基于 802.1X/EAP-PEAP 的个性化接入认证机制研究①

卞传政

(武汉大学 信息管理学院, 武汉 430072)

摘 要: 研究了在人们对网络依赖程度越来越高、网络安全问题不断涌现以及用户多台移动终端需要同时访问网络的情况下建立便捷高效的个性化接入认证机制的问题. 基于 IEEE802.1X 标准, 使用请求者系统、认证系统和认证服务器系统的三方架构, 分析了 EAP 协议, 对 EAP-MD5、EAP-TLS、EAP-SIM、EAP-PEAP 四类认证方法做了比较, 并根据需要选择 EAP-PEAP 认证方法. 然后在了 EAP-PEAP 标准的基础上设计了详细的认证流程, 从系统和用户两个角度实现了个性化的接入认证机制.

关键词: 802.1X; EAP-PEAP; 网络接入; 个性化

Personalization Access Authentication Mechanism Based on 802.1X/EAP-PEAP

BIAN Chuan-Zheng

(School of Information Management, Wuhan University, Wuhan 430072, China)

Abstract: This paper studied a convenient and efficient network situations personalized access authentication mechanisms. It uses the tripartite framework of requester, authenticator and authentication server based on IEEE802.1X standards. It analyzes the EAP protocol, compares the EAP-MD5, EAP-TLS, EAP-SIM, EAP-PEAP and chooses the EAP-PEAP authentication method. Then it gives the detailed design of the authentication process with system and user personalization.

Key words: 802.1X; EAP-PEAP; network access; personalization

近年来, 网络的发展越来越快, Web2.0 时代的全面到来, 智能终端设备的迅速普及, 各种新事物的不断涌现, 使得网络迅速而深刻的改变了人们的生活方式, 网络成了生活不可或缺的一部分, 人们希望能够尽可能方便的获取网络接入服务. 同时, 由于通信网络无法保证通信的安全性, 各方通信的内容随时都受到网络嗅探、内容篡改和破坏等的威胁, 网络安全性问题日益突出[1]. 在另一个方面, 目前大多数的实际使用的接入系统有校园网锐捷认证、Linux 环境中的 XSupplicant 以及专为无线网络设计的 Wpa_Supplicant, 这些都没有考虑单个用户多台设备需要同时访问网络的个性化需求, 这些用户的移动终端必须自行寻找额外的 Wi-Fi 接入点, 使得用户体验大打折扣. 用户一个账户只能登陆同时登陆一台设备, 多台设备同时登陆时需要多个账号, 这就增加了用户管理账户的复杂度. 面对这些问题, 如何建立一个通用便捷高效的个性化网络接入认证机制来适应这些新的形势是很多系统需要考虑的问题.

① 收稿时间:2013-12-17;收到修改稿时间:2014-02-17

1 802.1X认证框架

1.1 802.1X 简介

作为 IEEE802.1 网络协议族的一部分, 802.1X[2] 是 IEEE 提出关于用户接入网络的认证标准, 全称“基于端口的网络接入控制”. 为了配合无线网络的接入, 2004 年进行了修订改版, 从而为想要连接 LAN 或者 WLAN 的设备提供了一种认证机制.

802.1X 机制拒绝未经授权的非法用户通过接入设备的端口访问网络, 在这种机制中, 一般使用支持 802.1X 协议的网络交换机作为接入设备.

1.2 802.1X 的系统架构

802.1X 就好像国境上的关卡, 只有当用户拥有正确的凭据(用户名密码等信息), 才可以通过这个关

© 中国科学院软件研究所 http://www.c-s-a.org.cn

卡. 如图 1 所示, 完整的系统是一个典型的 C/S 架构, 由请求者系统、认证系统和认证服务器系统三部分构成:

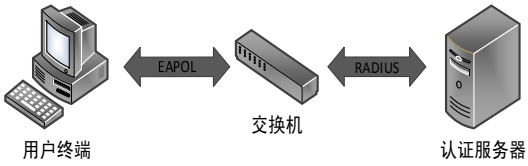

图 1 802.1X 系统架构

用户终端是一个设备实体, 可以是用户的台式机、笔记本, 随着智能设备的不断普及, 也可以是用户的智能手机、平板电脑等移动终端. 终端上需要安装有 802.1X 认证客户端.

接入设备一般采用交换机, 对用户终端的认证进行中继和相关的设置, 主流的网络交换机都可以进行 802.1X 认证.

认证服务器中多采用 RADIUS 服务器, 用于用户认证时的校验、用户行为的管理以及用户上下线时的计时计费.

1.3 802.1X 基本概念

在 802.1X 框架中, 规定了一下基本概念:

① 非受控端口与受控端口

用户通过任何一个端口与接入设备相连接, 这一个端口逻辑上都分为两个: 非受控端口与受控端口.

非受控端口用于用户在非授权模式下发送或接收 EAPOL 帧完成认证.

受控端口只有在用户通过认证后才开通, 用户通过此端口访问网络.

② 非授权状态与授权状态

接入设备接收到认证服务器返回的认证结果后, 对相应的端口状态进行设置.

③ 受控方向

受控方向仅在非授权状态下有效, 分为单项受控和双向受控.

双向受控表示该端口禁止任何帧的发送和接收, 单向受控表示可以向用户终端发送帧, 但不能从用户终端接收帧.

1.4 802.1X 认证的发起模式

802.1X 的接入过程既可以由用户终端来主动发起, 也可以由接入设备主动发起. 在前一种模式中, 由用户终端首先发送 EAPOL-Start 数据帧来发起认证过程. 在大多数网络应用情况下, EAPOL-Start 数据帧的目的地址是一个由标准规定的组播 MAC 地址: 01-80-C2-00-00-03, 当然, 目的地址也可以设为接入设备的 MAC 地址. 在第二种模式中, 接入设备需要定时主动发送 EAP-Request/Identity 报文来发起认证, 在这种模式中, 由于需要接入设备不断循环发送 EAP-Request/Identity 报文, 交换机负载增加, 故而网络效率会受到一定程度的影响, 因此一般多采用效率更高的用户终端主动发起模式, 只有当用户终端不支持主动发起时, 才采用接入设备主动发起模式.

2 EAP协议

2.1 EAP 协议的发展

802.1X 标准定义了一个在以太网框架内传输 IETF 定义的 EAP 数据帧的 EAPOL 协议[3], 在用户终端与接入设备中间使用可扩展认证协议(Extensible Authentication Protocol, EAP), 由于用户终端和接入设备往往通过局域网络直接连接, 因此 EAP 报文使用 EAPOL(EAP OVER LAN)封装后直接承载在 LAN 环境中.

可扩展认证协议(EAP), 是一个普遍使用的认证框架, 它在 RFC 3748《Extensible Authentication Protocol (EAP)》[4]中定义, 代替了之前的 RFC 2284《PPP Extensible Authentication Protocol (EAP)》, 后来又被 RFC 5247《Extensible Authentication Protocol (EAP) Key Management Framework》[5]再次更新定义.

与传统的在链路建立阶段提供认证不同, EAP 协议把认证过程推迟到认证阶段完成, 这使得认证方可以在获得更多的信息后, 采用某种特定的认证方式进行认证. 通常, 认证端将 EAP 报文(EAP Packet)直接传递给后方的认证服务器, 由认证服务器真正实现各种认证方式, 认证端只需根据认证服务器传递的成功(EAP-Success)或失败类型(EAP-Failure)的数据包来终止认证过程[6].

2.2 报文封装格式

EAPOL 使用在用户终端和接入设备之间发送 EAP 协议报文的一种封装规范, EAPOL 数据包的格式如图 2 所示:

© 中国科学院软件研究所 http://www.c-s-a.org.cn

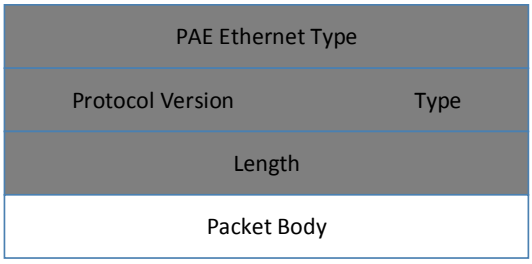

图 2 EAPOL 封装格式

各数据域对应的意义如表 1 所示:

表 1 EAPOL 各数据域

数据域	长度(字节)	代表的意义
PAE Ethernet Type	2	当前协议类型, 使用 0x888E
Protocol Version	1	发送方所支持的EAPOL 协议版本号
Type	1	表示 EAPOL 数据帧的类型
Length	2	Packet Body 的长度, 单位为字节

Packet Body 域包含 EAP 报文格式如图 3 所示:

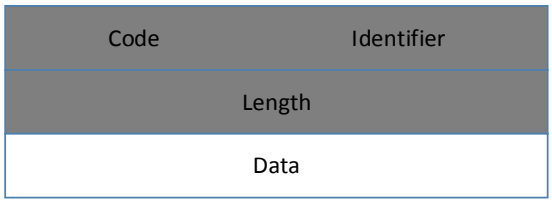

图 3 EAP 数据包格式

各数据域对应的意义如表 2 所示:

表 2 EAP 数据包各数据域

数据域	长度(字节)	代表的意义
Code	1	EAP 报文类型, 共 Request、Response、Success、Failure 四种, 当取 Success 或 Failure 时没有 Data 域
Identifier	1	用于匹配 Request 消息和 Response 消息.
Length	2	EAP 包的长度

2.3 EAP 认证方法比较和选择

作为一个总体的框架, EAP 在实际的认证流程中需要先选择具体的 EAP 方法, IETF 的 RFC 中提供了多达 40 余种认证方法. 尽管如此之多, 但是从逻辑上来讲, 大体分为四类, 第一类以 EAP-MD5 为代表, 实现极为简单, 部署和管理非常容易, 但安全性能也比较差, 难以抵抗中间人攻击, 也可以用字典式攻击破解. 如今已经不提倡使用, 但诸如很多学校的认证机制仍然采用 EAP-MD5, 安全难以保证. 第二类以 EAP-TLS 为代表, 作为 SSL 的继承者, EAP-TLS 提供了非常好的安全性能, 由于采用双向数字证书认证, 数字证书基于数字签名技术, 他是一种主动安全防御策略, 为信息传输提供安全保护[7], 它有效地防止了中间人攻击, 即便用户密码发生泄漏, 由于攻击者没有证书也无法认证成功. 它的缺点也是由于双向证书, 每一个客户端都要拥有自己的证书, 使得用户的负载比其他方法都大, 证书的发放和管理也需要专门的系统, 所以 EAP-TLS 因为配置困难而很少使用. 第三类以 EAP-SIM 为代表, 借助智能卡这样的物理设备来实现密钥的安全传输, 本质上是通信双方通过物理接触的方式协商好通信的加密套件和压缩算法等参数, 它主要应用在移动网络运营商和移动设备用户之间的通信之中, 在一个集团、学校、公司这样的应用场景中, 其部署难度太大.

为了同时兼顾认证安全性和部署管理复杂度, 在 RFC 5281《Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)》[8]中, IETF 提出了 EAP-TTLS 方法, 它提供极其优秀的安全性能, 而且不要求在用户终端上安装数字证书, 部署管理仅仅需要通过一个轻量级的客户端软件即可完成.

微软在《Microsoft's PEAP version 0 (Implementation in Windows XP SP1)》[9]和《Protected EAP Protocol (PEAP) Version 2》[10]中提出了 EAP-PEAP 的认证方式, 由于它仅仅要求认证服务器端安装数字证书而客户端不需要安装, 因此与 EAP-TTLS 非常相似. 通信过程中使用服务器端的证书协商出安全的对称加密密钥来保证通信安全, 由于 EAP-PEAP 是由思科、微软联合 RSA Security 共同提出的开放标准, 它已经被 WPA 和 WPA2 标准批准, 因此本文在 EAP-PEAP 认证机制上做一些思考和改进.

2.4 EAP-PEAP 认证流程

根据 PEAP 协议, 客户端软件和认证服务器之间的接入认证应分为两个阶段: TLS 握手阶段和 TLS 通道阶段.

第一个阶段是客户端通过数字证书验证服务器端身份并建立安全通道, 完成 TLS 握手, 成为 TLS 握手阶段, 由于对认证服务器的合法性校验、在不安全的网络上协商安全加密通道均是在 TLS 握手阶段完

© 中国科学院软件研究所 http://www.c-s-a.org.cn

成, 毫不夸张的说系统的安全性主要靠第一阶段的握手策略来保证.

我们通过引入用户设备清单对终端的 MAC 地址的合法性进行鉴别, 用户清单中包含用户名、设备号、设备 MAC 地址等信息.

下面让我们通过图 4 来详细分解下认证流程:

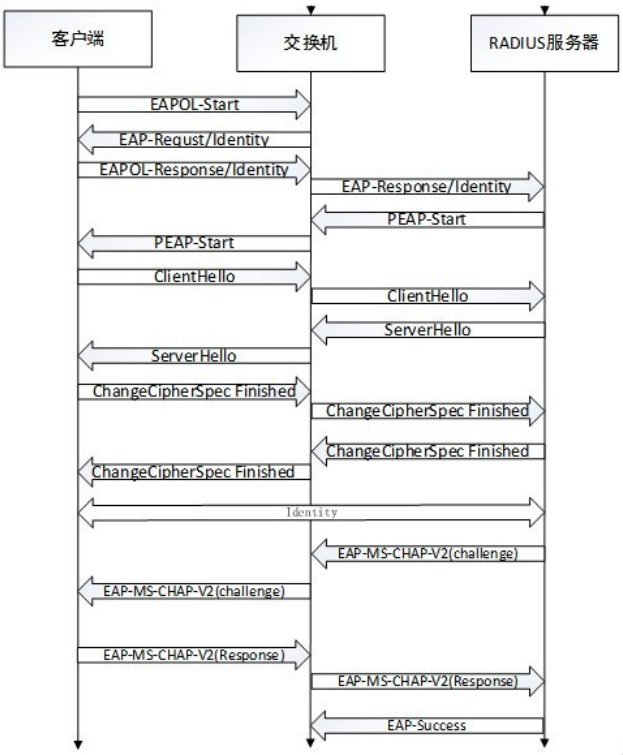

图 4 EAP-PEAP 认证流程

1) 客户端发送 EAPOL-Start 帧启动认证, 接入设备收到后, 发送 EAP-Request/Identity 数据帧, 请求用户的身份信息.

2) 客户端发送 EAP-Response/Identity 数据帧, 数据帧中的 Identity 域存放用户的身份标识符(如用户名@域名).

3) 服务器校验客户端的 MAC 地址是否在用户的设备清单之后, 如果不存在不予响应, 否则发送 EAP-Request/PEAP/Start 数据帧启动 PEAP 认证.

4) 客户端根据自己支持的协议版本、Session ID、加密套件、压缩算法以及随机数生成 EAP-Response/ClientHello 数据帧, 然后发送出去.

5)服务器挑选支持的一组加密算法, 加上自己生成的服务器端随机数、Session ID、压缩算法组成 ServerHello, 然后将 ServerHello、服务器的数字证书、ServerKeyExchange、ServerHelloDone 消息发送给客户端.

6) 客户端收到后首先使用 CA 证书颁发机构获取的根证书验证服务器的数字证书, 验证认证服务器证书的时间和名称是否合法, 如果合法则提取认证服务器数字证书中的公钥, 同时客户端软件随机产生一个预主密码 PreMasterSecret, 然后使用认证服务器的公钥对预主密钥加密得到 ClientKeyExchange, 然后将 ClientKeyExchange、ChangeCipherSpec 和 Finished 消息发送给服务器.

7) 服务器验证 Finished 消息的正确性, 如果正确使用自己的数字证书对应的私钥对 ClientKeyExchange 进行解密, 从而认证服务器就可以获得客户端随机生成的预主密钥, 通信双方对预主密钥、第 4 步中的客户端随机数以及第 5 步中的认证服务器随机数进行相同的运算得到对称加密密钥、加密初始化向量以及 HMAC 密钥[11]. 至此, 通信双方已经安全的协商出了一套加密通道, TLS 通道建立成功. 服务器发送自己的 ChangeCipherSpec 和 Finished 消息给客户端.

8) 客户端回复一个空响应启动第二阶段.

服务器收到后发送一个 EAP-Request/Identity 数据帧, 请求用户的身份标识.

9) 客户端回复 EAP-Response/Identity 数据帧, 数据帧中包含用户的身份标识.

10)服务器发送 EAP-Request/EAP-MS-CHAP-V2 挑战消息, 包含对用户认证的挑战消息.

11)客户端使用用户名、密码、MAC 地址、随机数等信息进行 SHA1 散列运算得到挑战应答, 然后回复 EAP-Response/EAP-MS-CHAP-V2 挑战应答, 并且包含自己的一个挑战消息.

12)服务器校验客户端的挑战应答, 如成功则发送 RADIUS Access-Challenge/EAP-Request/PEAP/EAP-ExtensionsSuccessResultTLV, 指出客户端软件的应答是正确的, 并且包含客户端软件发送的挑战消息.

13)客户端检查挑战信息是否与自己发送的一致, 如果一致则回复 EAP-Response/PEAP/EAPExtensionsSuccessResultTLV 来指出服务器的回应消息是正确的.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

14)服务器发送一个 EAP-Success 消息, 认证成功. 用户终端可以访问网络, 认证服务器开始计费.

3 个性化策略

从系统和用户两个方面考虑个性化策略.

尽管整个认证流程主要参考很多软件支持的国际标准, 但考虑到实际的需求, 很多集团并不允许一个第三方的接入申请软件在自己的内网中通过认证, 以免对集团的信息安全造成潜在的威胁; 在很多集团或者公司也会因为诸如上市之前要求对整体的设备终端情况有一个统计而希望有自己的客户端软件. 为了拒绝第三方软件通过认证, 就需要通过加入系统个性化的认证机制来实现. 在本认证机制中, 除了认证用户名和密码的组合, 还把用户终端设备的 MAC 地址按照自定义的方式参与到最终认证凭据的散列运算之中, 这就使得其他认证软件无法通过认证.

传统的接入控制系统设计之时, 单个用户往往只有一台终端需要访问网络, 因此并没有将单用户多终端的需求纳入考虑范围; 另一方面, 传统的系统为了认证的方便, 通常采用 IP 地址表或者 MAC 地址表的方式, 用户的认证凭据与这两种信息之一进行对应, 这就使得用户希望使用同一个账户在不同设备上同时登录的需求难以实现. 而随着单用户拥有设备数量的增加, BYOD(Bring Your Own Device)的现象越来越普遍, 因此同一个账户多处同时登陆就成了非常普遍的需求. 基于这样的考虑, 采用用户管理自己设备清单的方式来满足这一需求, 所有的用户都实现了自己管理自己的设备清单, 并且可以设置是否允许多处登陆等信息.

目前应用比较广泛的认证机制, 有校园网的锐捷认证、Linux 环境下的 XSupplicant 和为无线网络设计的 Wpa_Supplicant.

锐捷认证是很多校园网的认证机制, 包括其客户端和服务端都比较成熟, 尽管作为商业软件无法获知其实现细节, 但从与其兼容的开源版本 Mentohust 来看, 锐捷认证多采用 EAP-MD5 认证机制. XSupplicant 认证是 Linux 上面的开源认证客户端, 考虑了多种认证方式, 但仅仅是标准的实现, 并没有涉及到服务器端的设计. Wpa_Supplicant 重点考虑了无线网络环境下的接入认证. 这些认证机制都无法满足当前单用户单账户多处同时登陆的需求.

笔者所设计的这个接入认证机制, 着重考虑了在较低的代价的基础上实现单用户单账号多处同时登陆的需求, 并对认证过程进行了个性化定制, 大大降低了用户端的管理复杂度. 在 2.4 节对认证流程的研究基础上, 可以非常方便地实现一个个性化的接入认证系统. 以笔者 2.66GHz 双核 CPU、4GB 内存、Windows8 系统作为客户端, Cisco802.1X 交换机作为认证中介, I5 四核处理器、8GB 内存、Windows8 作为服务器的测试结果显示, 有线接入认证时间平均为 2.6 秒, 使用无线网卡接入认证时间平均为 3.7 秒, 并且可以用户自己管理设备清单的方式实现合法设备的多处同时登陆.

总体来说, 本文所研究的认证机制使用了 EAP/PEAP 机制来保证服务器端的合法性, 然后使用服务器的证书构造安全的加密通道, 再使用在认证协议中加入个性化的 MAC 散列元素和账户密码来共同保证客户端的合法性, 从而在避免用户端证书带来部署复杂度的情况下实现了非常高的安全性, 最后使用用户设备清单机制来支持单用户单账号多处同时登陆, 从而减低了用户端的管理复杂度.

4 结语

面对如今人们对网络越来越高的需求, 802.1X 仅仅通过较低的代价实现基本不受限制的认证, 并且可以保证非法用户不能访问网络, 从而迅速得到广泛应用. 在众多的 EAP 认证方法中, EAP-PEAP 仅对服务器端提出了证书要求, 从而在保证足够安全的情况下, 大大降低了管理和部署的复杂度. 针对系统的个性化的需求, 需要制定合理的认证流程, 对于如今更加重要的用户个性化需求, 通过引入用户设备清单的方式使用户可以定制自己的多台设备的网络访问方案.

参考文献

1 石东源,卢炎生,王星华,段献忠.SVG 及其在电力系统软件图形化中的应用初探.继电器,2004,32(16):37–40.

2 黄志清.网络安全中的数据加密技术研究.计算机系统应用,2000,9(7):24–26.

3 IEEE Std 802.1X.Port Based Network Access Control, 2001.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

4 Hecher A, Labiod H. Pre-authenticated signaling in Wireless LANs using 802.1X access control. Global Telecommunications Conference, 2004.

5 IETF RFC3748. Extensible Authentication Protocol (EAP). June 2004.

6 IETF RFC5247. Extensible Authentication Protocol (EAP) Key Management Framework. August 2008.

7 何玲娜,史烈,陈小平.基于 EAP 无线安全认证系统的设计和实现.计算机工程与设计,2005,26(2):423–425.

8 曾孜.网络安全中的数字签名技术分析与应用.计算机系统应用,2001,10(8):33–35.

9 IETF RFC5281. Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0). August 2008

10 Internet-draft. Microsoft's PEAP version 0 (Implementation in Windows XP SP1). 25 October 2002.

11 Internet-draft. Protected EAP Protocol (PEAP) Version 2. 15 October 2004.

12 IETF RFC2246. The TLS Protocol Version 1.0. January 1999.

www.c-s-a.org.cn

www.c-s-a.org.cn

© 中国科学院软件研究所 http://www.c-s-a.org.cn