Star Hacking Part1

wangyao@cs.hit.edu.cn
ipconfigme@gmail.com

很早就看到校园网内用户抱怨实达怎么怎么不好用，Linux 的客户端不好使，由于自己使用的是网通 PPPOE 认证，可以共享上网，没有多大的问题，就一直没有留意这个问题。但是真正促使我研究它是出于 Cowoo 的邀请，Cowoo 说它的 Linux 再校园网内无法上网，我就决定研究这个实达认证。

锐捷了解篇

IEEE802.1x（Port-Based Network Access Control）是一个基于端口的网络接入控制标准，为 LAN 接入提供点对点式的安全接入。这是 IEEE 标准委员会针对以太网的安全缺陷而专门制定的标 准，能够在利用 IEEE 802 LAN 的优势基础上，提供一种对连接到局域网设备或用户进行认证的手段。

IEEE 802.1x 标准定义了一种基于“客户端——服务器”（Client-Server）模式实现了限制未认证用户对网络的访问。客户端要访问网络必须先通过认证服务器的认证。

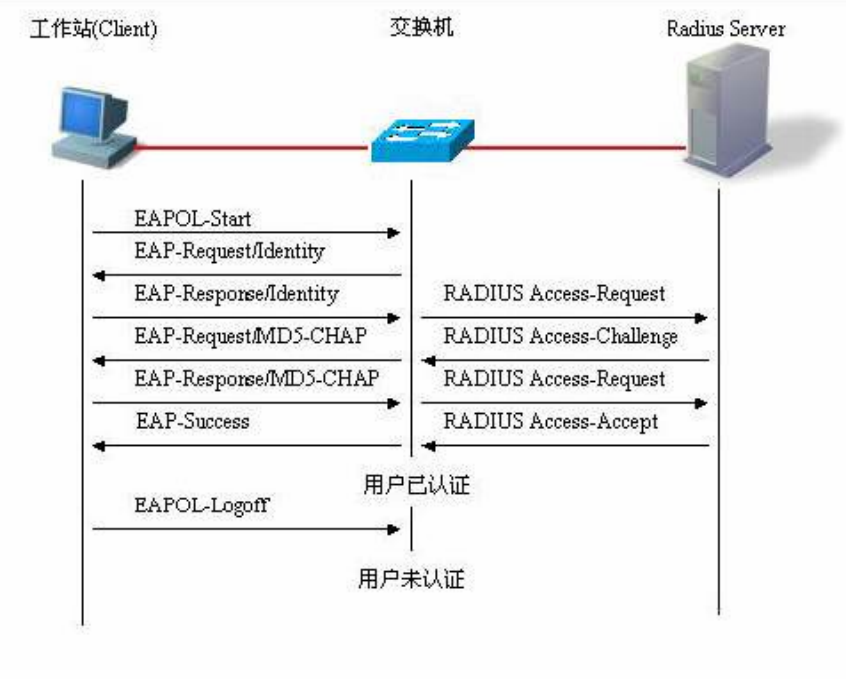
在客户端通过认证之前，只有 EAPOL 报文（Extensible Authentication Protocol over LAN）可以在网络上通行。在认证成功之后，通常的数据流便可在网络上通行。

认证的发起及认证过程中的报文交互

恳请者和认证者之间通过 EAPOL 协议交换信息，而认证者和认证服务器通过 RADIUS 协议交换信息，通过这种转换完成认证过程。EAPOL 协议封装于 MAC 层之上，类型号为 0x888E。同时，标准为该协议申请了一个组播 MAC 地址 01-80-C2-00-00-03，用于初始认证过程中的报文传递。

下图是一次典型的认证过程中，三个角色设备的报文交互过程

实达认证的总体流程图



锐捷网络产品特有的功能

为了方便宽带运营商及其他特殊场合的用途，锐捷对 802.1x 的功能在标准的基础上进行了扩展（该扩展是完全基于标准之上，没有任何的与 IEEE 802.1x 不兼容）

IP 授权模式

锐捷网络实现的 802.1x，可以强制要求已认证的用户使用固定的 IP。管理员通过配置 IP 授权模式来限定用户获得 IP 地址的方式。IP 授权模式有三种： DISABLE 模式、DHCP SERVER 模式、RADIUS SERVER 模式。下面分别介绍这三种工作模式的特性：

DISABLE 模式（默认）：在该模式下，交换机不对用户的 IP 做限制，用户只需认证通过便可以使用网络。

DHCP SERVER 模式：用户的 IP 通过指定的 DHCP SERVER 获得，只有指定的 DHCP SERVER 分配的 IP 才是合法的 IP。

RADIUS SERVER 模式：用户的 IP 通过 RADIUS SERVER 指定。用户只能用 RADIUS SERVER 指定的 IP 访问网络。

三种模式下的应用模型：

DISABLE 模式：适合不对用户限定 IP 的场合。用户只需通过认证便可以访问网络。

DHCP SERVER 模式：用户 PC 通过 DHCP 获得 IP 地址，管理员通过配置交换机的 DHCP RELAY 来限定用户访问的 DHCP SERVER，这样，只有指定的 DHCP SERVER 分配的 IP 才是合法的。

RADIUS SERVER 模式：用户 PC 使用固定的 IP，RADIUS SERVER 配置了<用户—IP>的对应关系，并通过 RADIUS 的 Framed-IP-Address 属性告知交换机，用户只能用该 IP 才能访问网络。

发布广告信息

锐捷网络实现的 802.1x，可以在 Radius Server 端配置 Reply-Message 字段，当认证成功后，该字段的信息可以在锐捷网络推出的 802.1x 客户端 Star-Supplicant 上显示出来，便于运营商发布一些信息。

该消息只有在用户第一次认证时显示，重认证时，不会显示，这样就避免了对用户的频繁打扰。

广告信息的显示窗口支持html，会自动把消息中的http://XXX.XXX.XX转换成可直接跳转的连接，便于用户查看详细的信息。

广告信息的发布：

1、运行商在 Radius Server 端，配置 Reply Message 属性的内容

2、只有 1 锐捷的客户端 Star-才支持（对本公司交换机的用户免费），其它的客户端 supplicant 看不到信息但不影响使用 、在交换机端无需设置

某端口下的可认证主机列表

为了增强 802.1x 的安全性，锐捷在不影响 IEEE 802.1x 的基础上进行了扩展，网管可以限定某个端口的认证的主机列表。如果一个端口下的可认证主机列表为空，则任何用户均可认证；若可认证主机列表不 为空，那么只有列表中的主机允许认证。允许认证的主机用 MAC 标识。

授权

为了方便运营商，我们的产品可以对不同类型的用户提供不同质量的服务，如：提供给用户的最大带宽不同。而这些信息集中于 Radius Server 上，管理员不必对每台交换机进行配置。

由于 Radius 没有标准的属性来表示最大数据率。锐捷只能通过厂商自定义属性来传递授权信息。我们定义的通用格式如下：

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+

| Type | Length | Vendor-Id

+-+

Vendor-Id (cont) | Vendor type | Vendor length |

+-+

| Attribute-Specific...

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

对于最大数据率应填的值如下：

+-+

| 0x1A | 0x0c | 0x00 0x00

+-+

0x13 0x11 | 0x01 | 0x06 |

+-+

| 最大数据率的十六进制值 |

+-+

最大数据率的单位：kbps

对于最大数据率为 10M 的用户应填如下：

+-+

| 0x1A | 0x0c | 0x00 0x00

+-+

0x13 0x11 | 0x01 | 0x06 |

+-+

| 0x00002710 |

+-+

自定义的头照上填，最大数据率为 10M，即 10000kbsp，转换成 16 进制则 0x00002710，填在相应的字段上即可。

代理服务器屏蔽和拨号屏蔽

用户自行架设代理服务器以及用户认证后再自行拨号上网，这是网络安全最大的两个隐患。锐捷网络交换机提供代理服务器屏蔽和拨号屏蔽功能。

实现该功能交换机端不需任何设置。只需在 RADIUS 服务器端配置相应的属性。由于 Radius 没有标准的属性来表示最大数据率。我们只能通过厂商自定义属性来传递授权信息。我们定义的通用格式见授权一节的说明。

代理服务器屏蔽功能定义的 Vendor type 为 0x20,拨号屏蔽功能定义的 Vendor type 为 0x21。Attribute-Specific 字段为四个字节的厂商自定义属性,定义了对使用代理服务器上网和拨号上网行为所采取的动作. 0x0000 表示正常连接，不进行检测屏蔽, 0x0001 表示进行检测屏蔽。

若要屏蔽通过代理服务器上网,用户应填如下信息:

+-+

| 0x1A | 0x0c | 0x00 0x00

+-+

0x13 0x11 | 0x20 | 0x06 |

+-+

|0x0001 |

+-+

客户端在线探测

为了保证计费的准确性，需要一种在线探测机制能够再短时间内获知用户是否在线。在标准实现中的重认证机制能够满足这种需求，但是标准实现中需要 RADIUS 服务器的参与，要实现准确的探测用户是否在线将会占用交换机和 RADIUS 服务器的大量资源。为了满足在占用少量资源的基础上实现计费的准确 性，我们采用了一种新的客户端在线探测机制。这种机制只需要在交换机和客户端之间交互，并且对网络流量的占用极小，能够实现分钟级的计费精度(用户可以通 过配置来确定计费的精度)。

锐捷协议分析篇

协议分析主要通过分析抓来的数据包以及对源代码的研读得来的。

注：下面我将以锐捷 2.56 版本的数据包进行描述。

一、数据包格式分析

认证过程中客户端和认证服务器之间运行 EAPOL 协议，二者之间传送的信息均为以太网帧格式。每一帧均包含如下头信息。

01 80 C2 00 00 03 AA BB CC DD EE FF 88 8E 01 01
00 00 ……

前 6 个字节为目标地址，之后 6 个字节是源地址，这里假定为 AA BB CC DD EE FF；第 13，14 个字节是协议类型（proctol type），802.1x 对应的值为 88 8E；第 15 个字节是协议版本号（proctol version）；16 字节是帧类型(packet type)；第 17、18 字节为帧的长度（Packet Body Length），是指头信息之后（从第 19 字节开始）的有效数据的长度。

数据结构定义为：

```
static byte PackageHeader[0x12] = {
////////////////////////////////////////////////////////////////////////////////////////
// Standard 802.1x
// 0x00 --> 0x11
        0x00,0x00,0x00,0x00,0x00,0x00,  // Destination MAC
        0x00,0x00,0x00,0x00,0x00,0x00,  // Source MAC
        0x88,0x8E,                      // Ethertype = 0x888E (8021X)
        0x01,                           // Version = 1
        // Packet Type    0x00 ;0x01,EAPOL-Start ;0x02 ;0x03 ;0x04
        0x01,
        0x00,0x00                       // Packet Body Length
        };    //
```

另外每一帧的有效数据之后还必须包含实达专用的附加数据包。

（固定字符串用黑色表示，灰色底色的是需要填充的）

FF FF 37 77 FF 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 13 11 38 30 32 31 78
2E 65 78 65 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 02 38 00 00 00
00 00 13 11 00 28 1A 28 00 00 13 11 17 22 37 34
39 33 34 34 39 38 32 32 31 32 32 43 33 37 34 34

批注 [w1]: 固定串不太清楚什么意思

批注 [w2]: Mystar 中的 Ablog 算法

37 39 34 35 35 30 37 36 45 44 33 38 38 34 1A 0C
00 00 13 11 18 06 00 00 00 00 1A 0E 00 00 13 11
2D 08 00 00 00 00 00 00 1A 08 00 00 13 11 2F 02

批注 [w3]: 跟 Mento 不一样的地方，添加了两个字节

这 144 字节为实达专用附加数据包，在每一帧的后面都要附加这个数据包。

5 --> 22 字节要填充本机相应的信息：IP、掩码、网关、 DNS，还有一个不知道是什么意思的两字节的 circleCheck，这些信息都要经过加密，加密算法在 mystar 里叫做 Alog。

23 --> 58 字节，是一个字符串说明为 ASCII 8021x.exe。

59 --> 62 字节表示锐捷的版本号：02 38 00 00，即为 2.56。

63 字节，我现在还不太清楚是什么意思，有时是 0，有时是 1。

64-->77 字节是一个固定的字符串，每一次的数据包都是一样的。

78 --> 109 这 32 个字节，好像是 0－F 的随机数，不过现在还不能够确定，因为这些随机数之间可能存在一些联系。

110-120 字节是一个固定的字符串。

121 字节是用来标记 DHCP 的启用的。（这是五山高校联盟论坛上说的，我没有测试）。

122 -->129 字节是一个固定的字符串。

130 --> 135 字节是网卡的 MAC 地址。

136 --> 143 是一串固定的字符串。

```
static byte RuijieExtra[144] = {            //Ruijie OEM Extra    by soar
    ////////////////////////////////////////////////////////////////////////////
    // OEM Extra
    // 0 --> 22
    0xFF,0xFF,0x37,0x77,0xFF,
    0x00,0x00,0x00,0x00,                    // Encode( IP )
    0x00,0x00,0x00,0x00,                    // Encode( SubNetMask )
    0x00,0x00,0x00,0x00,                    // Encode( NetGate )
    0x00,0x00,0x00,0x00,                    // Encode( DNS )
    0x00,0x00,                              // Checksum( )

    // 23 --> 58    ASCII 8021x.exe
    0x00,0x00,0x13,0x11,0x38,0x30,0x32,0x31,0x78,0x2E,0x65,0x78,0x65,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00,
    // 59 --> 62
    0x02,0x38,0x00,0x00,                    // 8021x.exe File Version (2.56.00)
    //63
    0x00,                                   // unknow flag(有时为；有时为)
    //64-->77    Const strings
    0x00,0x00,0x13,0x11,0x00,0x28,0x1A,0x28,0x00,0x00,0x13,0x11,0x17,0x
22,
    // 78 --> 109    32 bits spc. Random strings or MD5 Hash
    0x00,0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x0A,0x0B,0x0C,0x
0D,
```

```
    0x0E,0x0F,0x00,0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x0A,0x
0B,
    0x0C,0x0D,0x0E,0x0F,
    //110-120   Const strings
    0x1a,0x0c,0x00,0x00,0x13,0x11,0x18,0x06,0x00,0x00,0x00,
    // 121
    0x00,                                   // DHCP and first time flag

    // V2.56 (and upper?) added
    // 122 -->129 Const strings
    0x1A,0x0E,0x00,0x00,0x13,0x11,0x2D,0x08,
    // 130 --> 135   True NIC MAC
    0x00,0x00,0x00,0x00,0x00,0x00,
    //136 --> 143   Const strings
    0x1A,0x08,0x00,0x00,0x13,0x11,0x2F,0x02
};
```

二、认证的详细过程

1.客户端发起认证（EAPOL-Start）

01 80 C2 00 00 03 AA BB CC DD EE FF 88 8E 01 01
00 00

发起认证时客户端并不知道服务器的 MAC 地址，可以用标准组播地址（01 80 C2 00 00 03），也可以用实达的私有组播地址（01 D0 F8 00 00 03）。EAPOL-Start 帧的帧类型为 01。

2.服务器端请求用户名（EAP-Request/Identity）

AA BB CC DD EE FF 00 D0 F8 FD 5F 1C 88 8E 01 00
00 05 01 01 00 05 01

从这里可以得到服务器的地址 00 D0 F8 FD 5F 1C，并填充到以后的帧中；第 19 字节是帧类型，01 表示请求；第 20 字节是客户端发送用户名时要用的 Identity。

3.客户端发送用户名（EAP-Response/Identity）

00 D0 F8 FC 78 4E AA BB CC DD EE FF 88 8E 01 00
00 13 02 01 00 13 01 ……(Name)……
(这里 Name：a1106290100141) 0x13 = 5+14
Packet Type 为 00，Packet Body Length 是从第 19 字节到用户名最后一个字符的长度，则此处为 00 13,这里用的是网络字节顺序（network byte order）；19 字节 02 表示 Response；20 字节的 01 是上面包里的 Identity，以后每一步都相同，故不再重复；21、22 字节含义与第 17、18 字节相同；23 字节表示发送的是用户名；第 24 字节开始是用户名的 ASCII 码。

4.服务器端请求密码（EAP-Request）

AA BB CC DD EE FF 00 D0 F8 FD 5F 1C 88 8E 01 00
00 16 01 92 00 16 04 10 01 02 03 04 05 06 07 08
09 00 01 02 03 04 05 06

批注 [w4]: 密钥

这里需要用到的除了 Identity，还有从第 24 字节开始的密钥信息，第 24 字节是密钥的长度，之后是密钥。这个密钥用来加密将要发送的用户密码。

5.客户端发送密码（EAP-Response）

00 D0 F8 FD 5F 1C AA BB CC DD EE FF 88 8E 01 00
00 24 02 92 00 24 04 10 XX XX XX XX XX XX XX XX
XX XX XX XX XX XX XX XX ?? ?? ?? ?? ?? ?? ?? ??
(这里 Name：a1106290100141) 0x24 = 6+16+14
16 个字节的 XX 是加密后的密码，是用 MD5 算法算出来的，生成方法是把 Identity、密码和加密密钥按顺序保存在一个数组中，再计算其 MD5。之后的??是用户名。

6.服务器表明认证结果

AA BB CC DD EE FF 00 D0 F8 FD 5F 1C 88 8E 01 00
00 94 03 92 00 04 00 00 13 11 00 00 00 00 13 11
00 49
.....(省略 5 行)
00 00 00 00 00 00 00 00 00 00 00 00 13 11 01
01 00 00 13 11 01 01 FF FF 37 77 AF 7F FF FF A7
FC FF FF FF A7 FF
0x94=0xa5-0x12+1
第 19 个字节帧类型为 03 表示表示成功，04 表示认证失败。我们要记下最后一行的 FF FF A7 FC，它在帧中的偏移为 packet body length + 9；

7.保持激活状态

00 D0 F8 FD 5F 1C AA BB CC DD EE FF 88 8E 01 BF
00 1E FF FF 37 77 7F 9F F7 FF 00 00 FF FF 37 77
7F 9F F7 FF 00 00 FF FF 37 77 7F 3F FF

上面是 mystar 中的原始数据，其中第 35 至 38 字节由一个初值为 0x1000002A 的序列号经 Alog 算法加密生成的 mystar 里叫做 Alog，该序列号每次使用前加一。第 19 至第 22 字节由上面包中的 FF FF A7 FC 加上序列号再经 Alog 算法生成。每过一段时间就需要发送该帧一次以维持激活状态。而 hustauth 的做法则是每隔 45 秒就重新认证一次。

附录：

网络问题的基本分析方法：

1、确认物理层连接正常：看看你的网卡的信号，网线有没有问题，接到的交换机的端口正不正常

2、确认数据链路层连接正常：这里一般是交换协议和点对点协议跑的层。802.1x 是 Port-based Network Access Control，它是利用 EAP over Lan 的协议进行认证，也就是说，EAP 的 packet 是封装在 ethernet 的 frame，更准确的说，是封装在 802.2 LLC 协议的封包里广播出去的，接受到的包也是对方通过 ethernet 广播返回的。如果你在本机侦听 ethernet 协议的封包，可以侦听到交换机的端口地址。要检测数据链路层是否正常，可以用 tcpdump 来进行。首先，你需要在 windows 连接正常的情况下，获得交换机端口的 MAC 地址，还要记住本机的 MAC 地址。在 Linux 可以用 ifconfig 查看。假设你获得的对方的 MAC 地址为 00:0C:00:00:00:01，本机的 MAC 地址为 FF:FE:00:00:00:01，本机的网卡是 eth0，则可以用如下命令侦听：

代码:

```
tcpdump -i eth0 ether src FF:FE:00:00:00:01 and ether dst 00:0C:00:00:00:01
tcpdump -i eth0 ether dst FF:FE:00:00:00:01 and ether src 00:0C:00:00:00:01
```

当然，如果你对 tcpdump 熟悉，也可以用其他方法侦听 ethernet 层的封包。

第一个命令是抓取发到对方的封包，看在本机是否正常发出请求验证的包。第二个命令是抓取对方发出的封包，看在本机是否正常收到对方响应。如果第一个命令和第二个命令均能抓到包，那就表示数据链路层运作正常。这时就要看这个软件是否有用户名/密码错误，或者是其他软件设置的问题。如果数据链路层运作不正常，那可能是这个软件配置不正确，或者是设备没配置好，或者是链接库错误。

参考资料：

1、 mystar0.11 源码

2、 Mento Supplicant 源码

3、锐捷 supplicant 认证程序 Mento Supplicant for Ruijie（http://www.53uc.com/bbs/dispbbs.asp?boardid=8&replyid=174610&id=8022&page=4&skin=0&Star=1）

4、锐捷（实达）认证过程分析http://www.lcuc.org/node/49

5、星网锐捷 802.1x 客户端认证协议分析方法 http://www.53uc.com/bbs/dispbbs.asp?boardID=8&ID=7910&page=9

6、 802.1x http://blog.9zi.com/post/1/636

7、《网络安全开发包详解》 刘涛编著 电子工业出版社

8、 IEEE802.1x 访问控制协议(IEEE Std 802.1X-2001)

9、 EAP 协议 RFC3748、RFC3579

后注：另外 Linux 下的 802.1x Supplicant **开源项目**有

Xsupplicant http://open1x.sourceforge.net/

wpa_supplicant http://hostap.epitest.fi/wpa_supplicant/