(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 105282175 B
(45)授权公告日 2018.05.18

(21)申请号 201510778846.2

(22)申请日 2015.11.13

(65)同一申请的已公布的文献号
申请公布号 CN 105282175 A

(43)申请公布日 2016.01.27

(73)专利权人 上海斐讯数据通信技术有限公司
地址 201616 上海市松江区思贤路3666号

(72)发明人 李智荣

(74)专利代理机构 杭州千克知识产权代理有限公司 33246
代理人 周希良

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

(56)对比文件
CN 101370009 A,2009.02.18,
CN 103685242 A,2014.03.26,
尹刚.《基于SSH与代理的数据安全传输机制的研究》.《电子科学》.2009,正文第32页和第43页.

审查员 马旗超

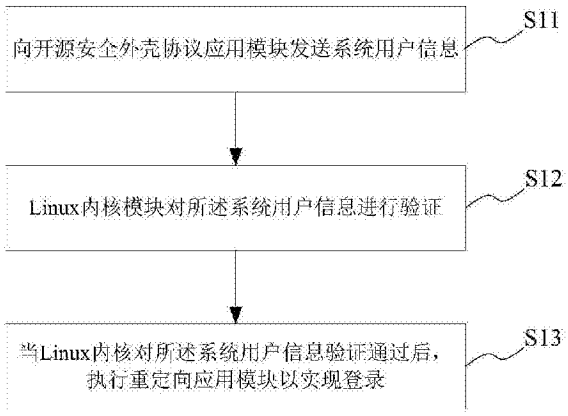
权利要求书1页 说明书6页 附图1页

(54)发明名称

基于开源安全外壳协议的登录方法及登录系统

(57)摘要

本发明提供一种基于开源安全外壳协议的登录方法及登录系统。所述基于开源安全外壳协议的登录方法包括以下步骤:向开源安全外壳协议应用模块发送系统用户信息;Linux内核模块对所述系统用户信息进行验证;当Linux内核对所述系统用户信息验证通过后,执行重定向应用模块以实现登录。本发明的登录方法及登录系统,通过重定向应用模块进行两次重定向,在不需要修改开源SSH源码的基础上实现SSH用户的登录,因此,大大降低了后续升级或者运营维护的成本。



CN 105282175 B

1.一种基于开源安全外壳协议的登录方法，其特征在于，所述基于开源安全外壳协议的登录方法包括以下步骤：

向开源安全外壳协议应用模块发送系统用户信息；

Linux内核模块对所述系统用户信息进行验证；

当Linux内核对所述系统用户信息验证通过后，执行重定向应用模块以实现登录；

所述执行重定向应用模块以实现登录的步骤包括：

在所述重定向应用模块与Linux内核之间建立重定向SSH连接；

在所述重定向应用模块与预制命令行界面之间建立TCP连接以实现登录。

2.根据权利要求1所述的基于开源安全外壳协议的登录方法，其特征在于：在所述重定向应用模块与预制命令行界面之间建立长TCP连接。

3.根据权利要求1所述的基于开源安全外壳协议的登录方法，其特征在于，还包括：在登录Linux系统后，向Linux内核中添加系统用户信息，所述系统用户信息包括：用户名和密码。

4.根据权利要求3所述的基于开源安全外壳协议的登录方法，其特征在于，所述向Linux内核中添加系统用户信息的步骤包括：Linux内核将所述系统用户信息保存在/etc/shadow和/etc/passwd文件夹中。

5.根据权利要求4所述的基于开源安全外壳协议的登录方法，其特征在于，Linux内核将所述系统用户信息保存在/etc/passwd文件夹中时，在所述/etc/passwd文件夹中指定系统用户信息验证通过后的执行程序为所述重定向应用模块。

6.一种基于开源安全外壳协议的登录系统，其特征在于，所述基于开源安全外壳协议的登录系统包括：

开源安全外壳协议应用模块，用于接收系统用户信息；

Linux内核，用于对所述系统用户信息进行验证；

重定向应用模块，用于当Linux系统对所述系统用户信息验证通过后，执行所述重定向应用模块以实现登录；

所述重定向应用模块包括：

第一重定向单元，用于在所述重定向应用模块与Linux系统内核之间建立重定向SSH连接；

第二重定向单元，用于在所述重定向应用模块与预制命令行界面之间建立TCP连接以实现登录。

7.根据权利要求6所述的基于开源安全外壳协议的登录系统，其特征在于，所述第二重定向单元在所述重定向应用模块与预制命令行界面之间建立长TCP连接。

8.根据权利要求6所述的基于开源安全外壳协议的登录系统，其特征在于，还包括：用户信息添加模块，用于在登录Linux系统后，向Linux内核中添加系统用户信息，所述系统用户信息包括：用户名和密码。

基于开源安全外壳协议的登录方法及登录系统

技术领域

[0001] 本发明涉及通信技术领域，特别是涉及一种基于开源安全外壳协议的登录方法及登录系统。

背景技术

[0002] 安全外壳协议(Secure Shell，SSH)是目前较可靠，专为远程登录会话和其他网络服务提供安全性的协议。利用SSH协议不仅可以有效防止远程管理过程中的信息泄露问题，还可以压缩传输的数据，加快传输的速度。SSH有很多功能，它既可以代替Telnet，又可以为FTP、PoP、甚至为PPP提供一个安全的“通道”。

[0003] 诸如交换机等网络设备系统都有自己定制的命令行界面(command-line interface，CLI)配置界面。用户通过SSH登录时进入的是嵌入式设备系统定制的CLI配置界面，而不是linux系统的shell界面。

[0004] 目前国内，除少数几家研发实力较强的公司会自己实现SSH协议外，大部分的网络设备研发厂商都是基于开源SSH软件(如OpenSSH)，针对自身系统做适配修改开源软件的源码来实现SSH用户认证登陆功能。

[0005] 然而，若公司自己实现SSH协议的开发，不使用开源软件，这个研发成功高，仅适合少数有一定研发实力的公司。而众多公司利用开源SSH软件(如OpenSSH)，针对自身系统做适配修改开源软件的源码来实现SSH用户认证登陆功能。但是，目前因为要修改开源软件源码，需要对开源软件的源码进行学习，后续升级开源软件版本或者更换开源软件、系统维护人员的成本高。

[0006] 因此，如何在不增加成本的基础上实现SSH用户的登录功能就成为本领域技术人员亟待解决的问题之一。

发明内容

[0007] 鉴于以上所述现有技术的缺点，本发明的目的在于提供一种基于开源安全外壳协议的登录方法及登录系统，用于解决现有技术中实现SSH用户登录功能成本较高的问题。

[0008] 为实现上述目的及其他相关目的，本发明提供一种基于开源安全外壳协议的登录方法，所述基于开源安全外壳协议的登录方法包括以下步骤：向开源安全外壳协议应用模块发送系统用户信息；Linux内核模块对所述系统用户信息进行验证；当Linux内核对所述系统用户信息验证通过后，执行重定向应用模块以实现登录。

[0009] 于本发明的一实施例中，所述执行重定向应用模块以实现登录的步骤包括：在所述重定向应用模块与Linux内核之间建立重定向SSH连接；在所述重定向应用模块与预制命令行界面之间建立TCP连接以实现登录。

[0010] 于本发明的一实施例中，在所述重定向应用模块与预制命令行界面之间建立长TCP连接。

[0011] 于本发明的一实施例中，所述基于开源安全外壳协议的登录方法还包括：在登录

Linux系统后，向Linux内核中添加系统用户信息，所述系统用户信息包括：用户名和密码。

[0012] 于本发明的一实施例中，所述向Linux内核中添加系统用户信息的步骤包括：Linux内核将所述系统用户信息保存在/etc/shadow和/etc/passwd文件夹中。

[0013] 于本发明的一实施例中，Linux内核将所述系统用户信息保存在/etc/passwd文件夹中时，在所述/etc/passwd文件夹中指定系统用户信息验证通过后的执行程序为所述重定向应用模块。

[0014] 本发明提供一种基于开源安全外壳协议的登录系统，所述基于开源安全外壳协议的登录系统包括：开源安全外壳协议应用模块，用于接收系统用户信息；Linux内核，用于对所述系统用户信息进行验证；重定向应用模块，用于当Linux系统对所述系统用户信息验证通过后，执行所述重定向应用模块以实现登录。

[0015] 于本发明的一实施例中，所述重定向应用模块包括：第一重定向单元，用于在所述重定向应用模块与Linux系统内核之间建立重定向SSH连接；第二重定向单元，用于在所述重定向应用模块与预制命令行界面之间建立TCP连接以实现登录。

[0016] 于本发明的一实施例中，所述第二重定向单元在所述所述重定向应用模块与预制命令行界面之间建立长TCP连接。

[0017] 于本发明的一实施例中，所述基于开源安全外壳协议的登录系统还包括：用户信息添加模块，用于在登录Linux系统后，向Linux内核中添加系统用户信息，所述系统用户信息包括：用户名和密码。

[0018] 如上所述，本发明的基于开源安全外壳协议的登录方法及登录系统，具有以下有益效果：

[0019] 本发明的基于开源安全外壳协议的登录方法，在Linux内核对系统用户信息验证通过后，通过重定向应用模块实现登录，不需要修改开源软件的源码，因此，大大降低了后续升级或者系统维护的成本。

附图说明

[0020] 图1显示为本发明基于开源安全外壳协议的登录方法的于一实施例中的流程示意图。

[0021] 图2显示为本发明基于开源安全外壳协议的登录系统的于一实施例中的结构示意图。

[0022] 元件标号说明

[0023] 2 基于开源安全外壳协议的登录系统

[0024] 21 开源安全外壳协议应用模块

[0025] 22 Linux内核

[0026] 23 重定向应用模块

[0027] S11～S13 步骤

具体实施方式

[0028] 以下通过特定的具体实例说明本发明的实施方式，本领域技术人员可由本说明书所揭露的内容轻易地了解本发明的其他优点与功效。本发明还可以通过另外不同的具体实

施方式加以实施或应用，本说明书中的各项细节也可以基于不同观点与应用，在没有背离本发明的精神下进行各种修饰或改变。需说明的是，在不冲突的情况下，以下实施例及实施例中的特征可以相互组合。

[0029] 需要说明的是，以下实施例中所提供的图示仅以示意方式说明本发明的基本构想，遂图式中仅显示与本发明中有关的组件而非按照实际实施时的组件数目、形状及尺寸绘制，其实际实施时各组件的型态、数量及比例可为一种随意的改变，且其组件布局型态也可能更为复杂。

[0030] 正如背景技术中所述的，目前通过SSH进行登录时，要么自己开发SSH协议，要么利用开源SSH软件来实现，但是无论哪种方式，开发的成本都很高，应用的范围非常受限。

[0031] 请参阅图1，本发明提供一种基于开源安全外壳协议的登录方法，所述基于开源安全外壳协议的登录方法包括以下步骤：

[0032] S11，向开源安全外壳协议应用模块发送系统用户信息；

[0033] S12，Linux内核模块对所述系统用户信息进行验证；

[0034] S13，当Linux内核对所述系统用户信息验证通过后，执行重定向应用模块以实现登录。

[0035] 具体地，用户通过SSH客户端软件(如SecureCRT)请求与OpenSSH建立SSH连接，在此过程中向OpenSSH发送SSH用户名和密码。

[0036] OpenSSH向linux查询用户名是否存在及用户名的密码是否正确，如果不正确，拒绝用户的连接请求，否则进行下一步处理。具体地，linux系统内核(linux kernel)对用户名及密码进行验证。

[0037] OpenSSH在得知用户认证通过后，执行RdirectIO应用程序(该应用程序为预开发的重定向应用模块)。OpenSSH在执行RdirectIO时进行重定向，从而实现SSH用户的登录。RdirectIO是本实施例中需要实现的一个应用程序，其功能参考后面的描述。

[0038] 在本实施例中，所述执行重定向应用模块以实现登录的步骤包括：在所述重定向应用模块与Linux内核之间建立重定向SSH连接；在所述重定向应用模块与预制命令行界面之间建立TCP连接以实现登录。具体地，在所述重定向应用模块与预制命令行界面之间建立长TCP连接。

[0039] 在实际应用中，OpenSSH在执行RdirectIO时，创建一个管道(pipe)。通过这个管道，RdirectIO的标准IO会重定向到ssh连接。同时，RdirectIO与系统定制CLI建立一条tcp连接，把RdirectIO的标准IO重定向到这条tcp连接。

[0040] 也就是说，RdirectIO的功能是将系统定制CLI的输入重定向到RdirectIO标准输出；同样，将RdirectIO的标准输入重定向到系统定制CLI的输出，这样就可以通过在系统定制CLI与RdirectIO之间建立TCP长连接来实现这里的IO重定向功能。

[0041] 具体地，所述RdirectIO的实现代码为：

```
While(1)
{
    testfds=readfds;
    result=select(FD_SETSIZE,&testfds,(fd_set *)0, (fd_set *)0,(struct timeval *)0);
    for (fd=0;   fd<FD_SETSIZE;   fd++)
    {
        if (FD_ISSET(fd, &testfds))
        {
[0042]              if(fd==0)
                    {
                      memset(buffer_read1, 0, sizeof(buffer_read1));
                      read(0, buffer_read1, sizeof(buffer_read1));
                      writer(sockfd, buffer_read1, sizeof(buffer_read1));
                    }
                    if(fd==sockfd)
                    {
                      memset(buffer_read2, 0, sizeof(buffer_read2));
                      read(sockfd, buffer_read2, sizeof(buffer_read2));
                      writer(0, buffer_read2, sizeof(buffer_read2));
                    }
[0043]          }
    }
}
```

[0044] 经过上述两步IO重定向(Input&Output)处理,SSH用户通过SSH认证登录系统后,在SSH客户端(如SecureCRT)上输入的数据经过两次重定向到达系统CLI界面;反之,系统定制CLI输入的数据也经过两次重定向到达SSH客户端。

[0045] 虽然实际上用户在SSH连接上输入的数据是先经过OpenSSH解密,解密后的数据经过两次重定向才到达系统定制CLI。但是,在系统使用用户的感知上,用户是直接进入了系统定制CLI界面。在系统使用者的感知上,用户是直接登录到了系统定制CLI界面。

[0046] 为了实现系统用户信息的验证,在本实施例中,所述基于开源安全外壳协议的登录方法还包括:在登录Linux系统后,向Linux内核中添加系统用户信息,所述系统用户信息包括:用户名和密码。

[0047] 具体地,所述向Linux内核中添加系统用户信息的步骤包括:Linux内核将所述系统用户信息保存在/etc/shadow和/etc/passwd文件夹中。同时,Linux内核将所述系统用户信息保存在/etc/passwd文件夹中时,在所述/etc/passwd文件夹中指定系统用户信息验证

通过后的执行程序为所述重定向应用模块。

[0048] 在实际应用中，开源OpenSSH的用户认证是向linux kernel(Linux内核)校验用户名及其密码的正确性。用户通过串口登录系统，在向系统添加SSH用户时同时向linux kernel添加相同用户名及密码的系统用户。Linux系统将系统用户及其密码保存在“/etc/shadow”和“/etc/passwd”两个文件中。

[0049] 例如，用户为交换机增加了名为admin的用户，交换机系统在将用户添加到自身的配置文件中时，向linux kernel增加一个名为admin的系统用户。系统用户增加成功后在“/etc/shadow”和“/etc/passwd”两个文件中各增加了一条记录。在往“/etc/passwd”文件添加admin用户的记录时，最后一个“:”后面填写的内容是OpenSSH在确定用户名及密码都正确后要执行的应用程序，这里填写的是RdirectIO应用程序。

[0050] 具体的，shadow文件的实现代码为：

[0051] #cat/etc/shadow

[0052] Root:5Gg.mRfhg Iz4g:10925:0:99999:7:::

[0053] Sshd:kVsqISda3pms.:0:0:99999:7:::

[0054] Admin:IjU5ugwlS2HnY:15357:0:99999:7:::

[0055] #

[0056] passwd文件的实现代码为：

[0057] #cat/etc/passwd

[0058] Root:x:0:0:root:/root?/bin/sh

[0059] Sshd:x:1000:1000:sshd privsep:/var/empty:/bin/ssh

[0060] Admin:x:1001:1001:Linux User,,,:/home:/mnt/app/RdirectIO

[0061] #

[0062] OpenSSH在执行RdirectIO应用程序时会在OpenSSH与RdirectIO之间创建一条管道(pipe)。用户在SSH客户端(如SecureCRT)上输入的数据经过OpenSSH解密后通过这条管道直接到达RdirectIO。反过来，RdirectIO的标准输出也通过这条管道经OpenSSH加密后输出到SSH客户端。这里的处理由OpenSSH开源软件完成，不需要修改OpenSSH源码。

[0063] 本实施例的基于开源安全外壳协议的登录方法，通过预制的重定向应用模块完成两次重定向，从而实现SSH用户的登录，并且不需要开源SSH源码，因此，大大降低了后续升级或者运营维护的成本。

[0064] 本发明提供一种基于开源安全外壳协议的登录系统，参考图2，所述基于开源安全外壳协议的登录系统包括：

[0065] 开源安全外壳协议应用模块21，用于接收系统用户信息；

[0066] Linux内核22，用于对所述系统用户信息进行验证；

[0067] 重定向应用模块23，用于当Linux系统对所述系统用户信息验证通过后，执行所述重定向应用模块以实现登录。

[0068] 本实施例中，所述重定向应用模块23包括：第一重定向单元和第二重定向单元(图中未示出)，所述第一重定向单元用于在所述重定向应用模块与Linux系统内核之间建立重定向SSH连接；所述第二重定向单元用于在所述重定向应用模块与预制命令行界面之间建立TCP连接以实现登录。具体地，所述第二重定向单元在所述所述重定向应用模块与预制命

令行界面之间建立长TCP连接。

[0069] 本实施例中，所述基于开源安全外壳协议的登录系统还包括：用户信息添加模块，用于在登录Linux系统后，向Linux内核中添加系统用户信息，所述系统用户信息包括：用户名和密码。

[0070] 本实施例的基于开源安全外壳协议的登录系统的具体实现过程可参考前述关于基于开源安全外壳协议的登录方法的详细描述，在此不再赘述。

[0071] 综上所述，本发明的基于开源安全外壳协议的登录方法及登录系统，利用预制的重定向模块实现两次重定向，从而实现SSH用户的登录，并且不需要修改开源SSH源码，从而大大降低了后续的升级及运营维护的成本。所以，本发明有效克服了现有技术中的种种缺点而具高度产业利用价值。

[0072] 上述实施例仅例示性说明本发明的原理及其功效，而非用于限制本发明。任何熟悉此技术的人士皆可在不违背本发明的精神及范畴下，对上述实施例进行修饰或改变。因此，举凡所属技术领域中具有通常知识者在未脱离本发明所揭示的精神与技术思想下所完成的一切等效修饰或改变，仍应由本发明的权利要求所涵盖。

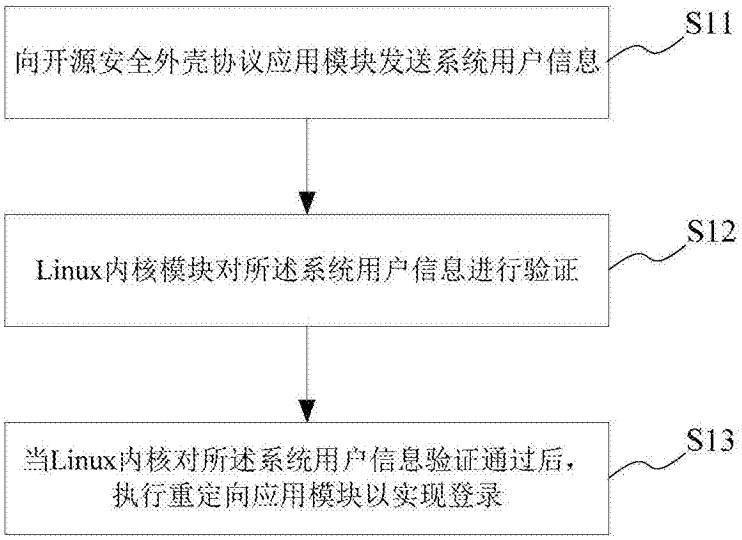

向开源安全外壳协议应用模块发送系统用户信息
S11
Linux内核模块对所述系统用户信息进行验证
S12
当Linux内核对所述系统用户信息验证通过后，
执行重定向应用模块以实现登录
S13

图1

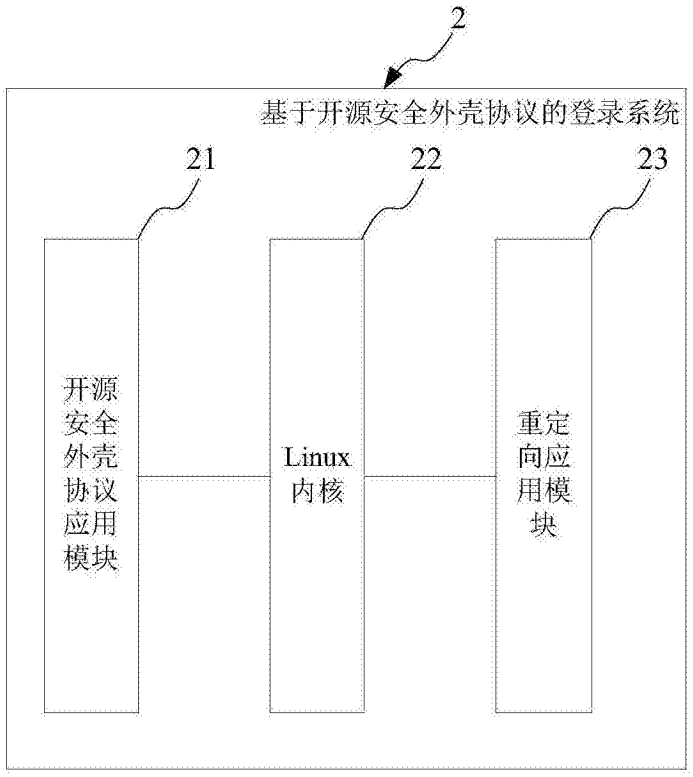

2
基于开源安全外壳协议的登录系统
21
22
23
开源安全外壳协议应用模块
Linux内核
重定向应用模块

图2