802.1x 协议解析

缩略语

802.1X **本文指** IEEE 802.1X **标准**

RADIUS **远程用户拨入认证服务** (Remote Authentication Dial In User Service)

PAP **密码验证协议** (Password Authentication Protocol)

CHAP **质询握手验证协议** (Challenge Handshake Authentication Protocol)

EAP **扩展验证协议** (Extensible Authentication Protocol)

EAPOL **基于局域网的** EAP (EAP over LAN)

MD5 **消息摘要算法** 5 **版本** (Message-Digest Algorithm 5)

PAE **端口认证实体** (Port Authentication Entity)

一 802.1x 认证起源

802.1x **协议起源于** 802.11 **协议，后者是标准的无线局域网协议，**802.1x **协议的主要目的是为了解决无线局域网用户的接入认证问题。现在已经开始被应用于一般的有线** LAN **的接入（微软的** Windows XP**，以及** Cisco**，华为** 3com**，北电，港湾等厂商的设备已经开始支持** 802.1X **协议)。**

在 802.1x **出现之前，企业网上有线** LAN **应用都没有直接控制到端口的方法。也不需要控制到端口。但是随着无线** LAN **的应用以及** 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 Access Control**）而定义的一个标准。**

二 802.1x 认证的作用

802.1X **首先是一个认证协议，是一种对用户进行认证的方法和策略。**

802.1X **是基于端口的认证策略（这里的端口可以是一个实实在在的物理端口也可以是一个就像** VLAN **一样的逻辑端口，对于无线局域网来说这个“ 端口 ”就是一条信道）**

802.1X **的认证的最终目的就是确定一个端口是否可用。对于一个端口，如果认证成功那么就“ 打开 ”这个端口，允许所有的报文通过 ;如果认证不成功就使这个端口保持“ 关闭 ”，此时只允许** 802.1X **的认证报文** EAPOL（Extensible Authentication Protocol over LAN）**通过。**

三 802.1X 认证体系的结构

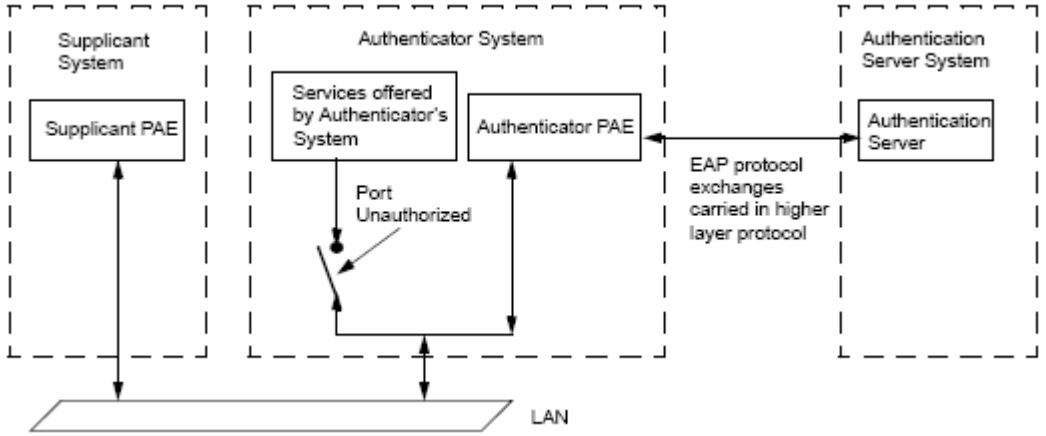

图 1 **客户端、认证系统、认证服务器所担任的角色**

802.1X 的认证体系分为三部分结构：

（1）Supplicant System，客户端(PC/网络设备)

（2）Authenticator System，认证系统

（3）Authentication Server System，认证服务器系统

Supplicant System，客户端(PC/网络设备)

Supplicant System—— Client（客户端）是一个需要接入 LAN 及享受 switch 提供服务的设备（如 PC 机），客户端需要支持 EAPOL 协议，客户端必须运行 802.1X 客户端软件，如：802.1X-complain，Microsoft Windows XP，H3C802.1X。

Authenticator System，认证系统

Authenticator System—— Switch （边缘交换机或无线接入设备）是一根据客户的认证状态控制物理接入的设备，switch 在客户和认证服务器间充当代理角色（proxy）。switch 与 client 间通过 EAPOL 协议进行通讯，switch 与认证服务器间通过 EAPoRadius 或 EAP 承载在其他高层协议上，以便穿越复杂的网络到达 Authentication Server （EAP Relay）；switch 要求客户端提供 identity，接收到后将 EAP 报文承载在 Radius 格式的报文中，再发送到认证服务器，返回等同； switch 根据认证结果控制端口是否可用；

需要指出的是：我们的 802.1x 协议在设备内终结并转换成标准的 RADIUS 协议报文，加密算法采用 PPP 的 CHAP 认证算法，所有支持 PPP CHAP 认证算法的认证计费服务器都可以与我们对接成功。

Authentication Sever System，认证服务器系统

Authentication server ——（认证服务器）对客户进行实际认证，认证服务器核实客户的 identity，通知 swtich 是否允许客户端访问 LAN 和交换机提供的服务。Authentication Sever 接受 Authenticator 传递过来的认证需求，认证完成后将认证结果下发给 Authenticator，完成对端口的管理。由于 EAP 协议较为灵活，除了 IEEE 802.1x 定义的端口状态外，Authentication Server 实际上也可以用于认证和下发更多用户相关的信息，如 VLAN、QOS、加密认证密钥、DHCP 响应等。

四 802.1x 的认证过程

802.1x 的认证中，端口的状态决定了客户端是否能接入网络，在启用 802.1x 认证时端口初始状态一般为非授权（unauthorized），在该状态下，除 802.1X 报文和广播报文外不允许任何业务输入、输出通讯。当客户通过认证后，则端口状态切换到授权状态（authorized），允许客户端通过端口进行正常通讯。

基于 802.1x 的认证系统在客户端和认证系统之间使用 EAPOL 格式封装 EAP 协议传送认证信息，认证系统与认证服务器之间通过 RADIUS 协议传送认证信息。由于 EAP 协议的可扩展性，基于 EAP 协议的认证系统可以使用多种不同的认证算法，如 EAP-MD5，EAP-TLS，EAP-SIM，EAP-TTLS 以及 EAP-AKA 等认证方法。

以 EAP—MD5 为例，描述 802.1x 的认证流程。EAP-MD5 是一种单向认证机制，可以完成网络对用户的认证，但认证过程不支持加密密钥的生成。基于 EAP—MD5 的 802.1x 认证系统功能实体协议栈如图 2 所示。基于 EAP-MD5 的 802.1x 认证流程如图 3 所示，认证流程包括以下步骤：

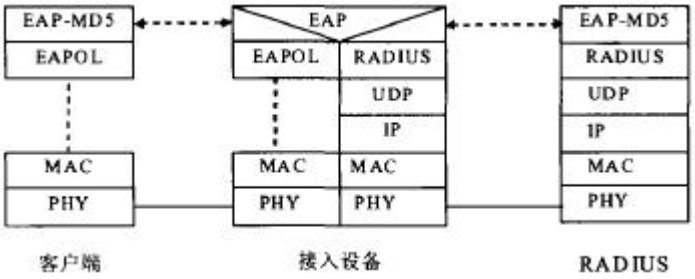

图 2 基于 EAP-MD5 的 802.1x 认证系统功能实体协议栈

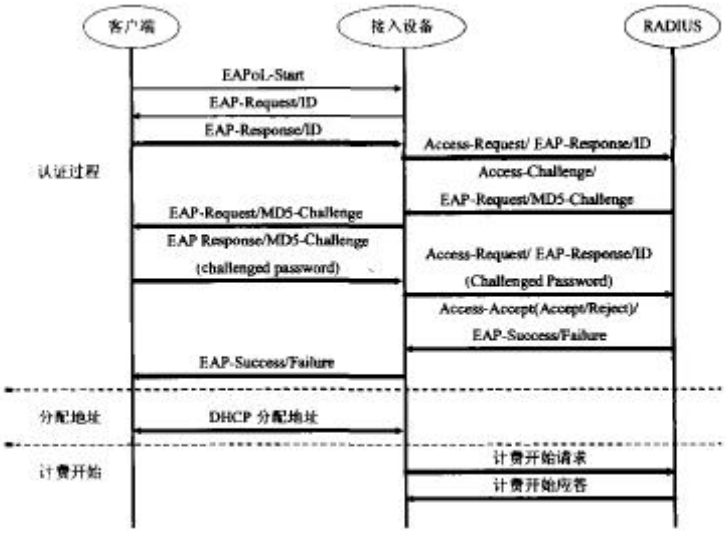

图 3 基于 EAP-MD5 的 802.1x 认证流程

(1)客户端向接人设备发送一个 EAPOL-Start 报文，开始 802.1x 认证接人；

(2)接人设备向客户端发送 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 给接人设备的请求，其中包括用户名；

(4)接人设备将 EAP-Response/Identity 报文封装到 RADIUSAccess．Request 报文中，发送给认证服务器；

(5)认证服务器产生一个 Challenge，通过接人设备将 RA DIUSAccess—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-Password，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 DHCPRelay)，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。

认证通过之后的保持：认证端 Authenticator 可以定时要求 Client 重新认证，时间可设。重新认证的过程对 User 是透明的(应该是 User 不需要重新输入密码)。

下线方式：物理端口 Down；重新认证不通过或者超时；客户端发起 EAP_Logoff 帧；网管控制导致下线；

现在的设备（switch）端口有三种认证方式：

（1）ForceAuthorized：端口一直维持授权状态，switch 的 Authenticator 不主动发起认证；

（2）ForceUnauthorized：端口一直维持非授权状态，忽略所有客户端发起的认证请求；

（3）Auto： 激活 802.1X，设置端口为非授权状态，同时通知设备管理模块要求进行端口认证控制，使端口仅允许 EAPOL 报文收发，当发生 UP 事件或接收到 EAPOL-start 报文，开始认证流程，请求客户端 Identify，并中继客户和认证服务器间的报文。认证通过后端口切换到授权状态，在退出前可以进行重认证。

802.1x 协议的认证端口

受控端口：在通过认证前，只允许认证报文 EAPOL 报文和广播报文（DHCP、ARP）通过端口，不允许任何其他业务数据流通过；

逻辑受控端口：多个 Supplicant 共用一个物理端口，当某个 Supplicant 没有通过认证前，只允许认证报文通过该物理端口，不允许业务数据，但其他已通过认证的 Supplicant 业务不受影响。

现在在使用中有下面三种情况：

(1)仅对使用同一物理端口的任何一个用户进行认证（仅对一个用户进行认证，认证过程中忽略其他用户的认证请求），认证通过后其他用户也就可以利用该物理端口访问网络服务

(2)对共用同一个物理端口的多个用户分别进行认证控制，限制同时使用同一个物理端口的用户数目（限制 MAC 地址数目），但不指定 MAC 地址，让系统根据先到先得原则进行 MAC 地址学习，系统将拒绝超过限制数目的请求，若有用户退出，则可以覆盖已退出的 MAC 地址。

(3)对利用不同物理端口的用户进行 VLAN 认证控制，即只允许访问指定 VLAN，限制用户访问非授权 VLAN；用户可以利用受控端口，访问指定 VLAN，同一用户可以在不同的端口访问相同的 VLAN。

五 EAPOL 协议的介绍

IEEE 802.1x 定义了基于端口的网络接入控制协议，需要注意的是该协议仅适用于接入设备与接入端口间点到点的连接方式。为了在点到点链路上建立通信，在链路建立阶段 PPP 链路的每一端都必须首先发送 LCP 数据包来对该数据链路进行配置。在链路已经建立起来后，在进入网络层协议之前，PPP 提供一个可选的认证阶段。而 EAPOL 就是 PPP 的一个可扩展的认证协议。

下面是一个典型的 PPP 协议的帧格式：

Flag	Address	Control	Protocol	Information

当 PPP 帧中的 protocol 域表明协议类型为 C227(PPP EAP)时，在 PPP 数据链路层帧的 Information 域中封装且仅封装 PPP EAP 数据包，此时表明将应用 PPP 的扩展认证协议 EAP。

这个时候这个封装着 EAP 报文的 information 域就担负起了下一步认证的全部任务，下一步的 EAP 认证都将通过它来进行。

1．一个典型的 EAP 认证的过程分为：request、response、success 或者 failure 阶段，每一个阶段的报文传送都由 Information 域所携带的 EAP 报文来承担。

EAP 报文的格式为：

Code	Identifier	Length	Data

Code 域为一个字节，表示了 EAP 数据包的类型，EAP 的 Code 的值指定和意义如下：

Code＝1 ——→Request

Code＝2 ——→Response

Code＝3 ——→Success

Code＝4 ——→Failure

Identifier 域为一个字节，辅助进行 request 和 response 的匹配——每一个 request 都应该有一个 response 相对应，这样的一个 Identifier 域就建立了这样的一个对应关系——相同的 Identifier 相匹配。

Length 域为两个字节，表明了 EAP 数据包的长度，包括 Code, Identifier, Length 以及 Data 等各域。超出 Length 域范围的字节应该视为数据链路层填充（padding），在接收时应该被忽略掉。

Data 域为 0 个或者多个字节，Data 域的格式由 Code 的值来决定。

2．分别介绍 Code 为不同的值的时候报文的格式和各个域的定义。

当 Code 域为 1 或者 2 的时候，这个时候为 EAP 的 request 和 response 报文，报文的格式为：

Code	Identifier	Length	Type	Type Data

（1）当 Code 为 1 的时候是 request 报文，当 Code 为 2 的时候是 response 报文。

Identifier 域为一个字节。在等待 Response 时根据 timeout 而重发的 Request 的 Identifier 域必须相同。任何新的（非重发的）Request 必须修改 Identifier 域。如果对方收到了重复的 Request，并且已经发送了对该 Request 的 Response，则对方必须重发该 Response。如果对方在给最初的 Request 发送 Response 之前收到重复的 Request（也就是说，它在等待用户输入），它必须悄悄的丢弃重复的 Request。

Length 域为两个字节，表明 EAP 数据包的长度，包括 Code，Identifier，Length，Type 以及 Type-Data 等各域。超出 Length 域的字节应视为数据链路层填充（padding），在接收时应该被忽略掉。

Type 域为一个字节,该域表明了 Request 或 Response 的类型。在 EAP 的 Request 或 Response 中必须出现且仅出现一个 Type。通常 Response 中的 Type 域和 Request 中的 Type 域相同。但是，Response 可以有个 Nak 类型，表明 Request 中的 Type 不能被对方接受。当对方发送 Nak 来响应一个 Request 时，它可以暗示它所希望使用并且支持的认证类型。Type Data 域随 Request 和相对应的 Response 的 Type 的不同而不同。

Type 域的说明如下：

Type 域总共分为 6 个值域，其中头 3 种 Type 被认为特殊情形的 Type，其余的 Type 定义了认证的交换流量。Nak 类型仅对 Response 数据包有效，不允许把它放在 Request 中发送。

Type＝1——→Identifier

Type＝2——→Notification

Type＝3——→Nak（Response Only）

Type＝4——→MD5-Challenge
Type＝5——→One-Time Password (OTP)
Type＝4——→Generic Token Card

（2）当 Code 域为 3 或者 4 的时候，这个时候为 EAP 的 Success 和 Failure 报文，报文的格式为：

Code	Identifier	Length

当 Code 为 3 的时候是 Success 报文，当 Code 为 4 的时候是 Failure 报文
Identifier **域为一个字节，辅助匹配** Response **应答。**Identifier **域必须与其正在应答的** Response **域中的** Identifier **域相匹配。**

六 总结

1．IEEE 802.1x **定义了基于端口的网络接入控制协议，其中端口可以是物理端口，也可以是逻辑端口。**
2．802.1X **关心的只是一个端口（物理的或者逻辑的）是否打开，而不关心打开之后上来的是什么样的报文。**
3．802.1x **协议只是提供了一种用户接入认证的手段，它也只是对用户的认证进行控制，而接入网络设备必须具备的其他的一些安全和管理特性，由各厂家设备自行来提供的。**

本文内容全部参考自其他论文或相关资料