-thru Mode with Proxy Logoff. When the device connected to the phone disconnects from the PC port, the Yealink Teams IP phone can provide additional security by sending an EAPOL Logoff message to the Ethernet switch. This functionality, also known as proxy logoff, prevents another device from using the port without first authenticating via 802.1X. The Pass-thru Mode is available on Yealink Teams IP phones running specified firmware version. You can ask your system administrator or contact Yealink Field Application Engineer (FAE) for more information.

Configuring 802.1X Settings

The 802.1X authentication on Yealink Teams IP phones is disabled by default. You can configure the 802.1X authentication in one of the following three ways:

- [Configuring 802.1X using Configuration Files](#)
- [Configuring 802.1X via Web User Interface](#)
- [Configuring 802.1X via Phone User Interface](#)

For detailed descriptions of the authentication parameters in configuration files, you can refer to [Configuring 802.1X using Configuration Files](#) on page 4. When setting up a large number of phones, Yealink recommends using configuration files. If you are provisioning a few phones, you can use the web user interface or phone user interface to configure 802.1X feature.

If the EAP-TLS, EAP-PEAP/MSCHAPv2, EAP-TTLS/EAP-MSCHAPv2, EAP-PEAP/GTC, EAP-TTLS/EAP-GTC or EAP-FAST protocol is preferred in your 802.1X environment, make sure that the firmware running on your new phone supports the protocol.




The followings provide system administrator with the procedures to successfully configure Yealink Teams IP phones in a secure 802.1X environment, and take configurations of a T58A Teams IP phone as examples.

Configuring 802.1X using Configuration Files

1. Add/Edit 802.1X authentication parameters in configuration files.

The following table shows the information of parameters:

Parameters	Permitted Values	Default
static.network.802_1x.mode	0, 1, 2, 3, 4, 5, 6 or 7	0
Description: Configures the 802.1x authentication method. 0 -Disabled 1 -EAP-MD5 2 -EAP-TLS 3 -EAP-PEAP/MSCHAPv2 4 -EAP-TTLS/EAP-MSCHAPv2 5 -EAP-PEAP/GTC 6 -EAP-TTLS/EAP-GTC 7 -EAP-FAST Note: If you change this parameter, the phone will reboot to make the change take effect.		

Parameters	Permitted Values	Default
Web User Interface: Network->Advanced->802.1x->802.1x Mode Phone User Interface: Tap  ->Settings->Device Settings->Network(default password: admin)->802.1x->802.1x Mode		
static.network.802_1x.identity	String within 32 characters	Blank
Description: Configures the identity (or user name) for 802.1x authentication. Example: static.network.802_1x.identity = admin Note: It works only if the value of the parameter "static.network.802_1x.mode" is set to 1, 2, 3, 4, 5, 6 or 7. If you change this parameter, the phone will reboot to make the change take effect. Web User Interface: Network->Advanced->802.1x->Identity Phone User Interface: Tap  ->Settings->Device Settings->Network(default password: admin)->802.1x->Identity		
static.network.802_1x.md5_password	String within 32 characters	Blank
Description: Configures the password for 802.1x authentication. Example: static.network.802_1x.md5_password = admin123 Note: It works only if the value of the parameter "static.network.802_1x.mode" is set to 1, 3, 4, 5, 6 or 7. If you change this parameter, the phone will reboot to make the change take effect. Web User Interface: Network->Advanced->802.1x->MD5 Password Phone User Interface: Tap  ->Settings->Device Settings->Network(default password: admin)->802.1x->MD5 Password		
static.network.802_1x.root_cert_url	URL within 511 characters	Blank

Parameters	Permitted Values	Default
Description: Configures the access URL of the CA certificate. Example: <code>static.network.802_1x.root_cert_url = http://192.168.1.10/ca.pem</code> Note: It works only if the value of the parameter "static.network.802_1x.mode" is set to 2, 3, 4, 5, 6 or 7. The format of the certificate must be *.pem, *.crt, *.cer or *.der. Web User Interface: Network->Advanced->802.1x->CA Certificates Phone User Interface: None		
static.network.802_1x.client_cert_url	URL within 511 characters	Blank
Description: Configures the access URL of the device certificate. Example: <code>static.network.802_1x.client_cert_url = http://192.168.1.10/client.pem</code> Note: It works only if the value of the parameter "static.network.802_1x.mode" is set to 2 (EAP-TLS). The format of the certificate must be *.pem. Web User Interface: Network->Advanced->802.1x->Device Certificates Phone User Interface: None		

The following shows an example of the EAP-TLS protocol for 802.1X authentication in configuration files:

```
network.802_1x.mode = 2
network.802_1x.identity = yealink
network.802_1x.root_cert_url = http://192.168.1.8:8080/ca.crt
network.802_1x.client_cert_url = http:// 192.168.1.8:8080/client.pem
```

2. Upload configuration files, CA certificate and client certificate to the root directory of the configuration server.

Applying the Configuration Files to your Phone

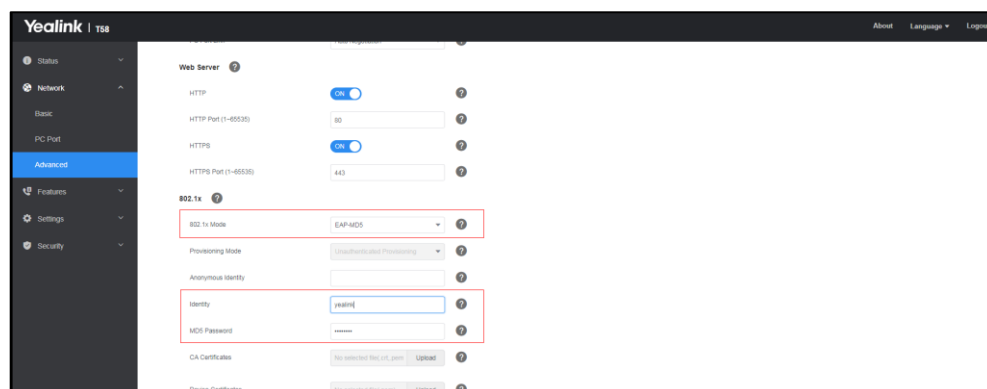
Once you have edited a configuration file (e.g., y0000000000xx.cfg) using the parameters introduced above, you need to do the following to apply the files to your phone:

1. Connect your phone to a network that is not 802.1X-enabled.

2. Perform the auto provisioning process to apply the configuration files to the phone.
Then the Teams phone will reboot to make the settings effective.
For more information on auto provisioning, refer to [Yealink_Teams_HD_IP_Phones_Auto_Provisioning_Guide](#).
3. Connect the phone to the 802.1X-enabled network and reboot the phone.
You can make a phone call to verify whether the phone is authenticated.

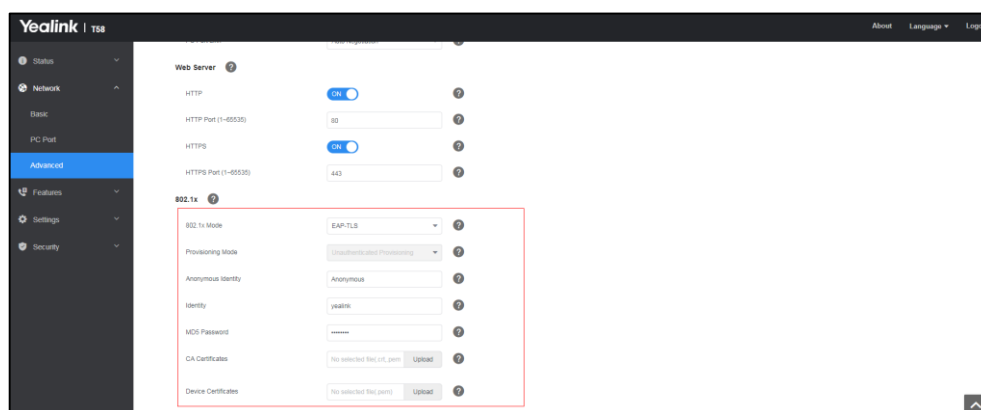
Configuring 802.1X via Web User Interface

1. Connect your phone to a network that is not 802.1X-enabled.
2. Login to the web user interface of the phone.
3. Click on **Network->Advanced**.
4. In the **802.1x** block, select the desired protocol from the pull-down list of **802.1x Mode**.
 - a) If you select **EAP-MD5**:
 - 1) Enter the user name for authentication in the **Identity** field.
 - 2) Enter the password for authentication in the **MD5 Password** field.



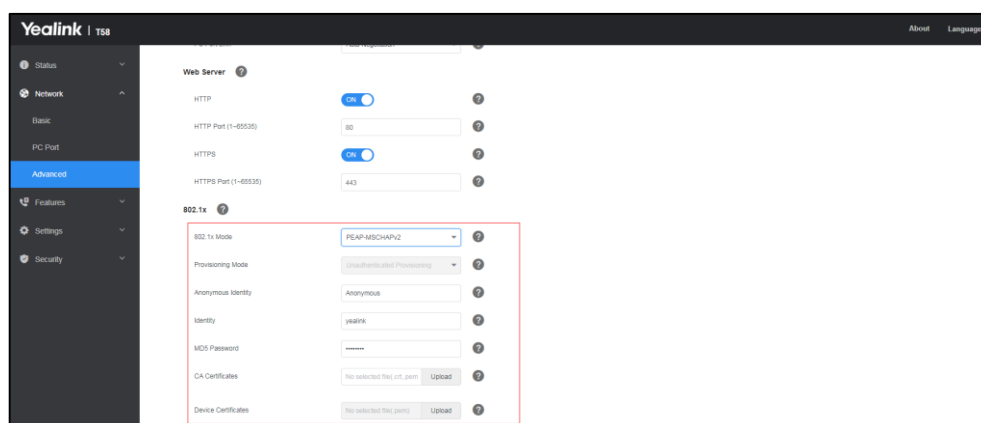
- b) If you select **EAP-TLS**:
 - 1) Enter the user name for authentication in the **Identity** field.
 - 2) Leave the **MD5 Password** field blank.
 - 3) In the **CA Certificates** field, click the blank box to select the desired CA certificate (*.pem, *.crt, *.cer or *.der) from your local system.
 - 4) In the **Device Certificates** field, click the blank box to select the desired client (*.pem or *.cer) certificate from your local system.

5) Click **Upload** to upload the certificates.



c) If you select **EAP-PEAP/MSCHAPv2**:

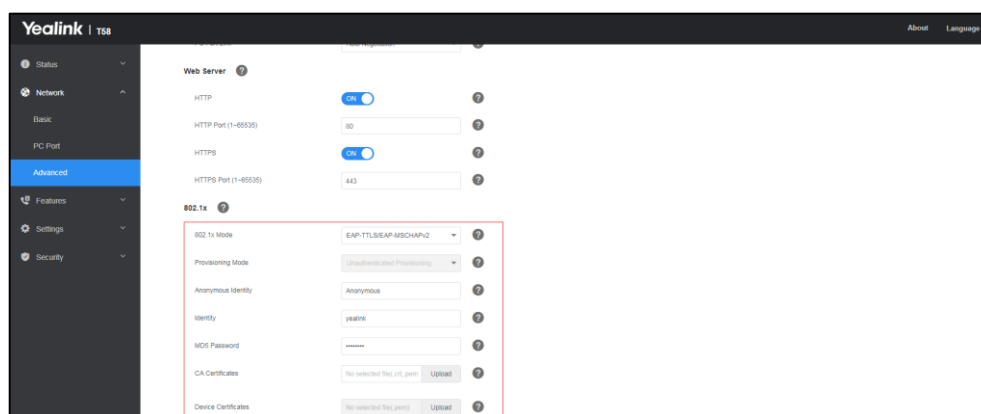
- 1) Enter the user name for authentication in the **Identity** field.
- 2) Enter the password for authentication in the **MD5 Password** field.
- 3) In the **CA Certificates** field, click the blank box to select the desired CA certificate (*.pem, *.crt, *.cer or *.der) from your local system.



d) If you select **EAP-TTLS/EAP-MSCHAPv2**:

- 1) Enter the user name for authentication in the **Identity** field.
- 2) Enter the password for authentication in the **MD5 Password** field.
- 3) In the **CA Certificates** field, click the blank box to select the desired CA certificate (*.pem, *.crt, *.cer or *.der) from your local system.

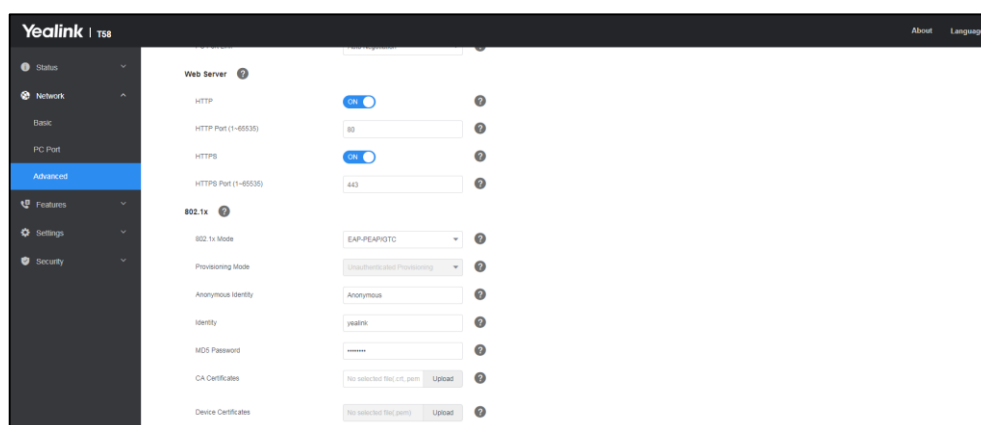
4) Click **Upload** to upload the certificate.



The screenshot shows the Yealink T58 Advanced settings page. The left sidebar has a menu with 'Status', 'Network', 'Basic', 'PC Port', 'Advanced' (selected), 'Features', 'Settings', and 'Security'. The main content area is titled 'Web Server' and contains settings for HTTP and HTTPS. Below this is the '802.1x' section, which is highlighted with a red box. The 802.1x settings include: '802.1x Mode' set to 'EAP-TLS/EAP-MSCHAPv2', 'Provisioning Mode' set to 'Local/Remote Provisioning', 'Anonymous Identity' set to 'Anonymous', 'Identity' set to 'yealink', 'MD5 Password' set to an empty field, 'CA Certificates' set to 'No selected file (.crt, .pem)' with an 'Upload' button, and 'Device Certificates' set to 'No selected file (.pem)' with an 'Upload' button.

e) If you select **EAP-PEAP/GTC**:

- 1) Enter the user name for authentication in the **Identity** field.
- 2) Enter the password for authentication in the **MD5 Password** field.
- 3) In the **CA Certificates** field, click the blank box to select the desired CA certificate (*.pem, *.crt, *.cer or *.der) from your local system.



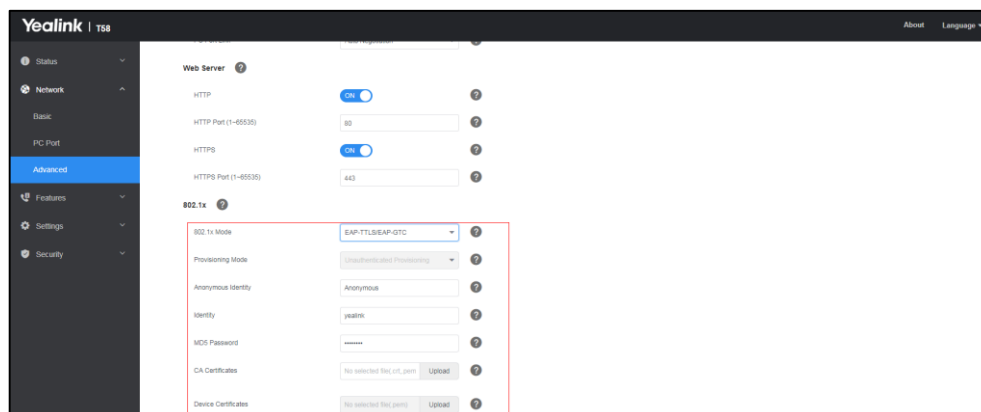
The screenshot shows the Yealink T58 Advanced settings page. The left sidebar has a menu with 'Status', 'Network', 'Basic', 'PC Port', 'Advanced' (selected), 'Features', 'Settings', and 'Security'. The main content area is titled 'Web Server' and contains settings for HTTP and HTTPS. Below this is the '802.1x' section, which is highlighted with a red box. The 802.1x settings include: '802.1x Mode' set to 'EAP-PEAP/GTC', 'Provisioning Mode' set to 'Local/Remote Provisioning', 'Anonymous Identity' set to 'Anonymous', 'Identity' set to 'yealink', 'MD5 Password' set to an empty field, 'CA Certificates' set to 'No selected file (.crt, .pem)' with an 'Upload' button, and 'Device Certificates' set to 'No selected file (.pem)' with an 'Upload' button.

4) Click **Upload** to upload the certificate.

f) If you select **EAP-TTLS/EAP-GTC**:

- 1) Enter the user name for authentication in the **Identity** field.
- 2) Enter the password for authentication in the **MD5 Password** field.

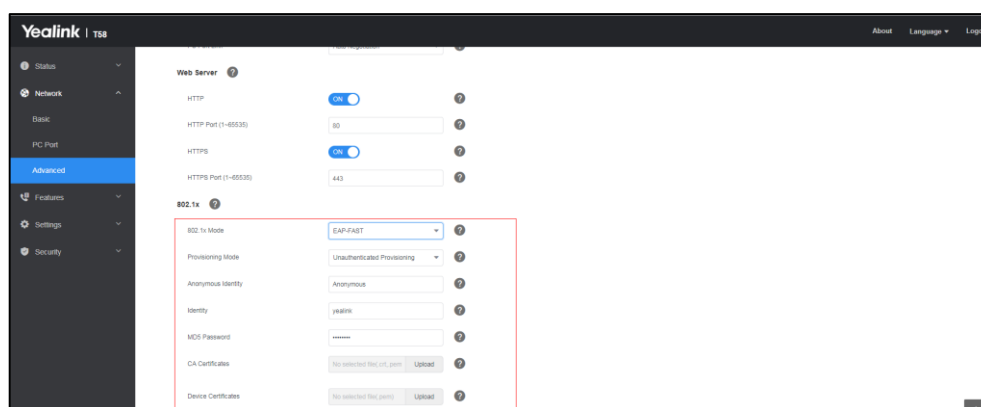
- 3) In the **CA Certificates** field, click the blank box to select the desired CA certificate (*.pem, *.crt, *.cer or *.der) from your local system.



- 4) Click **Upload** to upload the certificate.

g) If you select **EAP-FAST:**

- 1) Select the desired value from the pull-down list of **Provisioning Mode**.
- 2) Enter the user name for authentication in the **Identity** field.
- 3) Enter the password for authentication in the **MD5 Password** field.
- 3) In the **CA Certificates** field, click the blank box to select the desired CA certificate (*.pem, *.crt, *.cer or *.der) from your local system.



- 4) Click **Upload** to upload the certificate.

5. Click **Confirm** to accept the change.

A dialog box pops up to prompt that settings will take effect after a reboot.

6. Click **OK** to reboot the phone.

7. Connect the phone to the 802.1X-enabled network after reboot.

Note



If the Pass-thru mode is available on your new phone, you can select the Pass-thru mode from the pull-down list of **DOT1XSTAT Options** via web user interface.

Configuring 802.1X via Phone User Interface

If you select EAP-PEAP/MSCHAPv2, EAP-TTLS/EAP-MSCHAPv2, EAP-PEAP/GTC, EAP-TTLS/EAP-GTC or EAP-FAST mode, you should upload CA certificate in advance using configuration files or via web user interface.

If you select EAP-TLS mode, you should upload CA certificate and device certificate in advance using configuration files or via web user interface.

To configure the 802.1X authentication via phone user interface:

1. Tap  -> **Settings**->**Device Settings**->**Network**(default password: admin)->**802.1x**.
2. Select the desired value from the **802.1x Mode** field.
A dialog box pops up to prompt that the settings will take effect after a reboot.
3. Tap **CANCEL**.
4. Enter the user name for authentication in the **Identity** field.
5. Enter the password for authentication in the **MD5 Password** field.
6. Tap  to accept the change.
A dialog box pops up to prompt that the settings will take effect after a reboot.
7. Tap **OK** to reboot the phone.
The phone reboots automatically to make the settings effective after a period of time.

802.1X Authentication Process

Reboot the phone to activate the 802.1X authentication on the phone. The 802.1X authentication process is divided into two basic stages:

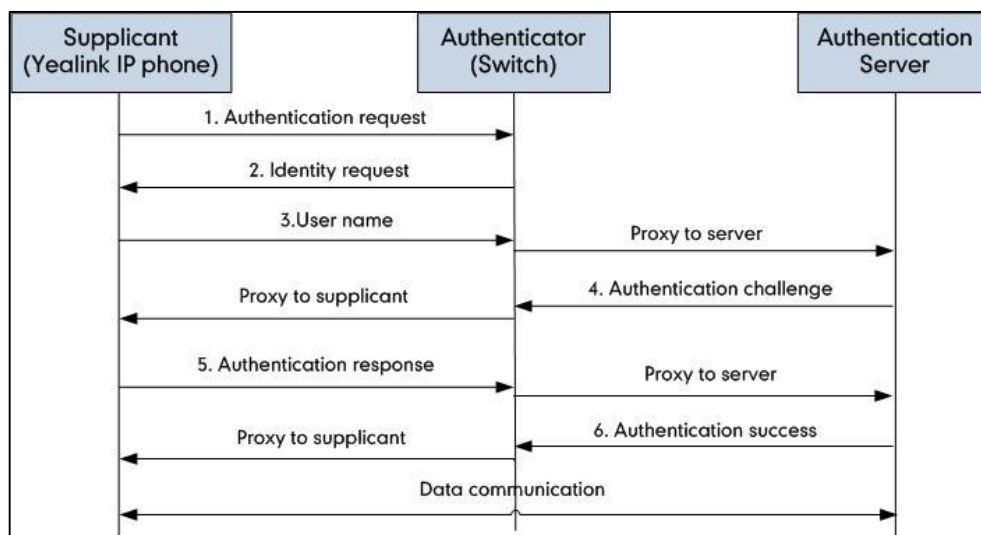
Pre-authentication

The 802.1X pre-authentication process begins with the Teams phone that contains a supplicant service used for negotiation and authentication. When the Teams phone connects to an unauthorized port, the authenticator blocks the Teams phone from connecting to the network. Using one of the authentication protocols, the authenticator establishes a security negotiation with the Teams phone and creates an 802.1X session. The Teams phone provides its authentication information for the authenticator, and then the authenticator forwards the information to the authentication server.

Authentication

After the authentication server authenticates the Teams phone, the authentication server initiates the authentication stage of the process. During this phase, the authenticator facilitates an exchange of keys between the Teams phone and the authentication server. After these keys are established, the authenticator grants the Teams phone access to the protected network on an authorized port.

The following figure summarizes an implementation of the 802.1X authentication process using a RADIUS server as the authentication server:



For more details about the 802.1X authentication process using EAP-MD5, EAP-TLS, EAP-PEAP/MSCHAPv2, EAP-TTLS/EAP-MSCHAPv2, EAP-PEAP/GTC, EAP-TTLS/EAP-GTC and EAP-FAST protocols, refer to [Appendix B: 802.1X Authentication Process](#) on page 16.

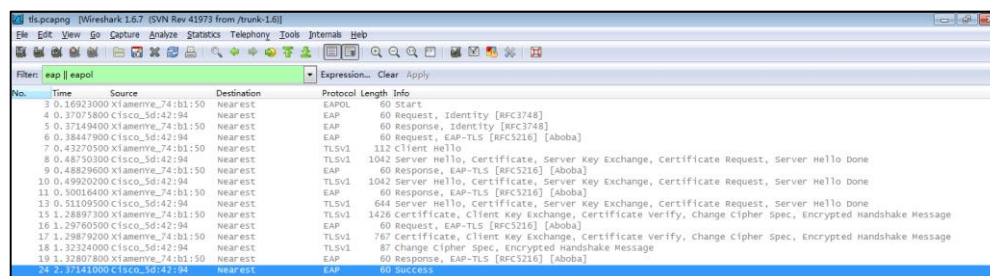
If you are interested in the packets exchanged during the authentication process, we recommend you to use the Wireshark tool. Refer to <http://wiki.wireshark.org> for more information about the Wireshark tool.

The following screenshot of the Wireshark shows a sample of a successful authentication process using the EAP-MD5 protocol:

The screenshot shows a Wireshark packet capture of an EAP-MD5 authentication process. The filter is set to 'eap || eapol'. The packet list shows the following details:

No.	Time	Source	Destination	Protocol	Length	Info
18	2.10898200	x1amennve_74:b1:50	Nearest	EAPOL	60	Start
19	2.11212800	Cisco_5d:42:94	Nearest	EAP	60	Request, Identity [RFC3748]
20	2.11322800	x1amennve_74:b1:50	Nearest	EAP	60	Response, Identity [RFC3748]
21	2.12152900	Cisco_5d:42:94	Nearest	EAP	60	Request, EAP-TLS [RFC5216] [aboba]
22	2.12225600	x1amennve_74:b1:50	Nearest	EAP	60	Response, Legacy Nak (Response only) [RFC3748]
23	2.13013600	Cisco_5d:42:94	Nearest	EAP	60	Request, MD5-Challenge [RFC3748]
24	2.13134100	x1amennve_74:b1:50	Nearest	EAP	60	Response, MD5-Challenge [RFC3748]
38	3.17184100	Cisco_5d:42:94	Nearest	EAP	60	Success

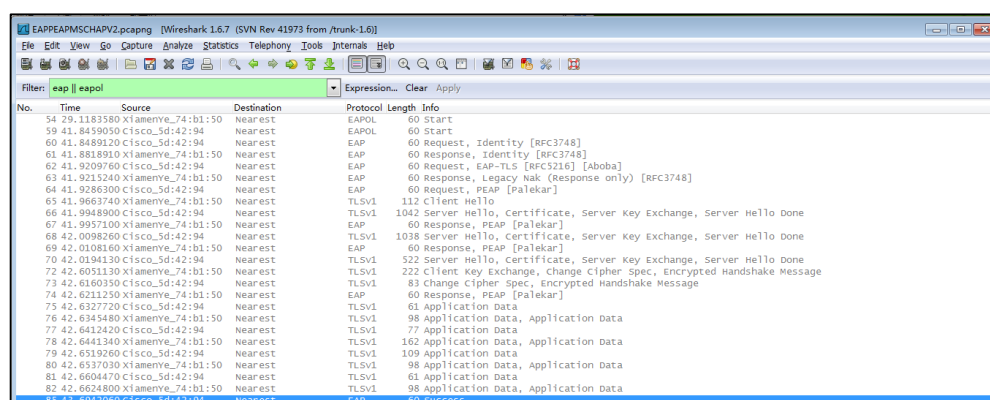
The following screenshot of the Wireshark shows a sample of a successful authentication process using the EAP-TLS protocol:



The screenshot shows a Wireshark capture of an EAP-TLS authentication process. The filter is set to 'eap || eapol'. The packet list shows the following sequence of events:

No.	Time	Source	Destination	Protocol	Length	Info
3	0.16923000	Xlanemv_74:b1:50	Nearst	EAPOL	60	Start
4	0.37075800	Cisco_Sd:42:94	Nearst	EAP	60	Request, Identity [RFC3748]
5	0.37149400	Xlanemv_74:b1:50	Nearst	EAP	60	Response, Identity [RFC3748]
6	0.38479800	Cisco_Sd:42:94	Nearst	EAP	60	Request, EAP-TLS [RFC5216] [Aboba]
7	0.43270500	Xlanemv_74:b1:50	Nearst	TLSv1	112	Client Hello
8	0.48750300	Cisco_Sd:42:94	Nearst	TLSv1	1042	Server Hello, Certificate, Server Key Exchange, Certificate Request, Server Hello Done
9	0.48829600	Xlanemv_74:b1:50	Nearst	EAP	60	Response, EAP-TLS [RFC5216] [Aboba]
10	0.49920200	Cisco_Sd:42:94	Nearst	TLSv1	1042	Server Hello, Certificate, Server Key Exchange, Certificate Request, Server Hello Done
11	0.50016400	Xlanemv_74:b1:50	Nearst	EAP	60	Response, EAP-TLS [RFC5216] [Aboba]
13	0.51109500	Cisco_Sd:42:94	Nearst	TLSv1	644	Server Hello, Certificate, Server Key Exchange, Certificate Request, Server Hello Done
15	1.28807300	Xlanemv_74:b1:50	Nearst	TLSv1	1426	Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake Message
16	1.29760500	Cisco_Sd:42:94	Nearst	EAP	60	Request, EAP-TLS [RFC5216] [Aboba]
17	1.29879200	Xlanemv_74:b1:50	Nearst	TLSv1	787	Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake Message
18	1.32240000	Cisco_Sd:42:94	Nearst	TLSv1	87	Change Cipher Spec, Encrypted Handshake Message
19	1.32807800	Xlanemv_74:b1:50	Nearst	EAP	60	Response, EAP-TLS [RFC5216] [Aboba]
24	2.37141000	Cisco_Sd:42:94	Nearst	EAP	60	Success

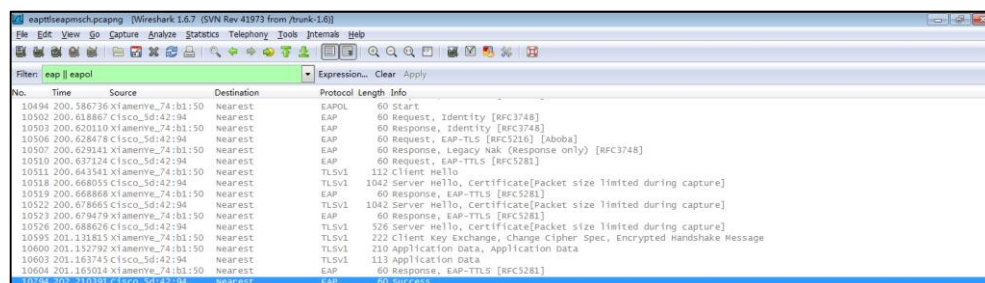
The following screenshot of the Wireshark shows a sample of a successful authentication process using the EAP-PEAP/MSCHAPv2 protocol:



The screenshot shows a Wireshark capture of an EAP-PEAP/MSCHAPv2 authentication process. The filter is set to 'eap || eapol'. The packet list shows the following sequence of events:

No.	Time	Source	Destination	Protocol	Length	Info
54	29.1183580	Xlanemv_74:b1:50	Nearst	EAPOL	60	Start
59	41.8459050	Cisco_Sd:42:94	Nearst	EAPOL	60	Start
60	41.8489120	Cisco_Sd:42:94	Nearst	EAP	60	Request, Identity [RFC3748]
61	41.8818910	Xlanemv_74:b1:50	Nearst	EAP	60	Response, Identity [RFC3748]
62	41.9209760	Cisco_Sd:42:94	Nearst	EAP	60	Request, EAP-TLS [RFC5216] [Aboba]
63	41.9215240	Xlanemv_74:b1:50	Nearst	EAP	60	Response, Legacy Nak (Response only) [RFC3748]
64	41.9286300	Cisco_Sd:42:94	Nearst	EAP	60	Request, PEAP [Pakekar]
65	41.9663740	Xlanemv_74:b1:50	Nearst	TLSv1	112	Client Hello
66	41.9948900	Cisco_Sd:42:94	Nearst	TLSv1	1042	Server Hello, Certificate, Server Key Exchange, Server Hello Done
67	41.9957100	Xlanemv_74:b1:50	Nearst	EAP	60	Response, PEAP [Pakekar]
68	42.0098260	Cisco_Sd:42:94	Nearst	TLSv1	1038	Server Hello, Certificate, Server Key Exchange, Server Hello Done
69	42.0108160	Xlanemv_74:b1:50	Nearst	EAP	60	Response, PEAP [Pakekar]
70	42.0194130	Cisco_Sd:42:94	Nearst	TLSv1	522	Server Hello, Certificate, Server Key Exchange, Server Hello Done
72	42.6051130	Xlanemv_74:b1:50	Nearst	TLSv1	222	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
73	42.6160350	Cisco_Sd:42:94	Nearst	TLSv1	83	Change Cipher Spec, Encrypted Handshake Message
74	42.6211250	Xlanemv_74:b1:50	Nearst	EAP	60	Response, PEAP [Pakekar]
75	42.6327720	Cisco_Sd:42:94	Nearst	TLSv1	61	Application Data
76	42.6345480	Xlanemv_74:b1:50	Nearst	TLSv1	98	Application Data, Application Data
77	42.6412420	Cisco_Sd:42:94	Nearst	TLSv1	77	Application Data
78	42.6441340	Xlanemv_74:b1:50	Nearst	TLSv1	162	Application Data, Application Data
79	42.6519260	Cisco_Sd:42:94	Nearst	TLSv1	109	Application Data
80	42.6537030	Xlanemv_74:b1:50	Nearst	TLSv1	98	Application Data, Application Data
81	42.6604470	Cisco_Sd:42:94	Nearst	TLSv1	61	Application Data
82	42.6624800	Xlanemv_74:b1:50	Nearst	TLSv1	98	Application Data, Application Data
83	43.6842060	Cisco_Sd:42:94	Nearst	EAP	60	Success

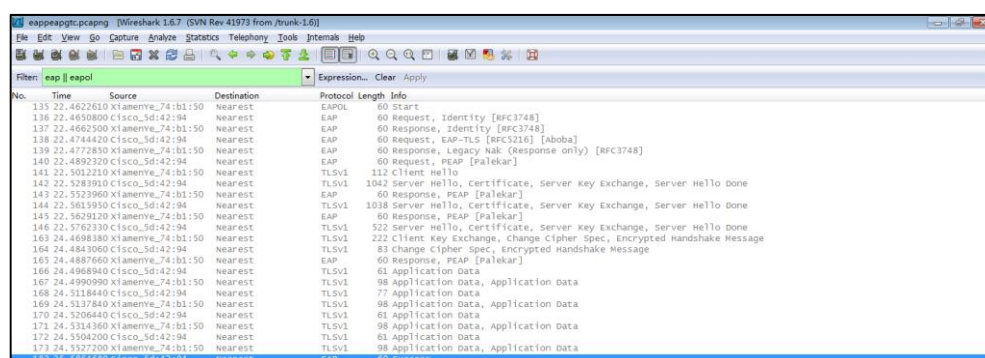
The following screenshot of the Wireshark shows a sample of a successful authentication process using the EAP-TTLS/EAP-MSCHAPv2 protocol:



The screenshot shows a Wireshark capture of an EAP-TTLS/EAP-MSCHAPv2 authentication process. The filter is set to 'eap || eapol'. The packet list shows the following sequence of events:

No.	Time	Source	Destination	Protocol	Length	Info
10494	200.586736	Xlanemv_74:b1:50	Nearst	EAPOL	60	Start
10502	200.618867	Cisco_Sd:42:94	Nearst	EAP	60	Request, Identity [RFC3748]
10503	200.620110	Xlanemv_74:b1:50	Nearst	EAP	60	Response, Identity [RFC3748]
10506	200.628478	Cisco_Sd:42:94	Nearst	EAP	60	Request, EAP-TLS [RFC5216] [Aboba]
10507	200.629141	Xlanemv_74:b1:50	Nearst	EAP	60	Response, Legacy Nak (Response only) [RFC3748]
10510	200.637124	Cisco_Sd:42:94	Nearst	EAP	60	Request, EAP-TTLS [RFC5281]
10511	200.643541	Xlanemv_74:b1:50	Nearst	TLSv1	112	Client Hello
10518	200.668055	Cisco_Sd:42:94	Nearst	TLSv1	1042	Server Hello, Certificate [Packet size limited during capture]
10519	200.668868	Xlanemv_74:b1:50	Nearst	EAP	60	Response, EAP-TTLS [RFC5281]
10522	200.678665	Cisco_Sd:42:94	Nearst	TLSv1	1042	Server Hello, Certificate [Packet size limited during capture]
10523	200.679479	Xlanemv_74:b1:50	Nearst	EAP	60	Response, EAP-TTLS [RFC5281]
10526	200.686026	Cisco_Sd:42:94	Nearst	TLSv1	526	Server Hello, Certificate [Packet size limited during capture]
10595	201.131815	Xlanemv_74:b1:50	Nearst	TLSv1	222	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
10600	201.152792	Xlanemv_74:b1:50	Nearst	TLSv1	210	Application Data, Application Data
10603	201.163745	Cisco_Sd:42:94	Nearst	TLSv1	113	Application Data
10604	201.165024	Xlanemv_74:b1:50	Nearst	EAP	60	Response, EAP-TTLS [RFC5281]
10794	202.210391	Cisco_Sd:42:94	Nearst	EAP	60	Success

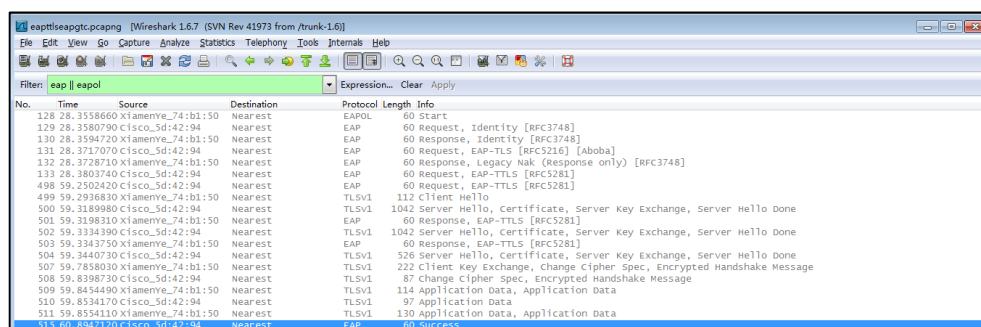
The following screenshot of the Wireshark shows a sample of a successful authentication process using the EAP-PEAP/GTC protocol:



The screenshot shows a Wireshark capture of an EAP-PEAP/GTC authentication process. The filter is set to 'eap || eapol'. The packet list shows the following sequence of events:

No.	Time	Source	Destination	Protocol	Length	Info
135	22.4622610	Xlanemv_74:b1:50	Nearst	EAPOL	60	Start
136	22.4650800	Cisco_Sd:42:94	Nearst	EAP	60	Request, Identity [RFC3748]
137	22.4662500	Xlanemv_74:b1:50	Nearst	EAP	60	Response, Identity [RFC3748]
138	22.4744420	Cisco_Sd:42:94	Nearst	EAP	60	Request, EAP-TLS [RFC5216] [Aboba]
139	22.4772850	Xlanemv_74:b1:50	Nearst	EAP	60	Response, Legacy Nak (Response only) [RFC3748]
140	22.4892320	Cisco_Sd:42:94	Nearst	EAP	60	Request, PEAP [Pakekar]
141	22.5012210	Xlanemv_74:b1:50	Nearst	TLSv1	112	Client Hello
142	22.5283910	Cisco_Sd:42:94	Nearst	TLSv1	1042	Server Hello, Certificate, Server Key Exchange, Server Hello Done
143	22.5523960	Xlanemv_74:b1:50	Nearst	EAP	60	Response, PEAP [Pakekar]
144	22.5615950	Cisco_Sd:42:94	Nearst	TLSv1	1038	Server Hello, Certificate, Server Key Exchange, Server Hello Done
145	22.5629120	Xlanemv_74:b1:50	Nearst	EAP	60	Response, PEAP [Pakekar]
146	22.5762330	Cisco_Sd:42:94	Nearst	TLSv1	522	Server Hello, Certificate, Server Key Exchange, Server Hello Done
149	24.4688380	Xlanemv_74:b1:50	Nearst	TLSv1	222	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
164	24.4843060	Cisco_Sd:42:94	Nearst	TLSv1	83	Change Cipher Spec, Encrypted Handshake Message
165	24.4887660	Xlanemv_74:b1:50	Nearst	EAP	60	Response, PEAP [Pakekar]
166	24.4968940	Cisco_Sd:42:94	Nearst	TLSv1	61	Application Data
167	24.4990990	Xlanemv_74:b1:50	Nearst	TLSv1	98	Application Data, Application Data
168	24.5118440	Cisco_Sd:42:94	Nearst	TLSv1	77	Application Data
169	24.5137840	Xlanemv_74:b1:50	Nearst	TLSv1	98	Application Data, Application Data
170	24.5206440	Cisco_Sd:42:94	Nearst	TLSv1	61	Application Data
171	24.5314360	Xlanemv_74:b1:50	Nearst	TLSv1	98	Application Data, Application Data
172	24.5504200	Cisco_Sd:42:94	Nearst	TLSv1	61	Application Data
173	24.5527200	Xlanemv_74:b1:50	Nearst	TLSv1	98	Application Data, Application Data
182	25.5864680	Cisco_Sd:42:94	Nearst	EAP	60	Success

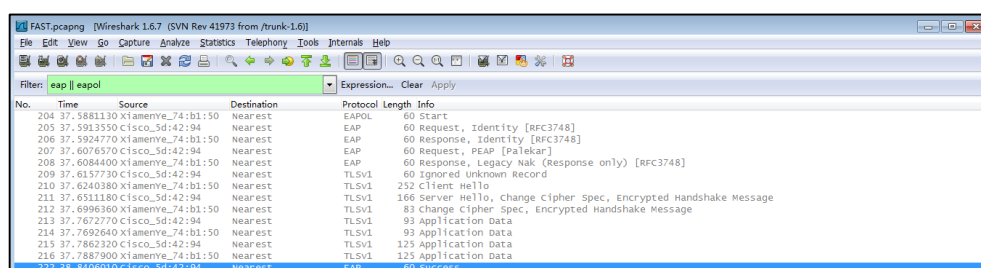
The following screenshot of the Wireshark shows a sample of a successful authentication process using the EAP-TTLS/EAP-GTC protocol:



The screenshot shows a Wireshark capture of an EAP-TTLS/EAP-GTC authentication process. The filter is set to 'eap || eapol'. The packet list shows the following sequence of events:

No.	Time	Source	Destination	Protocol	Length	Info
128	28.3558660	xiamenve_74:b1:50	Nearest	EAPOL	60	Start
129	28.3580790	Cisco_Sd:42:94	Nearest	EAP	60	Request, Identity [RFC3748]
130	28.3584720	xiamenve_74:b1:50	Nearest	EAP	60	Response, Identity [RFC3748]
131	28.3717070	Cisco_Sd:42:94	Nearest	EAP	60	Request, EAP-TLS [RFC5216] [Aboba]
132	28.3728710	xiamenve_74:b1:50	Nearest	EAP	60	Response, Legacy Nak (Response only) [RFC3748]
133	28.3803740	Cisco_Sd:42:94	Nearest	EAP	60	Request, EAP-TTLS [RFC5281]
498	59.2502420	Cisco_Sd:42:94	Nearest	EAP	60	Request, EAP-TTLS [RFC5281]
499	59.2936830	xiamenve_74:b1:50	Nearest	TLSv1	112	Client Hello
500	59.3189980	Cisco_Sd:42:94	Nearest	TLSv1	1042	Server Hello, Certificate, Server Key Exchange, Server Hello Done
501	59.3198310	xiamenve_74:b1:50	Nearest	EAP	60	Response, EAP-TTLS [RFC5281]
502	59.3334390	Cisco_Sd:42:94	Nearest	TLSv1	1042	Server Hello, Certificate, Server Key Exchange, Server Hello Done
503	59.3343750	xiamenve_74:b1:50	Nearest	EAP	60	Response, EAP-TTLS [RFC5281]
504	59.3440730	Cisco_Sd:42:94	Nearest	TLSv1	526	Server Hello, Certificate, Server Key Exchange, Server Hello Done
507	59.7858030	xiamenve_74:b1:50	Nearest	TLSv1	222	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
508	59.8398730	Cisco_Sd:42:94	Nearest	TLSv1	87	Change Cipher Spec, Encrypted Handshake Message
509	59.8454490	xiamenve_74:b1:50	Nearest	TLSv1	114	Application Data, Application Data
510	59.8534170	Cisco_Sd:42:94	Nearest	TLSv1	97	Application Data
511	59.8554110	xiamenve_74:b1:50	Nearest	TLSv1	130	Application Data, Application Data
513	60.8947170	Cisco_Sd:42:94	Nearest	EAP	60	Success

The following screenshot of the Wireshark shows a sample of a successful authentication process using the EAP-FAST protocol:



The screenshot shows a Wireshark capture of an EAP-FAST authentication process. The filter is set to 'eap || eapol'. The packet list shows the following sequence of events:

No.	Time	Source	Destination	Protocol	Length	Info
204	37.5881130	xiamenve_74:b1:50	Nearest	EAPOL	60	Start
205	37.5913550	Cisco_Sd:42:94	Nearest	EAP	60	Request, Identity [RFC3748]
206	37.5924770	xiamenve_74:b1:50	Nearest	EAP	60	Response, Identity [RFC3748]
207	37.6076570	Cisco_Sd:42:94	Nearest	EAP	60	Request, PEAP [Paklekar]
208	37.6084400	xiamenve_74:b1:50	Nearest	EAP	60	Response, Legacy Nak (Response only) [RFC3748]
209	37.6157730	Cisco_Sd:42:94	Nearest	TLSv1	60	Ignored unknown Record
210	37.6240380	xiamenve_74:b1:50	Nearest	TLSv1	252	Client Hello
211	37.6511180	Cisco_Sd:42:94	Nearest	TLSv1	166	Server Hello, change Cipher Spec, Encrypted Handshake Message
212	37.6996360	xiamenve_74:b1:50	Nearest	TLSv1	83	Change Cipher Spec, Encrypted Handshake Message
213	37.7672770	Cisco_Sd:42:94	Nearest	TLSv1	93	Application Data
214	37.7692640	xiamenve_74:b1:50	Nearest	TLSv1	93	Application Data
215	37.7862320	Cisco_Sd:42:94	Nearest	TLSv1	125	Application Data
216	37.7887900	xiamenve_74:b1:50	Nearest	TLSv1	125	Application Data
222	38.2406010	Cisco_Sd:42:94	Nearest	EAP	60	Success

Troubleshooting

Why doesn't the phone pass 802.1X authentication?

Do the following in sequence:

- Ensure that the 802.1X authentication environment is operational.
 - a) Connect another device (e.g., a computer) to the switch port.
 - b) Check if the device is authenticated successfully, and an IP address is assigned to it. If the device fails the authentication, check the configurations on the switch and authentication server.
- Ensure that the user name and password configured on the phone are correct. If EAP-TLS, EAP-PEAP/MSCHAPv2, EAP-TTLS/EAP-MSCHAPv2, EAP-PEAP/GTC, EAP-TTLS/EAP-GTC and EAP-FAST protocols are used, ensure that the certificate uploaded to the phone is valid.
 - a) Double click the certificate to check the validity time.
 - b) Check if the time and date on the phone is within the validity time of the uploaded certificate. If not, re-generate a certificate and upload it the phone.
- Ensure that the failure is not caused by network settings.
 - a) Disable LLDP feature and manually configure a VLAN ID for the Internet port of the phone to check if the authentication is successful. If the phone is authenticated

successfully, contact your network administrator to troubleshoot the LLDP-related problem.

- b)** Disable VLAN feature on the phone to check if the authentication passes successfully. If the phone is authenticated successfully, capture the packet and feed back to your network administrator.
- Contact Yealink FAE for support when the above steps cannot solve your problem.
 - a)** Capture the packet and export configurations of the phone, switch and authentication server.
 - b)** Provide the related information to Yealink FAE.

Appendix A: Glossary

IEEE (Institute of Electrical and Electronics Engineers) –A professional association headquartered in New York City that is dedicated to advancing technological innovation and excellence.

802.1X –A port-based network access control, meaning it only provides an authentication mechanism for devices wishing to attach to a LAN.

EAP (Extensible Authentication Protocol) –An authentication framework which supports multiple authentication methods.

TLS (Transport Layer Security) –Provides for mutual authentication, integrity-protected cipher suite negotiation between two endpoints.

MD5 (Message-Digest Algorithm) –Only provides authentication of the EAP peer for the EAP server but not mutual authentication.

PEAP (Protected Extensible Authentication Protocol) –A protocol that encapsulates the EAP within an encrypted and authenticated TLS tunnel.

MSCHAPv2 (Microsoft Challenge Handshake Authentication Protocol version 2) –Provides for mutual authentication, but does not require a supplicant-side certificate.

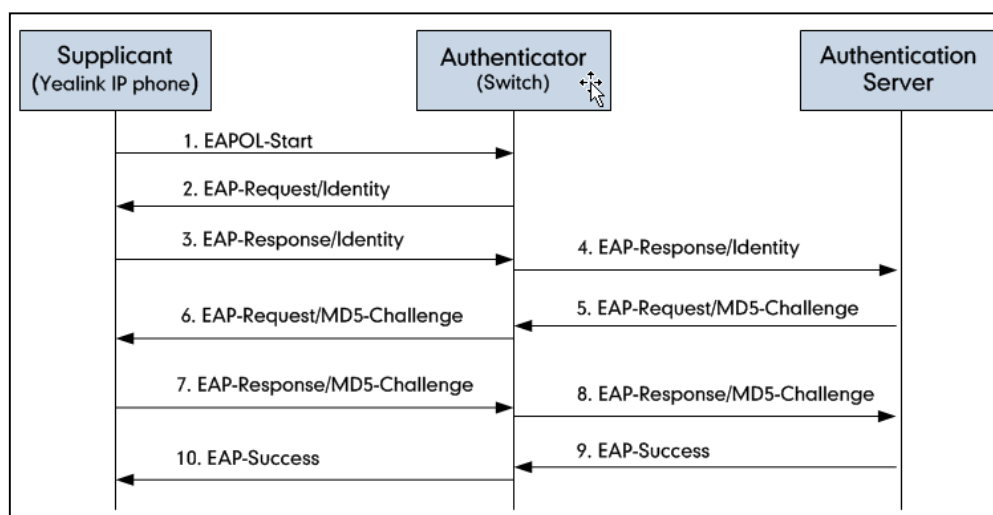
TTLS (Tunneled Transport Layer Security) –Extends TLS to improve some weak points, but it does not require a supplicant-side certificate.

EAPOL (Extensible Authentication Protocol over Local Area Network) –A delivery mechanism and doesn't provide the actual authentication mechanisms.

Appendix B: 802.1X Authentication Process

A Successful Authentication Using EAP-MD5 Protocol

The following figure illustrates the scenario of a successful 802.1X authentication process using the EAP-MD5 protocol.



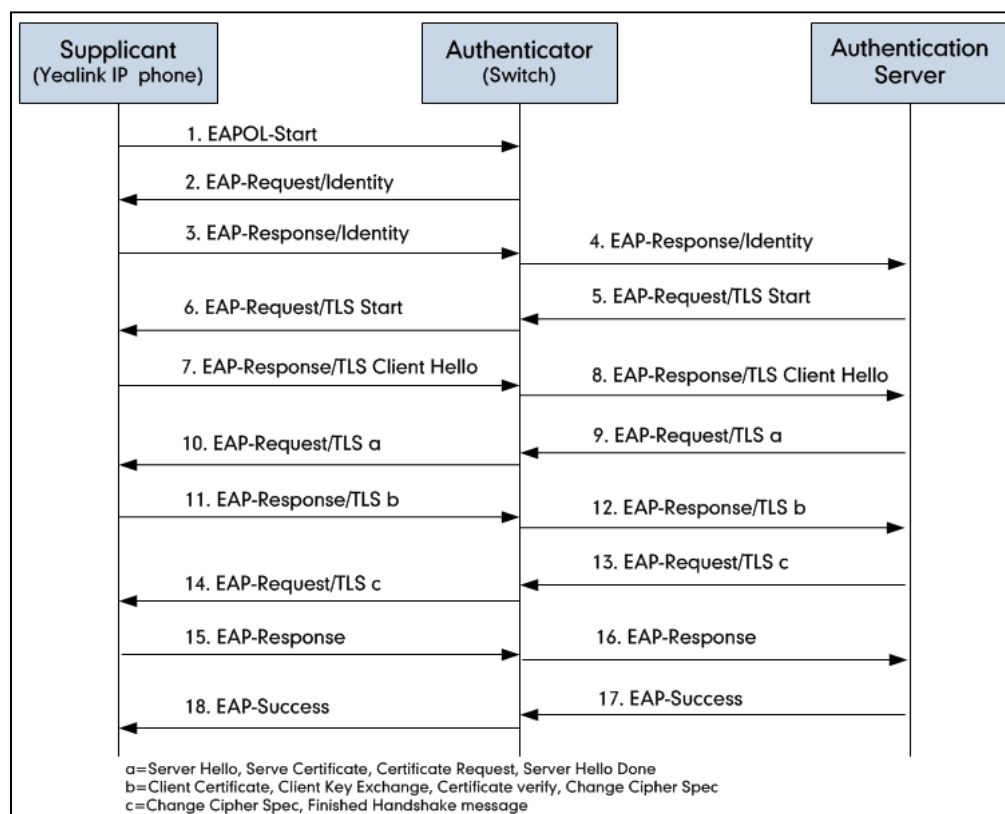
1. The supplicant sends an "EAPOL-Start" packet to the authenticator.
2. The authenticator responds with an "EAP-Request/Identity" packet to the supplicant.
3. The supplicant responds with an "EAP-Response/Identity" packet to the authenticator.
4. The authenticator strips the Ethernet header and encapsulates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

After the supplicant is authenticated successfully, the authenticator provides network access permissions. If the supplicant does not provide proper identification, the authentication server responds with a rejection message. The authenticator passes the message onto the supplicant and blocks access to the LAN.

If the supplicant is disabled or reset after successful authentication, the supplicant sends an EAPOL-Logoff message, which prompts the authenticator to block access to the LAN.

A Successful Authentication Using EAP-TLS Protocol

The following figure illustrates the scenario of a successful 802.1X authentication process using the EAP-TLS protocol.



1. The supplicant sends an "EAPOL-Start" packet to the authenticator.
2. The authenticator responds with an "EAP-Request/Identity" packet to the supplicant.
3. The supplicant responds with an "EAP-Response/Identity" packet to the authenticator.
4. The authenticator strips the Ethernet header and encapsulates the remaining EAP frame in the RADIUS format, and then sends it to the authentication 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 authenticator.
6. The authenticator strips the authentication server's frame header, encapsulates the remaining EAP frame in the EAPOL format, and then sends it to the supplicant.
7. The supplicant responds with an "EAP-Response" packet containing a TLS client hello handshake message to the authenticator. The client hello message includes the TLS version supported by the supplicant, a session ID, a random number and a set of cipher suites.
8. The authenticator passes the response to the authentication server.
9. The authentication server sends an "EAP-Request" packet to the authenticator. The packet includes a TLS server hello handshake message, a server certificate message, a certificate request message and a server hello done message.
10. The authenticator passes the request to the supplicant.

- 11.** The supplicant responds with an "EAP-Response" packet to the authenticator. The packet includes a TLS change cipher spec message, a client certificate message, a client key exchange message and a certificate verify message.
- 12.** The authenticator passes the response to the authentication server.
- 13.** The authentication server sends an "EAP-Request" packet to the authenticator. The packet includes a TLS change cipher spec message and a finished handshake message. The change cipher spec message is sent to notify the authenticator that subsequent records will be protected under the newly negotiated cipher spec.
- 14.** The authenticator passes the request to the supplicant.
- 15.** The supplicant responds with an "EAP-Response" packet to the authenticator.
- 16.** The authenticator passes the response to the authentication server.
- 17.** The authentication server responds with a success message indicating the supplicant and the authentication server have successfully authenticated each other.
- 18.** The authenticator passes the message to the supplicant.

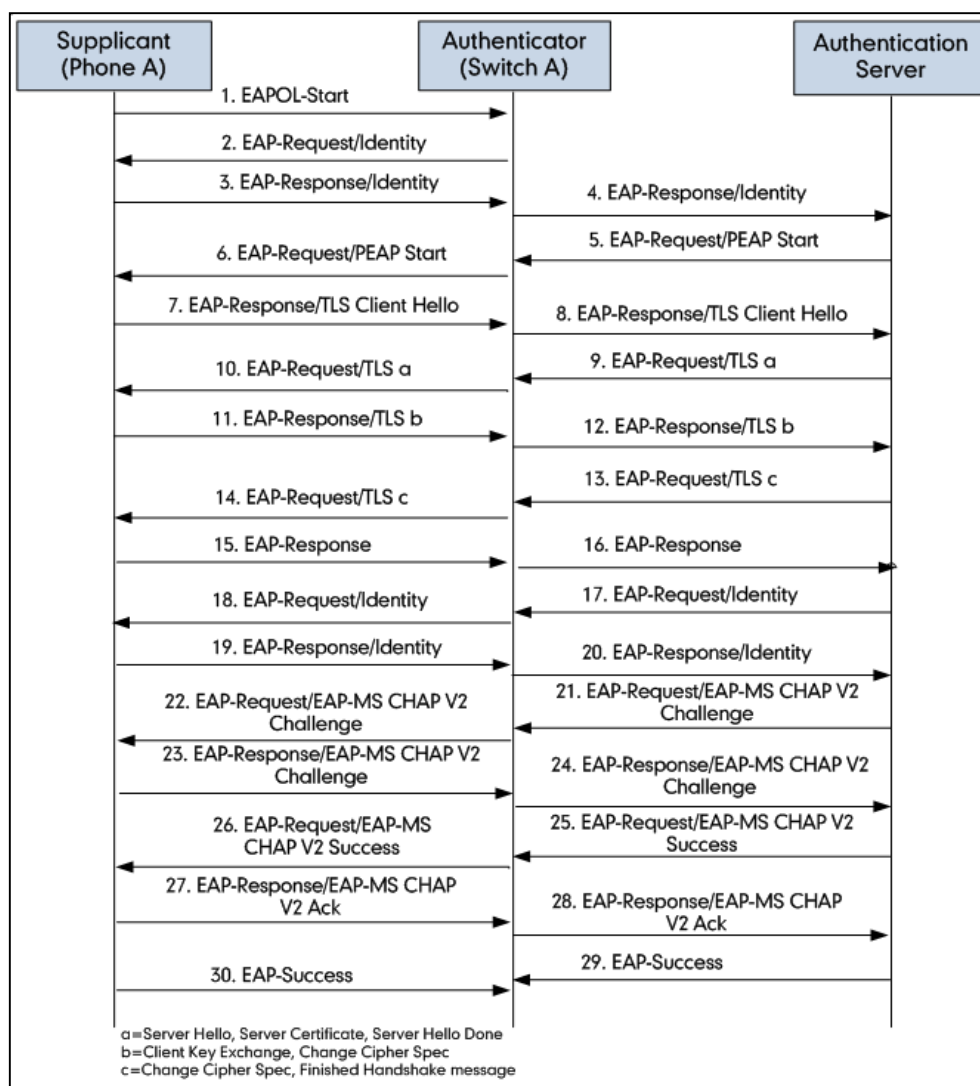
After the supplicant is authenticated successfully, the authenticator provides network access permissions. If the supplicant does not provide proper identification, the authentication server responds with a rejection message. The authenticator passes the message to the supplicant and blocks access to the LAN.

If the supplicant is disabled or reset after successful authentication, the supplicant sends an EAPOL-Logoff message, which prompts the authenticator to block access to the LAN.

A Successful Authentication Using EAP-PEAP/MSCHAPv2 Protocol

Protocol

The following figure illustrates the scenario of a successful 802.1X authentication process using the EAP-PEAP/MSCHAPv2 protocol.



1. The supplicant sends an "EAPOL-Start" packet to the authenticator.
2. The authenticator responds with an "EAP-Request/Identity" packet to the supplicant.
3. The supplicant responds with an "EAP-Response/Identity" packet to the authenticator.
4. The authenticator strips the Ethernet header and encapsulates the remaining EAP frame in the RADIUS format, and then sends it to the authentication server.
5. The authentication server recognizes the packet as a PEAP type and sends an "EAP-Request" packet with a PEAP start message to the authenticator.
6. The authenticator strips the authentication server's frame header, encapsulates the remaining EAP frame in the EAPOL format, and then sends it to the supplicant.

7. The supplicant responds with an "EAP-Respond" packet containing a TLS client hello handshake message to the authenticator. The TLS client hello message includes TLS version supported by the supplicant, a session ID, a random number and a set of cipher suites.
8. The authenticator passes the respond to the authentication server.
9. The authentication server sends an "EAP-Request" packet to the authenticator. The packet includes a TLS server hello handshake message, a server certificate message and a server hello done message.
10. The authenticator passes the request to the supplicant.
11. The supplicant responds with an "EAP-Response" packet to the authenticator. The packet includes a TLS change cipher spec message and a certificate verify message.
12. The authenticator passes the response to the authentication server.
13. The authentication server sends an "EAP-Request" packet to the authenticator. The packet includes a TLS change cipher spec message and a finished handshake message. The change cipher spec message is sent to notify the authenticator that subsequent records will be protected under the newly negotiated cipher spec.
14. The authenticator passes the request to the supplicant.
15. The supplicant responds with an "EAP-Response" packet to the authenticator.
16. The authenticator passes the response to the authentication server. The TLS tunnel is established.
17. The authentication server sends an "EAP-Request/Identity" packet to the authenticator.
18. The authenticator passes the request to the supplicant.
19. The supplicant responds with an "EAP-Response/Identity" packet to the authenticator.
20. The authenticator passes the response to the authentication server.
21. The authentication server sends an "EAP-Request" packet to the authenticator. The packet includes an MSCHAPv2 challenge message.
22. The authenticator passes the request to the supplicant.
23. The supplicant responds a challenge message to the authenticator.
24. The authenticator passes the message to the authentication server.
25. The authentication server sends a success message indicating that the supplicant provides proper identity.
26. The authenticator passes the message to the supplicant.
27. The supplicant responds with an ACK message to the authenticator.
28. The authenticator passes the respond message to the authentication server.
29. The authentication server sends a successful message to the authenticator.
30. The authenticator passes the message to the supplicant.

After the supplicant is authenticated successfully, the authenticator provides network access permissions. If the supplicant does not provide proper identification, the authentication server responds with a rejection message. The authenticator passes the message to the supplicant and blocks access to the LAN.

If the supplicant is disabled or reset after successful authentication, the supplicant sends an EAPOL-Logoff message, which prompts the authenticator to block access to the LAN.

A Successful Authentication Using EAP-TTLS/EAP-MSCHAPv2 Protocol

The 802.1X authentication process using the EAP-TTLS/EAP-MSCHAPv2 protocol is quite similar to that using the EAP-PEAP/MSCHAPv2 protocol. For more information, refer to the network resource.

A Successful Authentication Using EAP-PEAP/GTC Protocol

The 802.1X authentication process using the EAP-PEAP/GTC protocol is quite similar to that using the EAP-PEAP/MSCHAPv2 protocol. For more information, refer to the network resource.

A Successful Authentication Using EAP-TTLS/EAP-GTC Protocol

The 802.1X authentication process using the EAP-TTLS/EAP-GTC protocol is quite similar to that using the EAP-PEAP/MSCHAPv2 protocol. For more information, refer to the network resource.

A Successful Authentication Using EAP-FAST Protocol

The 802.1X authentication process using the EAP-FAST protocol is quite similar to that using the EAP-PEAP/MSCHAPv2 protocol. For more information, refer to the network resource.

Customer Feedback

We are striving to improve our documentation quality and we appreciate your feedback. Email your opinions and comments to DocsFeedback@yealink.com.