



# 802.1X 环境搭建 及话机配置使用说明

版本：〈2.0〉

发布日期：〈2018-7-30〉



# 目录

---

- 1 修订历史.....1
- 2 802.1X 介绍.....2
  - 2.1 概述.....2
  - 2.2 802.1x 认证体系及流程.....2
- 3 802.1x 环境搭建.....4
  - 3.1 服务器配置.....4
  - 3.2 修改配置文件.....7
  - 3.3 交换机配置（以深圳锐捷交换机 RG-S2312-P 为例）.....9
  - 3.4 802.1x 的三种认证配置.....11
  - 3.5 证书制作.....11
- 4 设备端配置.....13
  - 4.1 EAP-MD5 认证.....13
  - 4.2 EAP-TLS 认证.....13
  - 4.3 PEAP-mschapv2 认证.....14
- 5 认证过程抓包.....15
  - 5.1 服务器端抓包.....15
  - 5.2 设备端抓包.....15
- 6 可能遇到的问题解决方法.....17

# 1 修订历史

---

修订历史:

版本	作者	发布时间	说明
1.0	<刘蕾>	<2014-8-7>	<初始版本>
2.0	<宋蒙蒙>	<2018-05-22>	<更新认证抓包示例，添加设备端配置截图，添加交换机配置说明，更新文档格式和部分技术相关介绍>

## 2 802.1X 介绍

### 2.1 概述

802.1X 协议起源于 802.11 协议，后者是 IEEE 的无线局域网协议，制订 802.1X 协议的初衷是为了解决无线局域网用户的接入认证问题。IEEE802LAN 协议定义的局域网并不提供接入认证，只要用户能接入局域网控制设备(如 LANS witch)，就可以访问局域网中的设备或资源。这在早期企业网有线 LAN 应用环境下并不存在明显的安全隐患。

随着移动办公及驻地网运营等应用的大规模发展，服务提供者需要对用户的接入进行控制和配置。尤其是 WLAN 的应用和 LAN 接入在电信网上大规模开展，有必要对端口加以控制 以实现用户级的接入控制，802.1X 就是 IEEE 为了解决基于端口的接入控制(Port-Based Network Access Control)而定义的一个标准。

### 2.2 802.1x 认证体系及流程

802.1x 是根据用户 ID 或设备，对网络客户端(或端口)进行鉴权的标准。该流程被称为“端口级别的鉴权”。它采用 RADIUS(远程认证拨号用户服务)方法，并将其划分为三个不同小组:请求方、认证方和授权服务器。

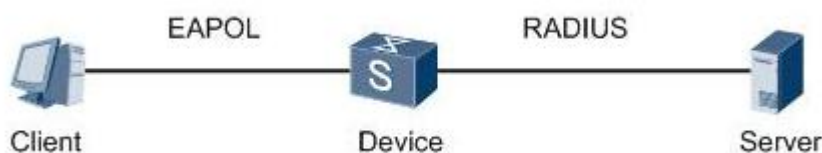


图 1 802.1X 认证的体系结构

802.1x 标准应用于试图连接到端口或其它设备(如 Cisco Catalyst 交换机或 Cisco Aironet 系列接入点)(认证方)的终端设备和用户(请求方)。认证和授权都通过鉴权服务器(如 Cisco Secure ACS)后端通信实现。IEEE 802.1x 提供自动用户身份识别，集中进行鉴权、密钥管理和 LAN 连接配置。整个 802.1x 的实现设计三个部分，请求者系统、认证系统和认证服务器系统。

认证过程：

- (1) 客户端向接入设备发送一个 EAPoL-Start 报文，开始 802.1x 认证接入；
- (2) 接入设备向客户端发送 EAP-Request/Identity 报文，要求客户端将用户名送上来；
- (3) 客户端回应一个 EAP-Response/Identity 给接入设备的请求，其中包括用户名；
- (4) 接入设备将 EAP-Response/Identity 报文封装到 RADIUS Access-Request 报文中，发送给认证服务器；
- (5) 认证服务器产生一个 Challenge，通过接入设备将 RADIUS Access-Challenge 报

文发送给客户端，其中包含有 EAP-Request/MD5-Challenge；

(6) 接入设备通过 EAP-Request/MD5-Challenge 发送给客户端，要求客户端进行认证

(7) 客户端收到 EAP-Request/MD5-Challenge 报文后，将密码和 Challenge 做 MD5 算法后的 Challenged-Pass-word，在 EAP-Response/MD5-Challenge 回应给接入设备

(8) 接入设备将 Challenge, Challenged Password 和用户名一起送到 RADIUS 服务器，由 RADIUS 服务器进行认证

(9) RADIUS 服务器根据用户信息，做 MD5 算法，判断用户是否合法，然后回应认证成功/失败报文到接入设备。如果成功，携带协商参数，以及用户的相关业务属性给用户授权。如果认证失败，则流程到此结束；

(10) 如果认证通过，用户通过标准的 DHCP 协议（可以是 DHCP Relay），通过接入设备获取规划的 IP 地址；

(11) 如果认证通过，接入设备发起计费开始请求给 RADIUS 用户认证服务器；

(12) RADIUS 用户认证服务器回应计费开始请求报文。用户上线完毕。

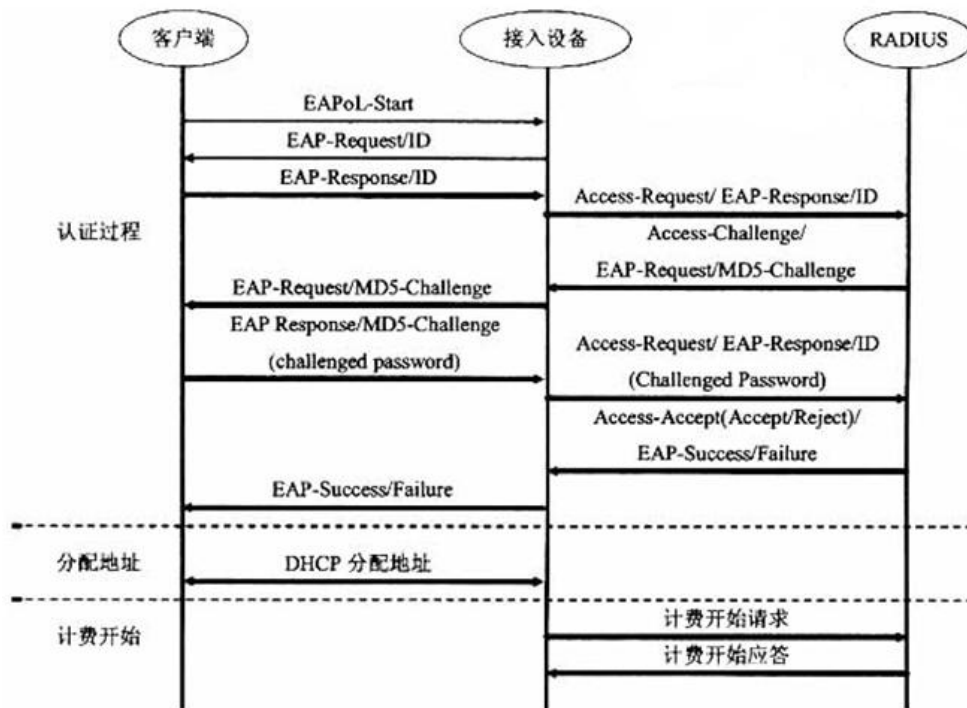


图 2 基于 EAP-MD5 的 802.1X 认证流程

## 3 802.1x 环境搭建

### 3.1 服务器配置

#### 1. 安装 FreeRadius

软件下载路径: [\\172.16.1.9\share\Testing\\_department\software](http://172.16.1.9/share/Testing_department/software)

安装 FreeRADIUS-server-2.2.0-x86.rar (此版本就是将 linux 的 freeRadius 编译成 windows 版本了), 或是去官网下载。这里以 FreeRADIUS-server-2.2.0-x86.rar 为例。

解压 FreeRADIUS-server-2.2.0-x86.rar, 双击 FreeRADIUS-server-2.2.0-x86.exe, 安装程序, 这里按默认路径安装, 如图:



图 3



图 4

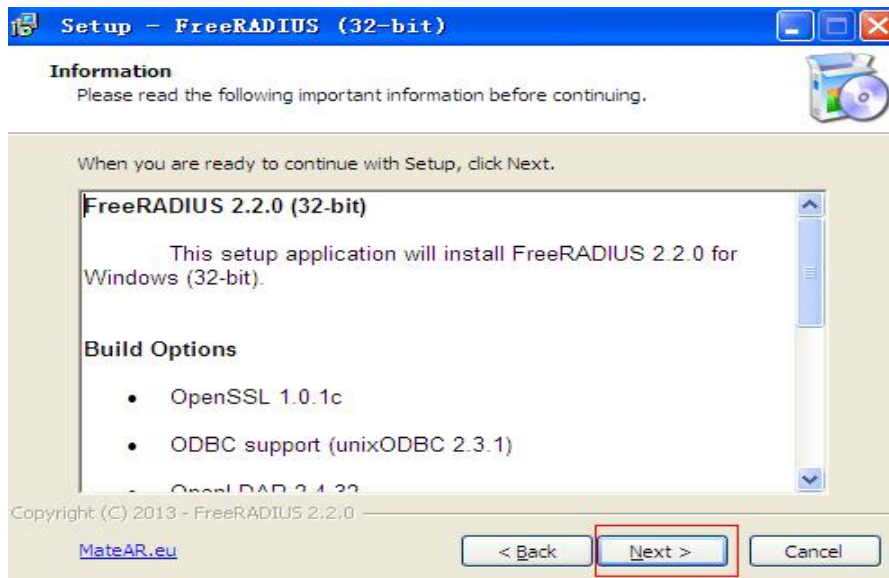


图 5



图 6

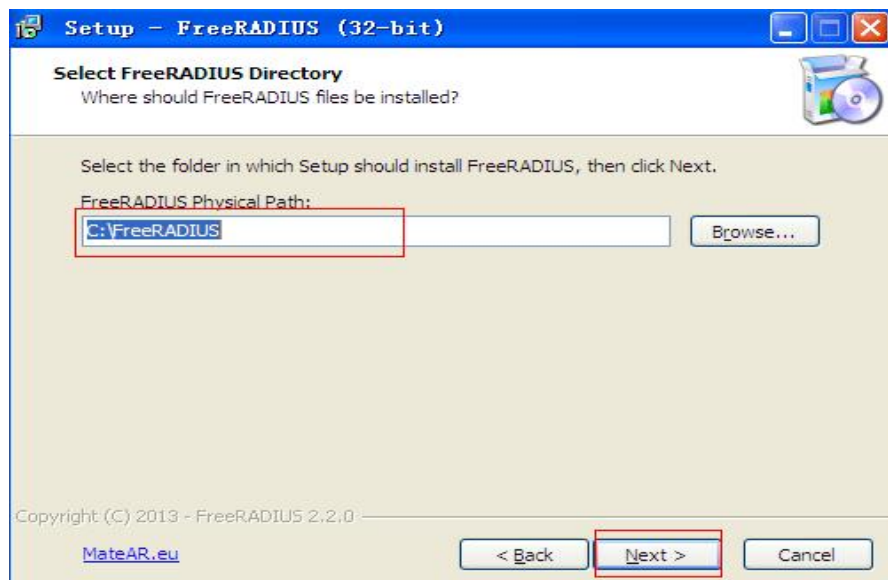


图 7

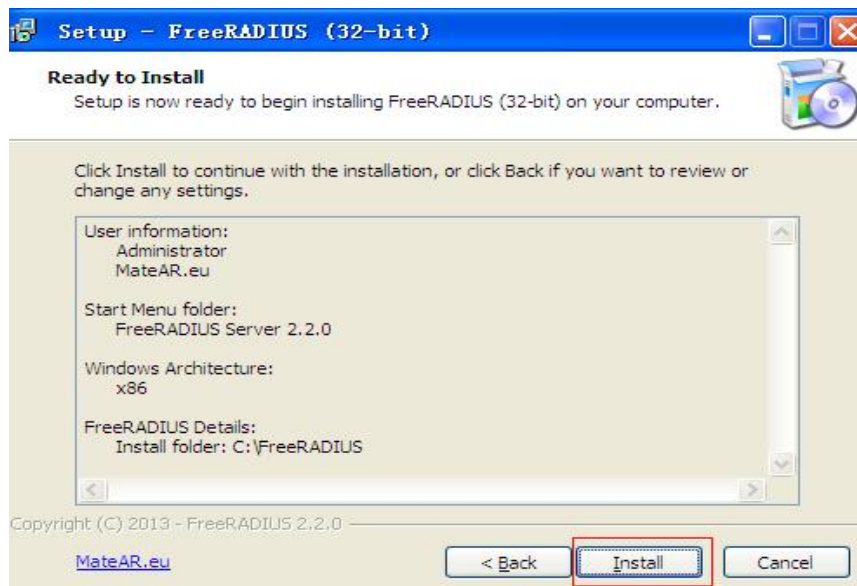


图 8





图 9

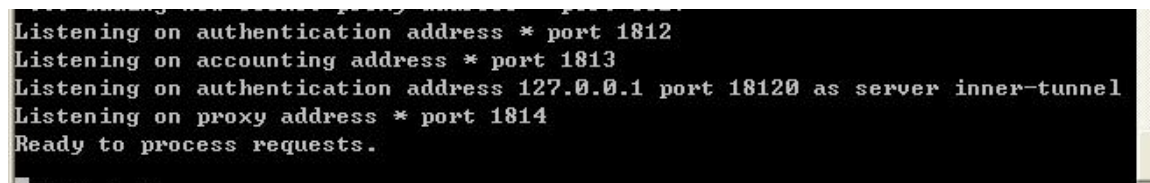
## 2. 测试软件

点击开始-所有程序-FreeRADIUS Server 2.2.0-Start RADIUS Server，如图：



图 10

成功后，显示如下信息：



## 3.2 修改配置文件

1. 安装完成后，进入 C:\FreeRADIUS\etc\raddb 目录，修改 clients.conf：

```
client_server localhost {
    ipaddr = 127.0.0.1           #127.0.0.1 是服务器保留测试地址
    port = 1812                 #服务器默认的认证端口
    type = "auth"               #认证类型为 auth
    secret = "testing123"       #共享密钥
```

```

response_window = 20                                #响应端口
max_outstanding = 65536                             #
require_message_authenticator = yes                 #是否进行消息认证
zombie_period = 40                                  #
status_check = "status-server"                     #服务器状态检查
ping_interval = 30                                  #
check_interval = 30                                 #检查时间间隔
num_answers_to_alive = 3                            #
num_pings_to_alive = 3                              #
revive_interval = 120                               #恢复时间间隔
status_check_timeout = 4                            #状态检查超时时间

coa {
    irt = 2                                           #初始重传时间
    mrt = 16                                         #最大重传时间
    mrc = 5                                           #最大重传次数
    mrd = 30                                         #最大重传持续时间
}
}

# 这里的 client 是指交换机
client 10.1.1.2/8 {                                #服务器地址/子网掩码为 255.0.0.0
    require_message_authenticator = yes             #是否认证信息
    secret = qq                                     # NAS 与 radius 间的通信密码 key
    shortname = ruijie                             # 域名，可以随便写，这里的 ruijie 是我们要认证
                                                    交换机的型号
}

```

2. 进入 C:\FreeRADIUS\etc\raddb 目录，打开 users，在里面设置用户名、密码（也可以使用默认的），如图：

```

qq Cleartext-Password := "qq"
    Reply-Message = "Hello, %{User-Name}"

iketestuser EAP-IKEv2-IDType := KEY_ID, EAP-IKEv2-Secret := "qq"

bob Digest-HA1 := "12af60467a33e8518da5c68bbff12b11"

```

所以设备的 802.1x 认证的用户名：qq，密码是：qq。相应的服务器端也要配置为此用户名及密码。

注：如果话机认证不能通过，将 testing Cleartext-Password := “testing” 加在第一行，用此用户名密码进行认证。

3. 运行服务器，看是否正常，正常会显示如下信息

```

Module: Checking session <...> for more modules to load
Module: Checking post-proxy <...> for more modules to load
Module: Checking post-auth <...> for more modules to load
Module: Instantiating module "attr_filter.access_reject" from file ../etc/raddb
/modules/attr_filter
  attr_filter attr_filter.access_reject <
    attrsfile = "../etc/raddb/attrs.access_reject"
    key = "%{User-Name}"
    relaxed = no
  >
reading pairlist file ../etc/raddb/attrs.access_reject
> # modules
> # server
server inner-tunnel < # from file ../etc/raddb/sites-enabled/inner-tunnel
modules <
Module: Checking authenticate <...> for more modules to load
Module: Checking authorize <...> for more modules to load
Module: Checking session <...> for more modules to load
Module: Checking post-proxy <...> for more modules to load
Module: Checking post-auth <...> for more modules to load
> # modules
> # server
radiusd: ##### Opening IP addresses and Ports #####
listen <
  type = "auth"
  ipaddr = *
  port = 0
>
listen <
  type = "acct"
  ipaddr = *
  port = 0
>
listen <
  type = "auth"
  ipaddr = 127.0.0.1
  port = 18120
>
... adding new socket proxy address * port 1452
Listening on authentication address * port 1812
Listening on accounting address * port 1813
Listening on authentication address 127.0.0.1 port 18120 as server inner-tunnel
Listening on proxy address * port 1814
Ready to process requests.

```

图 11

## 3.3 交换机配置

### 3.3.1 以深圳锐捷交换机 RG-S2312-P 为例

1. 认证服务器必须能与 pc 互通，目前没有找到修改交换机 IP 的方法，所以只能修改自己 pc 网段和交换机 IP 在同一网段即可。交换机 IP 地址：10.1.1.1./8
2. 进入交换机 web 配置界面（http://10.1.1.1），配置 802.1x 认证。也可以使用命令来配置（具体参考交换机的命令配置文档）。
3. 配置交换机 802.1x 功能，注意交换机与认证服务器通信密钥，认证用户名/密码必须配置正确。通信密钥、认证名、密码等在本文 3.2 中已作说明。

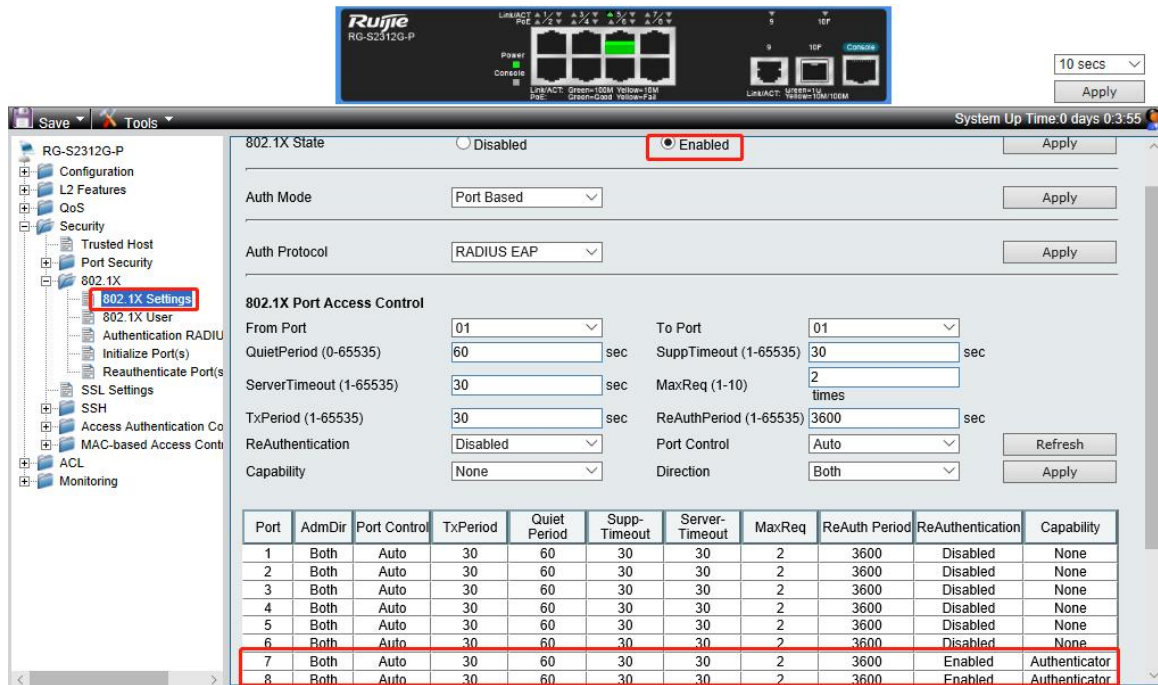


图 12

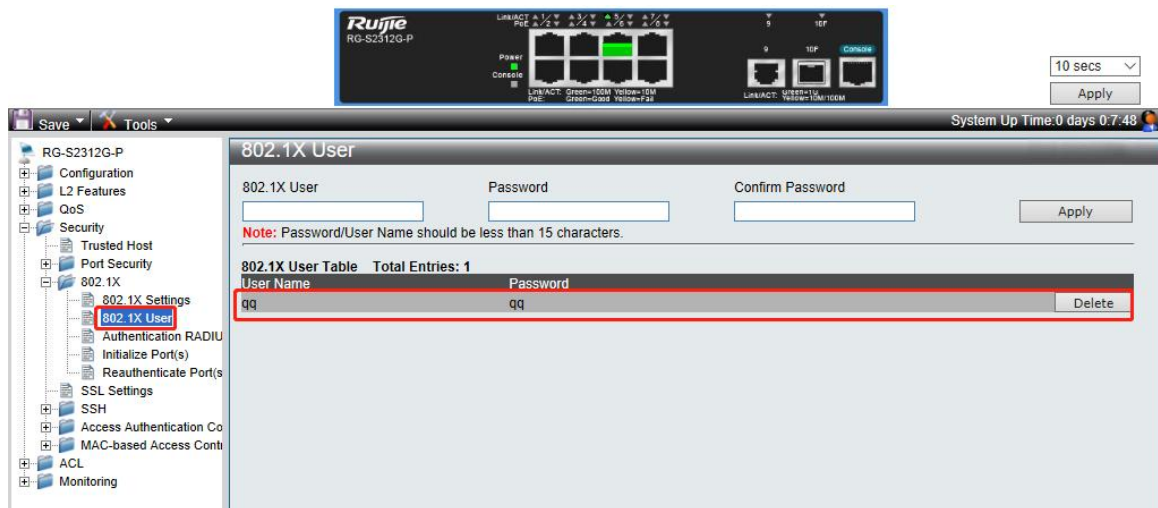


图 13



图 14

4. 认证服务器连接在绿色标示对应的对口，认证客户端连接在红线标识的端口即可。

Port	AdmDir	Port Control	TxPeriod	Quiet Period	Supp-Timeout	Server-Timeout	MaxReq	ReAuth Period	ReAuthentication	Capability
1	Both	Auto	30	60	30	30	2	3600	Enabled	Authenticator
2	Both	Auto	30	60	30	30	2	3600	Enabled	Authenticator
3	Both	Auto	30	60	30	30	2	3600	Enabled	Authenticator
4	Both	Auto	30	60	30	30	2	3600	Enabled	None
5	Both	Auto	30	60	30	30	2	3600	Disabled	None
6	Both	Auto	30	60	30	30	2	3600	Disabled	None
7	Both	Auto	30	60	30	30	2	3600	Disabled	None
8	Both	Auto	30	60	30	30	2	3600	Disabled	None
9	Both	Auto	30	60	30	30	2	3600	Disabled	None
10	Both	Auto	30	60	30	30	2	3600	Disabled	None

图 15

注：认证服务器端抓包过滤 EAP 看到服务器与认证交换机之间的数据往来，所以要使用借助 hub 来抓取设备认证过程的数据包。

### 3.3.2 以北京思科 2560 交换机为例（新添加）

PC 通过 SecureCRT 连接 console 口 目前设置 24 口为验证口

Dot1x (802.1x) configuration method for switches.

Command	Purpose
<b>configure terminal</b>	进入全局配置模式 Enter global configuration mode

	<b>aaa new-mode</b>	<p>启用 AAA.</p> <p>Enable AAA.</p>
	<b>aaa authentication dot1x default group radius</b>	<p>创建一个缺省 IEEE 802.1x 认证方法列表</p> <p>Create a list of default IEEE 802.1x authentication methods</p>
	<b>dot1x system-auth-control</b>	<p>启用 IEEE 802.1x 认证的全局配置</p> <p>Enable global configuration for IEEE 802.1x certification</p>
	<b>aaa authorization network{default} group radius</b>	<p>(可选) 启用 VLAN 分配特性时需要此项配置</p> <p>This configuration is required when enabling the VLAN allocation feature</p>
	<b>radius-server host <i>ip-address</i> auth-port 1812</b>	<p>(可选) 指定 radius 认证服务器的地址</p> <p>默认 udp 认证端口是 1812，范围 0~65536.</p> <p>Optionally specify the address of the radius authentication server,</p> <p>The default udp authentication port is 1812, with a range of 0~</p>

		65536.
	<b>radius-server key string</b>	(可选)指定交换机与认证服务器通讯所需的密钥 Optionally, specify the key required for the switch to communicate with the authentication server
	<b>Interface interface-id</b>	进入需要启用 802.1x 认证的端口 Enter the port where 802.1x authentication is required
	<b>switchport mode access</b>	(可选) 设置端口的访问模式（如果 step6、7 已配置了 radius 服务器） (optional) set the access mode of the port (if step6, 7 has configured the radius server)
	<b>authentication port-control auto</b> or <b>dot1x port-control auto</b>	启用此端口的 IEEE 802.1x 认证 Enable IEEE 802.1x certification for this port
	<b>dot1x host-mode multi-host</b>	host-mode 是针对在端口下通过 hub 有多台机器上网的问题设置的。默认的 single-host 只允许一台机器能够使用该端口 Host-mode is set for problems where multiple machines are connected to the port through a hub



		ected via the hub under the port. The default single-host is only allowed a machine can use this port
	<b>dot1x max-rea uth-req count</b>	<p>（可选）设置此端口在重启认证过程之前向客户端发送 EAP-request/identity 帧的次数，范围是 1~10，默认是 2；建议为 10。</p> <p>(Optional) set the number of times this port sends eap-request /identity frames to the client before restarting the authentication process in a range of 1 to 10, with the default of 2. The recommended number is 10</p>
	<b>end</b>	<p>返回特权模式</p> <p>Return privilege mode</p>
	<b>show authenti cation or show dot1x</b>	<p>验证你的 802.1x 配置.</p> <p>Verify your 802.1x configuration.</p>
	<b>copy running- config startup- config</b>	<p>(可选) 保存配置。建议在完全确定你的配置的情况下再保存你的配置.</p> <p>(Optional) save the configuration.</p> <p>It is recommended that you save your configuration after you are completely sure of it.</p>



## 3.4 802.1x 的三种认证配置

### 1. EAP-MD5 认证

打开 C:\FreeRADIUS\etc\raddb\eap.conf

确认 eap {

```
    default_eap_type = md5
```

(一般默认为 md5)

### 2. EAP-TLS 认证

将 default\_eap\_type = md5 改为 default\_eap\_type = tls

### 3. PEAP-mschapv2 认证

将 default\_eap\_type = md5 改为 default\_eap\_type = peap

## 3.5 证书制作

参考 openvpn 文档，将 openvpn 制作的 4 个证书做成 802.1x 的认证证书：

1. client.pem: 将 client.key 中的全部内容 copy 到 client.crt 文件的最后
2. RootCA.pem: 将 ca.crt 改名为 RootCA.pem
3. server.pem: 将 server.crt 改名为 server.pem
4. server-key.pem: 将 server.key 改名为 server-key.pem

制作的 4 个证书放在 FreeRADIUS\etc\raddb\certs 下

## 4 设备端配置

本文以 X6 为准。

### 4.1 EAP-MD5 认证



图 16

### 4.2 EAP-TLS 认证



图 17

### 4.3 PEAP-mschapv2 认证



图 18

注意：

认证需要上传的两个证书，请按照如下所示格式来命名证书文件。

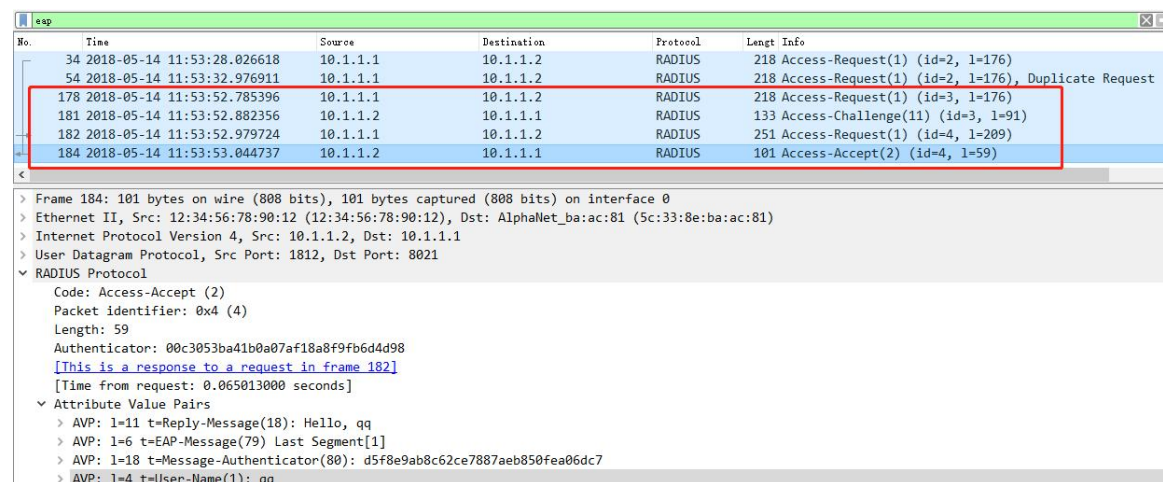
	client.pem-DOT1X_CLIENT-	2016/9/23 14:31	PEM-DOT1X_CLIENT...	5 KB
	RootCA.pem-DOT1X_CA-	2016/9/23 14:30	PEM-DOT1X_CA- 文件	2 KB

注：1. PEAP-mschapv2 认证只需要上传 RootCA.pem (RootCA.pem-DOT1X\_CA-) 这一个证书即可，tls 认证需要同时上传 RootCA.pem-DOT1X\_CA-和 client.pem-DOT1X\_CLIENT-这 2 个证书。

2. 测 802.1x 时，话机 ip 为静态，需要用到证书的，话机时间要确保在证书使用范围内，时间最好设置为 20140801. (date -s "2014-08-01 16:18") )

## 5 认证过程抓包

### 5.1 服务器端抓包



No.	Time	Source	Destination	Protocol	Length	Info
34	2018-05-14 11:53:28.026618	10.1.1.1	10.1.1.2	RADIUS	218	Access-Request(1) (id=2, l=176)
54	2018-05-14 11:53:32.976911	10.1.1.1	10.1.1.2	RADIUS	218	Access-Request(1) (id=2, l=176), Duplicate Request
178	2018-05-14 11:53:52.785396	10.1.1.1	10.1.1.2	RADIUS	218	Access-Request(1) (id=3, l=176)
181	2018-05-14 11:53:52.882356	10.1.1.2	10.1.1.1	RADIUS	133	Access-Challenge(11) (id=3, l=91)
182	2018-05-14 11:53:52.979724	10.1.1.1	10.1.1.2	RADIUS	251	Access-Request(1) (id=4, l=209)
184	2018-05-14 11:53:53.044737	10.1.1.2	10.1.1.1	RADIUS	101	Access-Accept(2) (id=4, l=59)

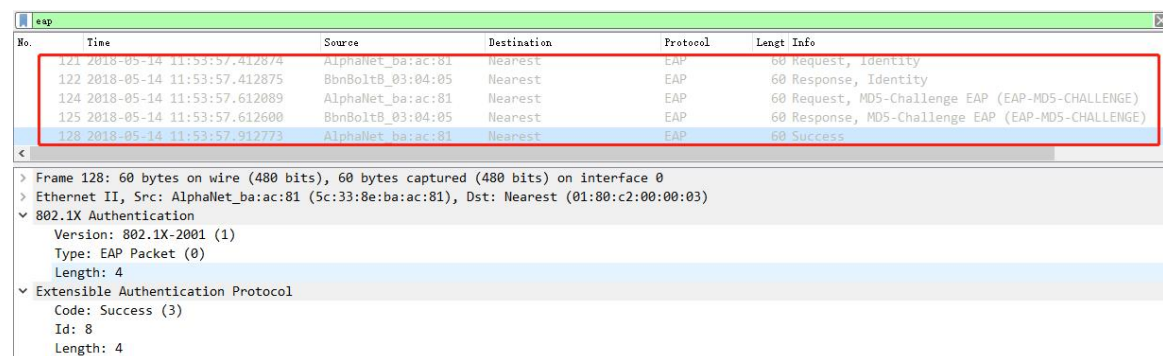
Frame 184: 101 bytes on wire (808 bits), 101 bytes captured (808 bits) on interface 0  
Ethernet II, Src: 12:34:56:78:90:12 (12:34:56:78:90:12), Dst: AlphaNet\_ba:ac:81 (5c:33:8e:ba:ac:81)  
Internet Protocol Version 4, Src: 10.1.1.2, Dst: 10.1.1.1  
User Datagram Protocol, Src Port: 1812, Dst Port: 8021  
RADIUS Protocol  
Code: Access-Accept (2)  
Packet identifier: 0x4 (4)  
Length: 59  
Authenticator: 00c3053ba41b0a07af18a8f9fb6d4d98  
[This is a response to a request in frame 182]  
[Time from request: 0.065013000 seconds]  
Attribute Value Pairs  
AVP: l=11 t=Reply-Message(18): Hello, qq  
AVP: l=6 t=EAP-Message(79) Last Segment[1]  
AVP: l=18 t=Message-Authenticator(80): d5f8e9ab8c62ce7887aeb850fea06dc7  
AVP: l=4 t=User-Name(1): qq

图 19

可以看到 user-name(1): qq 字段。

### 5.2 设备端抓包

#### 1. EAP-MD5 认证



No.	Time	Source	Destination	Protocol	Length	Info
121	2018-05-14 11:53:57.412874	AlphaNet_ba:ac:81	Nearest	EAP	60	Request, Identity
122	2018-05-14 11:53:57.412875	BbnBolt8_03:04:05	Nearest	EAP	60	Response, Identity
124	2018-05-14 11:53:57.612089	AlphaNet_ba:ac:81	Nearest	EAP	60	Request, MD5-Challenge EAP (EAP-MD5-CHALLENGE)
125	2018-05-14 11:53:57.612600	BbnBolt8_03:04:05	Nearest	EAP	60	Response, MD5-Challenge EAP (EAP-MD5-CHALLENGE)
128	2018-05-14 11:53:57.912773	AlphaNet_ba:ac:81	Nearest	EAP	60	Success

Frame 128: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0  
Ethernet II, Src: AlphaNet\_ba:ac:81 (5c:33:8e:ba:ac:81), Dst: Nearest (01:80:c2:00:00:03)  
802.1X Authentication  
Version: 802.1X-2001 (1)  
Type: EAP Packet (0)  
Length: 4  
Extensible Authentication Protocol  
Code: Success (3)  
Id: 8  
Length: 4

图 20

#### 2. EAP-TLS 认证

No.	Time	Source	Destination	Protocol	Length	Info
13228	2018-05-14...	AlphaNet_b...	Nearest	EAP		60 Request, Identity
13229	2018-05-14...	BbnBolt8_0...	Nearest	EAP		60 Response, Identity
13230	2018-05-14...	AlphaNet_b...	Nearest	EAP		60 Request, TLS EAP (EAP-TLS)
13231	2018-05-14...	BbnBolt8_0...	Nearest	TLSv1		120 Client Hello
13232	2018-05-14...	AlphaNet_b...	Nearest	TLSv1		1042 Server Hello, Certificate, Server Key Exchange, Certificate Request, Server Hello Done
13233	2018-05-14...	BbnBolt8_0...	Nearest	EAP		60 Response, TLS EAP (EAP-TLS)
13236	2018-05-14...	AlphaNet_b...	Nearest	TLSv1		1042 Server Hello, Certificate, Server Key Exchange, Certificate Request, Server Hello Done
13237	2018-05-14...	BbnBolt8_0...	Nearest	EAP		60 Response, TLS EAP (EAP-TLS)
13239	2018-05-14...	AlphaNet_b...	Nearest	TLSv1		617 Server Hello, Certificate, Server Key Exchange, Certificate Request, Server Hello Done
13240	2018-05-14...	BbnBolt8_0...	Nearest	TLSv1		1426 Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake...
13241	2018-05-14...	AlphaNet_b...	Nearest	EAP		60 Request, TLS EAP (EAP-TLS)
13242	2018-05-14...	BbnBolt8_0...	Nearest	TLSv1		924 Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake...
13243	2018-05-14...	AlphaNet_b...	Nearest	TLSv1		87 Change Cipher Spec, Encrypted Handshake Message
13244	2018-05-14...	BbnBolt8_0...	Nearest	EAP		60 Response, TLS EAP (EAP-TLS)
13246	2018-05-14...	AlphaNet_b...	Nearest	EAP		60 Success

> Frame 13228: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0

> Ethernet II, Src: AlphaNet\_ba:ac:81 (5c:33:8e:ba:ac:81), Dst: Nearest (01:80:c2:00:00:03)

> 802.1X Authentication

Extensible Authentication Protocol

Code: Request (1)

Id: 1

Length: 15

Type: Identity (1)

Identity: User name:

图 21

### 3. PEAP-mschapv2 认证

No.	Time	Source	Destination	Protocol	Length	Info
2999	2018-05-14 14:08:37.977783	AlphaNet_ba:ac:81	Nearest	EAP		60 Request, Identity
3000	2018-05-14 14:08:37.978464	BbnBolt8_03:04:05	Nearest	EAP		60 Response, Identity
3002	2018-05-14 14:08:38.137068	AlphaNet_ba:ac:81	Nearest	EAP		60 Request, Protected EAP (EAP-PEAP)
3004	2018-05-14 14:08:38.146463	BbnBolt8_03:04:05	Nearest	TLSv1		120 Client Hello
3005	2018-05-14 14:08:38.437874	AlphaNet_ba:ac:81	Nearest	TLSv1		1042 Server Hello, Certificate, Server Key Exchange, Server Hello Done
3006	2018-05-14 14:08:38.438558	BbnBolt8_03:04:05	Nearest	EAP		60 Response, Protected EAP (EAP-PEAP)
3007	2018-05-14 14:08:38.742164	AlphaNet_ba:ac:81	Nearest	TLSv1		1038 Server Hello, Certificate, Server Key Exchange, Server Hello Done
3008	2018-05-14 14:08:38.742818	BbnBolt8_03:04:05	Nearest	EAP		60 Response, Protected EAP (EAP-PEAP)
3009	2018-05-14 14:08:38.967748	AlphaNet_ba:ac:81	Nearest	TLSv1		448 Server Hello, Certificate, Server Key Exchange, Server Hello Done
3011	2018-05-14 14:08:39.239028	BbnBolt8_03:04:05	Nearest	TLSv1		222 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
3012	2018-05-14 14:08:39.639538	AlphaNet_ba:ac:81	Nearest	TLSv1		83 Change Cipher Spec, Encrypted Handshake Message
3013	2018-05-14 14:08:39.641606	BbnBolt8_03:04:05	Nearest	EAP		60 Response, Protected EAP (EAP-PEAP)
3014	2018-05-14 14:08:39.937022	AlphaNet_ba:ac:81	Nearest	TLSv1		61 Application Data
3015	2018-05-14 14:08:39.938376	BbnBolt8_03:04:05	Nearest	TLSv1		98 Application Data, Application Data
3017	2018-05-14 14:08:40.237092	AlphaNet_ba:ac:81	Nearest	TLSv1		61 Application Data
3018	2018-05-14 14:08:40.238274	BbnBolt8_03:04:05	Nearest	TLSv1		98 Application Data, Application Data
3019	2018-05-14 14:08:40.537281	AlphaNet_ba:ac:81	Nearest	TLSv1		77 Application Data
3022	2018-05-14 14:08:40.551899	BbnBolt8_03:04:05	Nearest	TLSv1		146 Application Data, Application Data
3026	2018-05-14 14:08:40.837108	AlphaNet_ba:ac:81	Nearest	TLSv1		109 Application Data
3027	2018-05-14 14:08:40.838514	BbnBolt8_03:04:05	Nearest	TLSv1		98 Application Data, Application Data
3028	2018-05-14 14:08:41.137035	AlphaNet_ba:ac:81	Nearest	TLSv1		61 Application Data
3029	2018-05-14 14:08:41.137912	BbnBolt8_03:04:05	Nearest	TLSv1		98 Application Data, Application Data
3033	2018-05-14 14:08:41.437214	AlphaNet_ba:ac:81	Nearest	EAP		60 Success

图 22

## 6 可能遇到的问题解决方法

1.

```
../etc/raddb/clients.conf[176]: Failed to look up hostname ::1: ip_hton: 不知道这样的主机。  
C:\FreeRADIUS\sbin>
```

找到对应目录下的 clients.conf 文件，将下面几行屏蔽

```
175 # IPv6 Client  
176 #client ::1 {  
177 # secret = testing123  
178 # shortname = localhost  
179 #}  
180
```

2.

```
port = 0  
Failed opening authentication address 0:0:0:0:0:0:0 port 1812: Unknown error  
..\etc\raddb\radiusd.conf[312]: Error binding to port for 0:0:0:0:0:0:0 port 1812  
C:\FreeRADIUS\sbin>
```

找到对应目录下的 radiusd.conf 文件，将下面几行屏蔽

```
312 #listen {  
313 #     ipv6addr = ::  
314 # port = 0  
315 # type = auth  
316 #}
```

3.

```
Failed opening accounting address 0:0:0:0:0:0:0 port 1813: Unknown error  
..\etc\raddb\radiusd.conf[330]: Error binding to port for 0:0:0:0:0:0:0 port 1813  
C:\FreeRADIUS\sbin>
```

找到对应目录下的 radiusd.conf 文件，将下面几行屏蔽

```
330 #listen {  
331 #     ipv6addr = ::  
332 # port = 0  
333 # type = acct  
334 #}
```