(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 106790308 A
(43)申请公布日 2017.05.31

(21)申请号 201710192304.6

(22)申请日 2017.03.28

(71)申请人 北京中电普华信息技术有限公司
地址 100192 北京市海淀区清河小营东路15号科研楼710室
申请人 国网信息通信产业集团有限公司
国家电网公司
国网冀北电力有限公司信息通信分公司

(72)发明人 张立新 苏丹 吴佳 刘超 董爱强 冯扬 廖明耀 齐志超 牟鹏

(74)专利代理机构 北京集佳知识产权代理有限公司 11227
代理人 王宝筠

(51)Int.Cl.
H04L 29/06(2006.01)
H04L 29/08(2006.01)

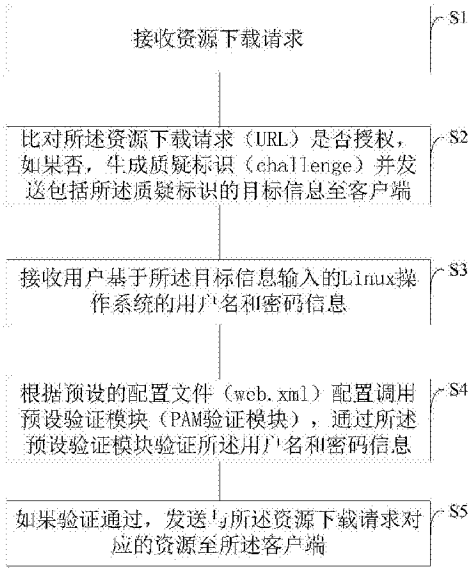
权利要求书1页 说明书7页 附图4页

(54)发明名称

一种用户认证方法、装置及系统

(57)摘要

本发明提供了一种用户认证方法、装置及系统，基于Http中的basic和Digest认证方法，结合Servlet规范中的资源授权管理，将web应用的用户管理和登录认证，委托给Linux系统用户管理和Pam认证模块；web应用不提供用户管理和认证的功能。从而提升用户访问的安全性，同时简化web客户端登录验证的过程，提升验证效率。



CN 106790308 A

1.一种用户认证方法,其特征在于,应用于应用服务器,所述用户认证方法包括:

接收资源下载请求;

比对所述资源下载请求是否授权,如果否,生成质疑标识并发送包括所述质疑标识的目标信息至客户端;

接收用户基于所述目标信息输入的Linux操作系统的用户名和密码信息;

根据预设的配置文件配置调用预设验证模块,通过所述预设验证模块验证所述用户名和密码信息;

如果验证通过,发送与所述资源下载请求对应的资源至所述客户端。

2.根据权利要求1所述的用户认证方法,其特征在于,所述比对所述资源下载请求是否授权,包括:

根据所述预设的配置文件,比对所述资源下载请求是否授权。

3.根据权利要求1所述的用户认证方法,其特征在于,所述根据预设的配置文件配置调用预设验证模块,包括:

在Linux操作系统中,预先配置各客户端的用户名和密码以及预设验证模块;

建立所述用户名和密码与所述预设验证模块的关联关系。

4.根据权利要求3所述的用户认证方法,其特征在于,所述通过所述预设验证模块验证所述用户名和密码信息,包括:

查找所述用户基于所述目标信息输入的Linux操作系统的用户名和密码信息是否属于所述预先配置的各客户端的用户名和密码,如果属于,则确定验证通过。

5.一种用户认证装置,其特征在于,应用于应用服务器,所述用户认证装置包括:

第一接收模块,用于接收资源下载请求;

比对模块,用于比对所述资源下载请求是否授权,如果否,生成质疑标识并发送包括所述质疑标识的目标信息至客户端;

第二接收模块,用于接收用户基于所述目标信息输入的Linux操作系统的用户名和密码信息;

验证模块,用于根据预设的配置文件配置调用预设验证模块,通过所述预设验证模块验证所述用户名和密码信息;

发送模块,用于当所述验证通过,发送与所述资源下载请求对应的资源至所述客户端。

6.根据权利要求5所述的用户认证装置,其特征在于,所述比对模块包括:

比对单元,用于根据所述预设的配置文件,比对所述资源下载请求是否授权。

7.根据权利要求5所述的用户认证装置,其特征在于,所述验证模块包括:

配置单元,用于在Linux操作系统中,预先配置各客户端的用户名和密码以及所述预设验证模块;

建立单元,用于建立所述用户名和密码与所述预设验证模块的关联关系。

8.根据权利要求5所述的用户认证装置,其特征在于,所述验证模块还包括:

查找单元,用于查找所述用户基于所述目标信息输入的Linux操作系统的用户名和密码信息是否属于所述预先配置的各客户端的用户名和密码,如果属于,则确定验证通过。

9.一种用户认证系统,其特征在于,包括如权利要求5-8所述的任意一项所述用户认证装置。

一种用户认证方法、装置及系统

技术领域

[0001] 本发明涉及数据处理技术，更具体涉及一种用户认证方法、装置及系统。

背景技术

[0002] 随着计算机技术以及安全技术的不断发展，用户认证的安全性日益凸显。通常，一般的应用程序在进行用户认证时，会将某种特定的认证方式硬编码到程序内部。比如，传统的用户登录程序，它先获得用户的用户名和密码，然后，将用户输入的密码进行计算得到密文，最后，将得到的密文和/etc/shadow文件中的该用户所在行的第二个字段进行比较，如果相同，则认证通过，否则，认证失败。

[0003] 具体的，目前http协议(超文本传输协议，Hyper Text Transfer Protocol)提供了Basic、Digest、Form、Cert四种方式的安全认证标准，其中，Basic和Digest认证方式简单、有效，但安全性不高；Form，Cert认证方式的认证工作量大，认证过程复杂。

[0004] 可见，如何提供一种用户认证方法，以提高用户认证的安全性，且解决目前web应用需同时开发完整的用户管理和认证管理功能，开发复杂、认证效率低的问题，成为当前亟待解决的一大技术问题。

发明内容

[0005] 有鉴于此，本发明提供了一种用户认证方法、装置及系统，结合了http(basic)和PAM模块的认证方式，增强了用户认证的安全性，且认证效率高。

[0006] 为实现上述目的，本发明提供如下技术方案：

[0007] 一种用户认证方法，应用于应用服务器，所述用户认证方法包括：

[0008] 接收资源下载请求；

[0009] 比对所述资源下载请求是否授权，如果否，生成质疑标识并发送包括所述质疑标识的目标信息至客户端；

[0010] 接收用户基于所述目标信息输入的Linux操作系统的用户名和密码信息；

[0011] 根据预设的配置文件配置调用预设验证模块，通过所述预设验证模块验证所述用户名和密码信息；

[0012] 如果验证通过，发送与所述资源下载请求对应的资源至所述客户端。

[0013] 优选的，所述比对所述资源下载请求是否授权，包括：

[0014] 根据所述预设的配置文件，比对所述资源下载请求是否授权。

[0015] 优选的，所述根据预设的配置文件配置调用预设验证模块，包括：

[0016] 在Linux操作系统中，预先配置各客户端的用户名和密码以及预设验证模块；

[0017] 建立所述用户名和密码与所述预设验证模块的关联关系。

[0018] 优选的，所述通过所述预设验证模块验证所述用户名和密码信息，包括：

[0019] 查找所述用户基于所述目标信息输入的Linux操作系统的用户名和密码信息是否属于所述预先配置的各客户端的用户名和密码，如果属于，则确定验证通过。

[0020] 一种用户认证装置，应用于应用服务器，所述用户认证装置包括：

[0021] 第一接收模块，用于接收资源下载请求；

[0022] 比对模块，用于比对所述资源下载请求是否授权，如果否，生成质疑标识并发送包括所述质疑标识的目标信息至客户端；

[0023] 第二接收模块，用于接收用户基于所述目标信息输入的Linux操作系统的用户名和密码信息；

[0024] 验证模块，用于根据预设的配置文件配置调用预设验证模块，通过所述预设验证模块验证所述用户名和密码信息；

[0025] 发送模块，用于当所述验证通过，发送与所述资源下载请求对应的资源至所述客户端。

[0026] 优选的，所述比对模块包括：

[0027] 比对单元，用于根据所述预设的配置文件，比对所述资源下载请求是否授权。

[0028] 优选的，所述验证模块包括：

[0029] 配置单元，用于在Linux操作系统中，预先配置各客户端的用户名和密码以及所述预设验证模块；

[0030] 建立单元，用于建立所述用户名和密码与所述预设验证模块的关联关系。

[0031] 优选的，所述验证模块还包括：

[0032] 查找单元，用于查找所述用户基于所述目标信息输入的Linux操作系统的用户名和密码信息是否属于所述预先配置的各客户端的用户名和密码，如果属于，则确定验证通过。

[0033] 一种用户认证系统，包括上述的任意一项所述用户认证装置。

[0034] 经由上述的技术方案可知，与现有技术相比，本发明实施例提供了一种用户认证方法，基于Http中的basic和Digest认证方法，结合Servlet规范中的资源授权管理，将web应用的用户管理和登录认证，委托给Linux系统用户管理和Pam认证模块；web应用不提供用户管理和认证的功能。从而提升用户访问的安全性，同时简化web客户端登录验证的过程，提升验证效率。

附图说明

[0035] 为了更清楚地说明本发明实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本发明的实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据提供的附图获得其他的附图。

[0036] 图1为本发明实施例提供的一种用户认证方法的流程示意图；

[0037] 图2为本发明实施例提供的Basic和Digest的认证的处理流程图；

[0038] 图3为本发明实施例提供的PAM认证模块的结构示意图；

[0039] 图4为本发明实施例提供的一种用户认证方法所应用的产品流程图；

[0040] 图5为本发明实施例提供的一种用户认证装置的结构示意图。

具体实施方式

[0041] 下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0042] 相关术语解释：

[0043] (1)http协议：超文本传输协议(HTTP,Hyper Text Transfer Protocol)是互联网上应用最为广泛的一种网络协议。

[0044] (2)servlet规范：Servlet是基于Java技术web组件容器托管的用于生成动态内容的规范。像其他基于Java的组件技术规范一样，Servlet也是基于平台无关的Java类格式，被编译为与平台无关的字节码，可以被基于Java技术的web server动态加载并运行。

[0045] (3)用户认证：用户认证是用户(客户端)向系统(服务器)证明自己的标识的过程。

[0046] (4)SG-APS：由本公司开发的基于java技术的应用服务器中间件产品。

[0047] 本发明实施例提供了一种用户认证方法，基于Http中的basic和Digest认证方法，结合Servlet规范中的资源授权管理，将web应用的用户管理和登录认证，委托给Linux系统用户管理和Pam认证模块；web应用不提供用户管理和认证的功能。从而提升用户访问的安全性，同时简化web客户端登录验证的过程，提升验证效率。

[0048] 具体的，请参阅图1，为本发明实施例提供的一种用户认证方法的流程示意图，该用户认证方法应用于应用服务器，包括步骤：

[0049] S1、接收资源下载请求(URL)。

[0050] S2、比对所述资源下载请求(URL)是否授权，如果否，生成质疑标识(challenge)并发送包括所述质疑标识的目标信息至客户端。

[0051] 具体的，所述比对所述URL请求是否授权可以根据所述web.xml配置，比对所述URL请求是否授权。

[0052] S3、接收用户基于所述目标信息输入的Linux操作系统的用户名和密码信息。

[0053] Java Linux平台的Web应用是由许多Servlet、HTML页面、类和其他资源组成的集合，这些资源组成了一个运行在Web服务器上的完整应用程序。一个终端用户可以使用多种受支持的客户端验证方式或类型进行用户验证。

[0054] 本实施例采用Linux系统上的轻量级认证方式，其中，HTTP基本验证是被HTTP协议支持的验证机制，这种机制基于用户名和密码。Web服务器要求Web客户端对用户进行验证，作为请求的一部分，Web服务器将传递一个域，用户会在这个域中接受验证。Web客户端从用户获得用户名和密码，并且将它们传送到Web服务器，然后，Web服务器会在指定的域中验证用户。

[0055] 正如背景技术所述，常用的HTTP验证方式分为四种：basic、digest、form以及cert。其中，本实施例优先选用第一种和第二种验证方式：Basic验证以及Digest验证。

[0056] 其中，Basic和Digest的认证的处理流程相同，如图2所示，包括步骤：

[0057] S21，客户端请求受保护的资源。

[0058] S22，服务器检测到没有授权，则生成一个challenge返回给客户端。

[0059] S23，客户端根据challenge和相关信息计算出digest。

[0060] S24，附带步骤S23计算出的信息再次请求步骤S21中的资源。

[0061] S25,服务端根据已知的用户密码信息计算出digest并与步骤S24中请求的digest比较验证。

[0062] S26,服务端验证通过后返回资源给合法用户。

[0063] 需要说明的是,Basic验证以及Digest验证的不同点是步骤S23和步骤S25中计算digest的算法不同。其中,Basic是将密码直接base64编码(明文),而Digest是用MD5进行加密后传输。

[0064] 即,Basic是http的基础的认证方式,digest是basic的升级版。理论上digest更加安全。发明人发现:因为basic是明文传输密码信息,而digest是加密后传输。然而,digest默认用MD5对密码进行加密,虽然相比basic认证的明文传输更安全,但是加密算法本身MD5可以反推出原文,而且digest只是对认证信息的加密,后续的内容传输安全性得不到保障。因此,本方案在上述步骤的基础上,增加了步骤S4,采用PAM认证模块对数据进行安全性的认证。

[0065] 具体的,Basic验证是在HTTP协议进行通信的过程中,HTTP协议定义了基本认证过程以允许HTTP服务器对WEB浏览器进行用户身份证的方法,当一个客户端向HTTP服务器进行数据请求时,服务器会发送一个未授权的响应,在响应中带了Realm信息表示使用Basic认证,本实施例提供了一个Basic认证的具体实例,如下:

[0066] 浏览器接收到这个响应后会弹出一个框,输入用户名和密码。点取消表示取消认证,点确定会提交用户名、密码到服务器。提交的方式是在HTTP头中加入:WWW-Authorization:Basic XXXXXXX。

[0067] 其中,Basic后面是用户名、密码的BASE64编码。服务器端接收后,提起Http head中的信息,验证用户和密码,通过则允许访问,返回客户端所需要的数据;不通过禁止访问,返回错误代码或重新要求客户端提供用户名及密码。

[0068] S4、根据预设的配置文件(web.xml)配置调用预设验证模块(PAM验证模块),通过所述预设验证模块验证所述用户名和密码信息。

[0069] 具体的,本实施例可以通过在Linux操作系统中,预先配置各客户端的用户名和密码以及PAM验证模块,然后建立所述用户名和密码与所述PAM验证模块的关联关系。当所述通过所述PAM验证模块验证所述用户名和密码信息时,实际上是查找所述用户基于所述目标信息输入的Linux操作系统的用户名和密码信息是否属于所述预先配置的各客户端的用户名和密码,如果属于,则确定验证通过。

[0070] S5、如果验证通过,发送与所述资源下载请求对应的资源至所述客户端。

[0071] 正如上文所述,发明人发现HTTP基本验证并不安全,因此,本方案在上述步骤的基础上,增加了步骤S4,采用PAM认证模块对数据进行安全性的认证。

[0072] 其中,Linux-PAM是PLUGGABLE AUTHENTICATION MODULES的缩写,它是一套共享库。它可以让系统管理员选择应用程序怎样去认证用户,而不需要知道应用程序的内部的实现细节,也不需要重新编译代码。

[0073] 而传统认证方式是,一般的应用程序,在要去认证用户时,它们会将某种特定的认证方式硬编码到程序内部。比如,传统的用户登录程序,它先获得用户的用户名和密码,然后,将用户输入的密码进行计算得到密文,最后,将得到的密文和/etc/shadow文件中的该用户所在行的第二个字段进行比较,如果相同,则认证通过,否则,认证失败。这种认证过程

的缺点在于无法方便地去改变登录程序所使用的这种认证方式。

[0074] 随着计算机技术和安全技术的发展，越来越多的旧的认证机制变得非常得脆弱；同时，传统认证机制下用户认证经常需要去改变应用程序的认证过程，也不便于认证管理的方便实现。

[0075] 如果某个应用程序使用了PAM，当它需要进行用户认证的时候，只需要把认证过程简单得交给PAM模块，然后，由PAM模块对用户进行认证，PAM再将认证的结果返回给应用程序。应用程序并不知道PAM到底使用了什么方法对用户进行了认证，这个由系统管理员来决定。管理员可以使用，simple trust(pam_permit)来进行认证，也可以使用非常复杂的认证方式，如网膜认证、声纹认证等。节省了用户调整认证过程的需要时间，提高了认证效率。

[0076] 具体的，在本实施例中，PAM采取了分层设计思想，如图3所示，实现了模块的可插入性和易用性。PAM让各鉴别模块从应用程序中独立出来，通过PAM API作为两者联系的纽带，应用程序就可以根据需要(通过配置文件配置)灵活地在其中"插入"所需鉴别功能模块，从而真正实现了"鉴别功能，随需应变"。

[0077] 综上，结合图4，为本实施例提供的一种用户认证方法所应用的产品流程图，首先，开始用户认证，用户侧提交URL请求。然后，Web应用服务器在web.xml配置了通过HTTP BASIC作为用户登录验证机制。其中，SG-APS中间件根据web.xml中的配置，返回Http.response应答。

[0078] 具体验证的实现流程如下：

[0079] (1)在浏览器请求:http://localhost:8080/index.html。

[0080] (2)SG-APS服务器根据web.xml配置，服务器返回401(unauthentication)代码，并附带一个包含challenge的头，格式如下:WWW-Authenticate Basic realm="Admin All"。

[0081] (3)浏览器接收的返回的请求，弹出“用户名和密码输入框”。

[0082] (4)用户在浏览器的输入框中输入Linux操作系统的用户ing和密码，浏览器将：(username:password)编码后的信息添加到请求头中提交到服务器端。

[0083] 之后，用户在浏览器中输入Linux操作系统的用户名密码提交到服务器。然后，SG-APS根据web.xml配置调用PAM验证模块。具体验证实现流程如下：

[0084] (1)在Linux操作系统中，添加web应用的用户和密码。

[0085] (2)在Linux操作系统中，添加Pam的配置文件，在该配置文件中定义一个pam的认证过程。

[0086] (3)将SG-APS委托PAM认证的代码打成jar包，添加到SG-APS应用服务器的“modules”目录中。

[0087] (4)在SG-APS配置文件中，定义通过pam认证的安全域，如为:"pamreaml"，该安全域的认证代码包路径为在“moudules”目录中部署的认证代码包。

[0088] (5)在web.xml文件中，将pam域“pamrealm”配置到web.xml中。

[0089] (6)SG-APS中间件接收到用户密码验证请求后，根据web.xml中“pamrealm”配置定位到SG-APS中的“pamrealm"配置，根据配置SG-APS调用Pam验证代码，pam验证代码委托Linux Pam模块验证用户名和密码。

[0090] 再之后，验证通过后，根据web.xml验证资源权限。具体的，根据Servlet规范定义一个用户角色，然后，配置用户角色访问管控的资源pam，最后，创建用户，绑定角色和资源。

[0091] 最后，权限验证通过，认证过程结束，用户读取URL资源。

[0092] 可见，本实施例采用HTTP和PAM组合认证，多个安全认证约束通过单个安全约束合并来实现。即，将Web应用的认证管理，委托给Linux操作系统的Pam验证模块，通过pam模块的可定制性，提升用户登录和密码验证的灵活度。且改进用户认证过程，提升用户认证效率

[0093] 除此，本实施例还提供了一种用户认证装置，应用于应用服务器，如图5所示，所述用户认证装置包括：

[0094] 第一接收模块101，用于接收URL请求；

[0095] 比对模块102，用于比对所述URL请求是否授权，如果否，生成质疑标识(challenge)并发送包括所述质疑标识的目标信息至客户端；

[0096] 第二接收模块103，用于接收用户基于所述目标信息输入的Linux操作系统的用户名和密码信息；

[0097] 验证模块104，用于根据web.xml配置调用PAM验证模块，通过所述PAM验证模块验证所述用户名和密码信息；

[0098] 发送模块105，用于如果验证通过，发送与所述URL请求对应的资源至所述客户端。

[0099] 优选的，所述比对模块包括：比对单元，用于根据所述web.xml配置，比对所述URL请求是否授权。所述验证模块包括：配置单元以及建立单元，其中，配置单元用于在Linux操作系统中，预先配置各客户端的用户名和密码以及PAM验证模块；建立单元用于建立所述用户名和密码与所述PAM验证模块的关联关系。

[0100] 可选的，所述验证模块还包括：

[0101] 查找单元，用于查找所述用户基于所述目标信息输入的Linux操作系统的用户名和密码信息是否属于所述预先配置的各客户端的用户名和密码，如果属于，则确定验证通过。

[0102] 需要说明的是，本实施例提供的用户认证装置的工作原理请参见上述用户认证方法的工作原理，在此不重复叙述。

[0103] 除此，本实施例还提供了一种用户认证系统，包括上述的任意一项所述用户认证装置。

[0104] 综上，本发明实施例提供了一种用户认证方法，通过接收资源下载请求；比对所述资源下载请求是否授权，如果否，生成质疑标识并发送包括所述质疑标识的目标信息至客户端；接收用户基于所述目标信息输入的Linux操作系统的用户名和密码信息；根据预设的配置文件配置调用预设验证模块，通过所述预设验证模块验证所述用户名和密码信息；如果验证通过，发送与所述资源下载请求对应的资源至所述客户端。

[0105] 即，基于Http中的basic和Digest认证方法，结合Servlet规范中的资源授权管理，将web应用的用户管理和登录认证，委托给Linux系统用户管理和Pam认证模块；web应用不提供用户管理和认证的功能。从而提升用户访问的安全性，同时简化web客户端登录验证的过程，提升验证效率。

[0106] 最后，还需要说明的是，在本文中，诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来，而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那

些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

[0107] 本说明书中各个实施例采用递进的方式描述，每个实施例重点说明的都是与其他实施例的不同之处，各个实施例之间相同相似部分互相参见即可。

[0108] 对所公开的实施例的上述说明，使本领域专业技术人员能够实现或使用本申请。对这些实施例的多种修改对本领域的专业技术人员来说将是显而易见的，本文中所定义的一般原理可以在不脱离本申请的精神或范围的情况下，在其它实施例中实现。因此，本申请将不会被限制于本文所示的这些实施例，而是要符合与本文所公开的原理和新颖特点相一致的最宽的范围。

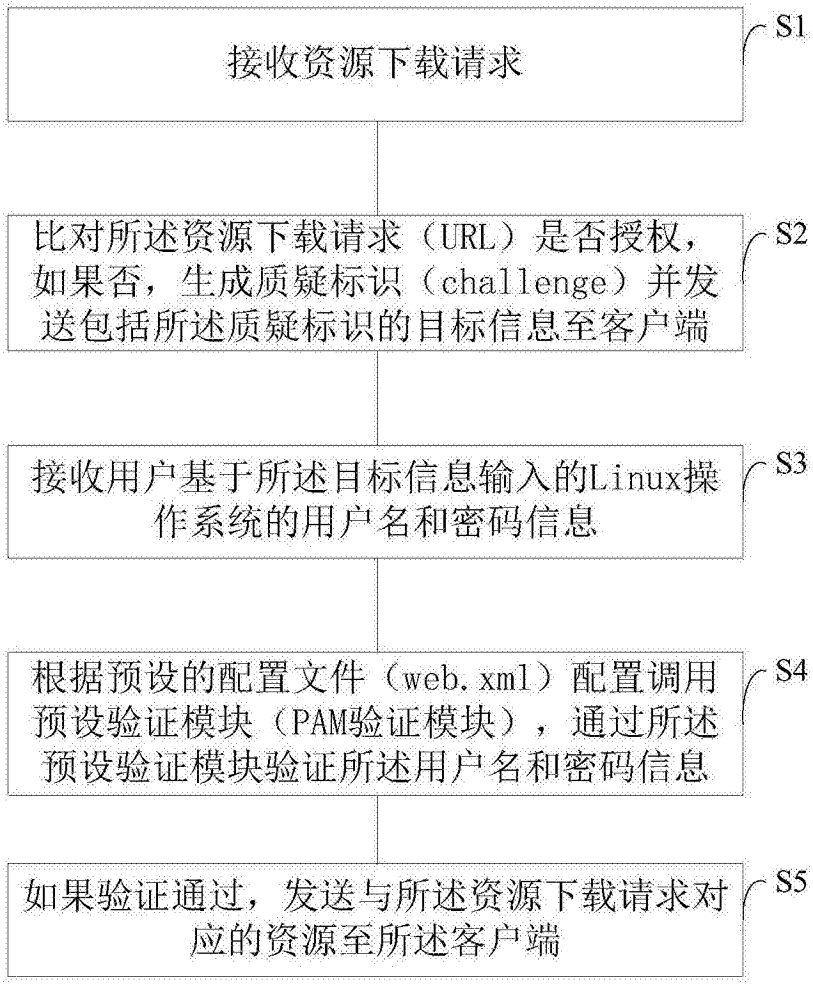

接收资源下载请求
S1
比对所述资源下载请求（URL）是否授权，如果否，生成质疑标识（challenge）并发送包括所述质疑标识的目标信息至客户端
S2
接收用户基于所述目标信息输入的Linux操作系统的用户名和密码信息
S3
根据预设的配置文件（web.xml）配置调用预设验证模块（PAM验证模块），通过所述预设验证模块验证所述用户名和密码信息
S4
如果验证通过，发送与所述资源下载请求对应的资源至所述客户端
S5

图1

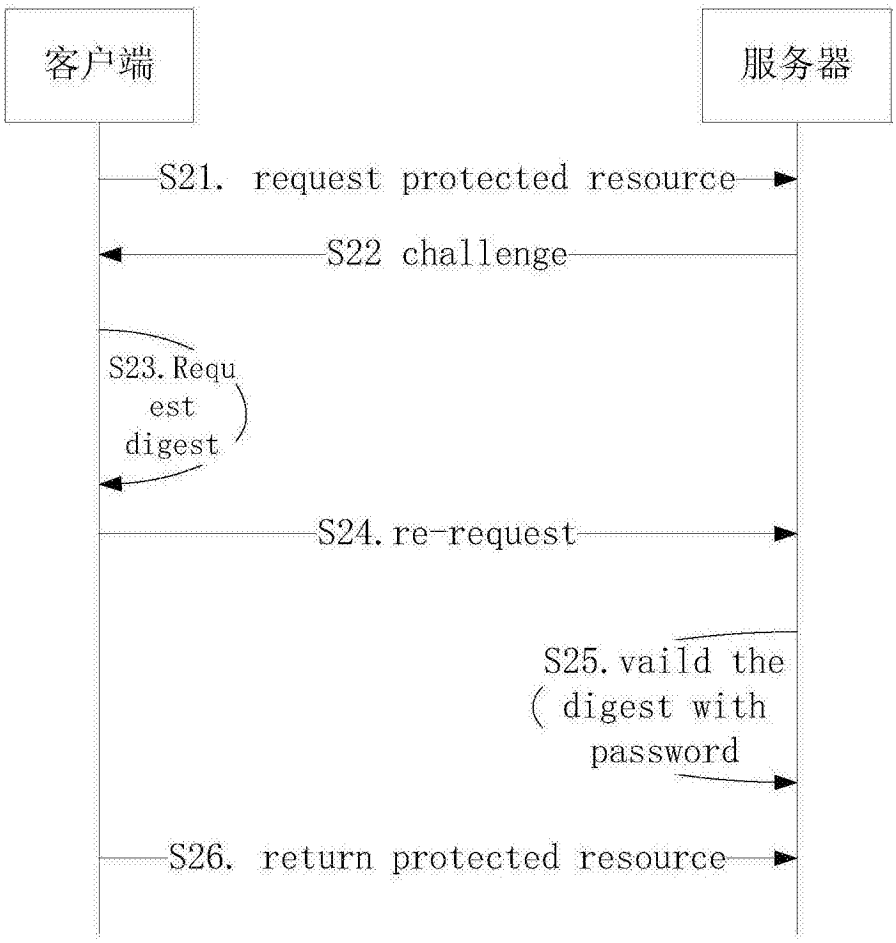

客户端
服务器
S21. request protected resource
S22 challenge
S23.Requ
est
digest
S24.re-request
S25.vaild the
digest with
password
S26. return protected resource

图2

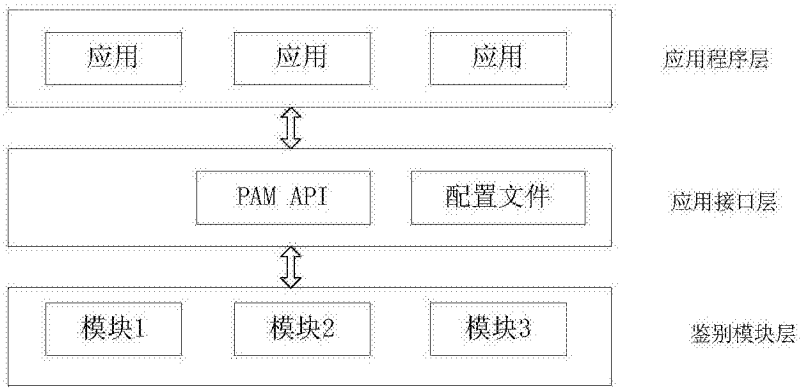

应用
应用
应用
应用程序层
PAM API
配置文件
应用接口层
模块1
模块2
模块3
鉴别模块层

图3

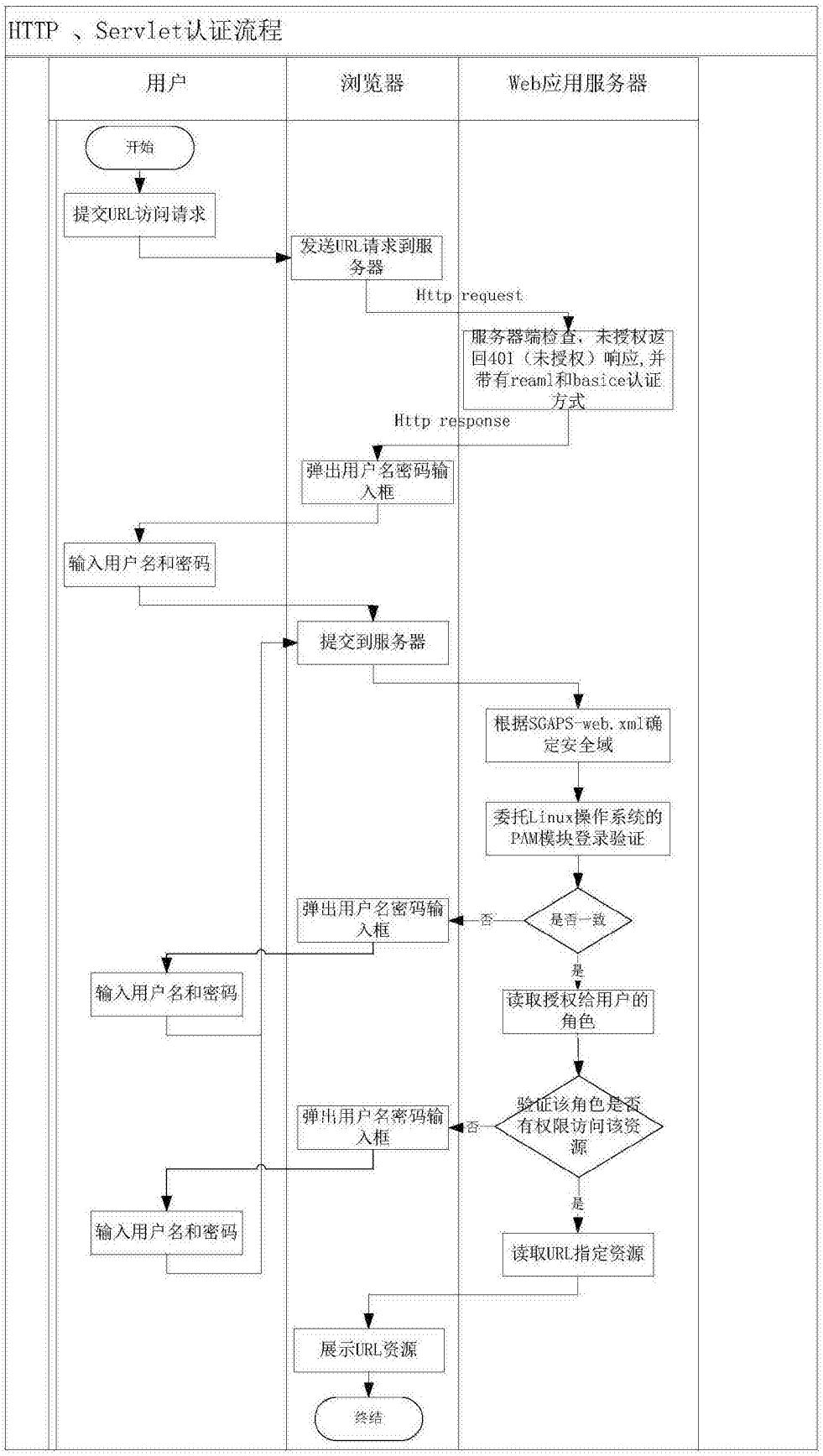

HTTP 、Servlet认证流程
用户
浏览器
Web应用服务器
开始
提交URL访问请求
发送URL请求到服务器
Http request
服务器端检查，未授权返回401（未授权）响应,并带有reaml和basice认证方式
Http response
弹出用户名密码输入框
输入用户名和密码
提交到服务器
根据SGAPS-web.xml确定安全域
委托Linux操作系统的PAM模块登录验证
是否一致
否
是
弹出用户名密码输入框
输入用户名和密码
读取授权给用户的角色
验证该角色是否有权限访问该资源
否
是
弹出用户名密码输入框
输入用户名和密码
读取URL指定资源
展示URL资源
终结

图4

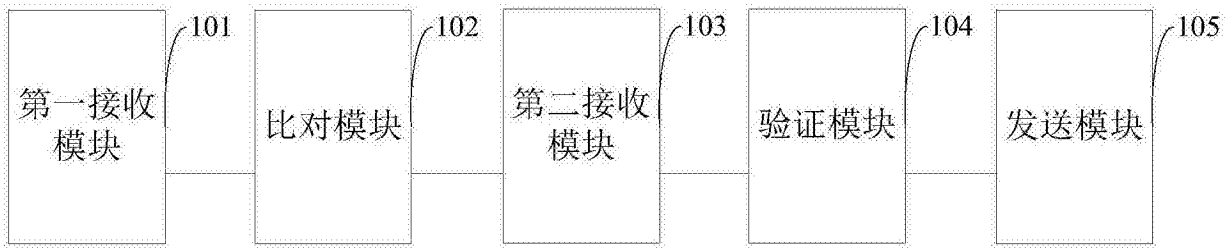


图5