网络安全协议及分析
会话安全SSH

密码与网络安全系　刘虹

2025年春季学期

课程体系

第一章 概述

第二章 链路层扩展L2TP

第三章 IP层安全IPSec

第四章 传输层安全SSL和TLS

第五章 会话安全SSH

第六章 代理安全Socks

第七章 网管安全SNMPv3

第八章 认证协议Kerberos

第九章 应用安全

本章学习目标

- SSH功能及组成
- SSH数据类型及算法
- SSH传输协议
- SSH身份认证协议

提纲

SSH概述

- SSH（SecureShell，安全命令解释器）
  - 对数据进行加密处理，所以可以提供对账号、口令的机密性保护。
  - 提供身份认证和数据完整性保护功能。
  - 设计初衷是为了保障远程登录及交互式会话安全，但最终可以为FTP、SMTP等各种应用层协议提供安全屏障。

历史及现状

- SSH（Secure Shell，安全Shell）是基于TCP的应用层协议，能对数据进行加密处理，为保障远程登录及交互式会话安全而提出。
  - 1995年，Tatu Ylonen首次设计并实现SSH（SSH1版本）。Tatu Ylonen创立SSH通信安全公司，提供企业级的SSH安全方案和产品。
  - 1998年，SSH通信公司推出SSH2版本。
  - 2005年，SSH通信公司推出SSH G3版本。
  - 2006年，SSH2.0正式成为IETF标准，文档为RFC4250-4256。
  - 2008年，SecSh工作组公布两个草案。

功能及组成

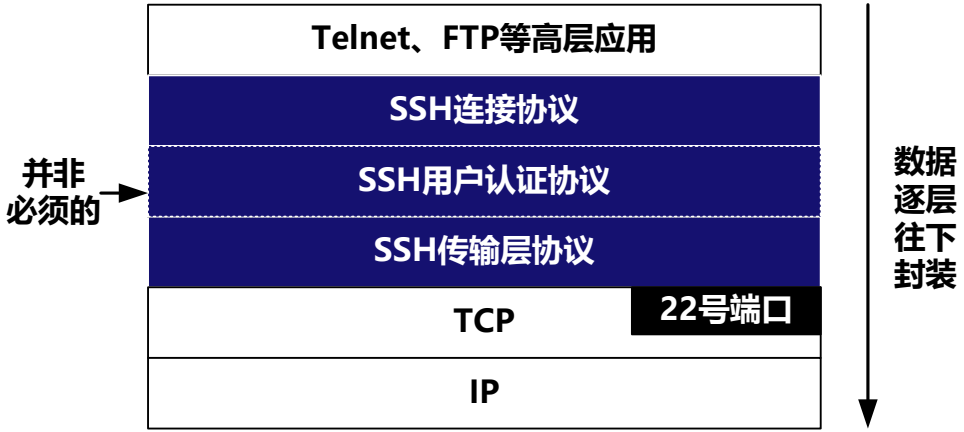
- SSH为应用层协议，基于TCP，使用端口22。
- SSH所提供服务：
  - 机密性
  - 完整性
  - 身份认证功能（必选的服务器认证，可选的客户端认证）
- SSH所包含内容：
  - 算法协商（包括加密、认证和压缩算法）
  - 密钥交换
  - 服务器身份认证、用户身份认证
  - 应用层数据封装处理等

功能及组成

- SSH协议组成：
  - **传输层协议**：
    - 协商和数据处理。位于SSH协议套件的最底层。位于SSH协议套件的最底层，为SSH其它协议和高层应用提供安全保护。(包含了服务器身份认证)
  - **用户认证协议**：
    - 规定了服务器认证客户端用户身份的流程和报文内容。
  - **连接协议**：
    - 将安全通道多路分解为多个逻辑通道，以便多个高层应用共享SSH提供的安全服务。

SSH的组成协议关系图

- SSH连接协议虽然位于用户认证协议之上，二者并不存在严格意义上的依赖关系，
  - 连接协议的报文封装在传输层协议报文中进行投递。
  - 在某些情况下，服务器必须在认证客户端用户身份后才能进行随后的通信操作。



SSH数据类型

- SSH规定7种数据类型来描述协议消息格式
  - 单字节整数
  - 布尔值
  - 32比特整数
  - 64比特整数
  - 字符串
  - 大整数
  - 列表

SSH方法及算法描述

- SSH用字符串描述算法，并给出了可用算法的标准字符串表示
  - “3des-cbc”表示使用CBC模式的3DES算法
  - ”blowfish-cbc”表示使用CBC模式的blowfish算法。
  - 如果同时使用两个算法，则用“列表”数据类型表示出来。
- SSH规定了厂商实现中自定义名字的统一格式

提纲

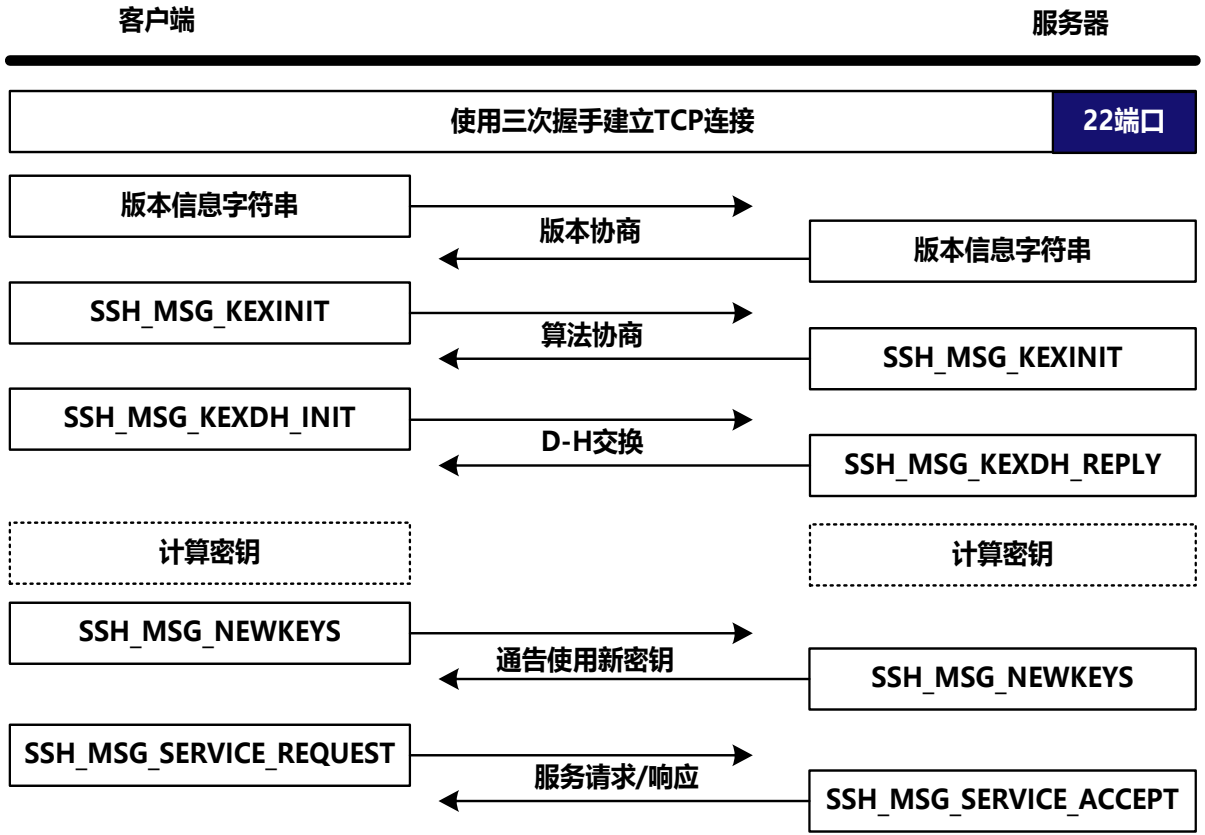
一、SSH概述

二、SSH传输协议

三、SSH身份认证协议

二 协议流程

- 版本交换
- 密钥交换
  - 算法协商
  - D-H交换
  - 密钥计算
- 服务请求/响应



二 SSH传输协议

- **（1）版本协商：协定最终使用的SSH版本**
  - 从服务器的角度看，如果它具备版本兼容功能，能与较低版本的客户端进行通信；
  - 从客户端的角度看，当自己的版本比服务器版本更新时，客户端应立刻终止当前连接。如果客户具备版本兼容功能，则应重新建立连接并进行版本通告。

二 SSH传输协议

（2）密钥交换：

- 算法协商
- D-H交换
- 计算密钥

SSH支持两种认证方法：

- 显式方法：报文包含数字签名。客户端必须获取服务器公钥（SSH也称为热键hot key)
- 隐式方法：报文包含利用预共享密钥计算的MAC。

二 SSH传输协议

- 2.1 算法协商：通信双方交互SSH_MSG_KEXINIT消息进行算法协商。
  - Cookie：为随机数，用来和其它参数共同生成密钥和会话ID，以防止重放攻击。
  - 密钥交换算法列表
  - 服务器热键算法列表
  - 客户端和服务器加密算法列表
  - 客户端和服务器MAC算法列表
  - 客户端和服务器压缩算法列表
  - 客户端和服务器语言列表
  - 标志。

二 SSH传输协议

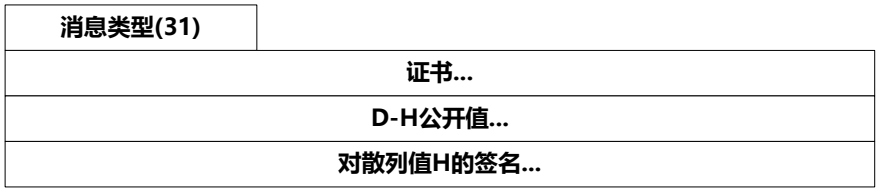
2.2 D-H交换

- 若通信双方协定使用SSH定义的密钥交换方法，则将开始D-H交换过程。
- 客户端首先向服务器发送SSH-MSG-KEXDH-INIT消息，其中包含了D-H交换公开值；
- 服务器则返回SSH-MSG-KEXDH-REPLY消息。

二 SSH传输协议

- SSH_MSG_KEXDH_REPLY消息完成以下功能：
  - **密钥交换**：双方交换D-H公开值
  - **服务器身份认证**：对散列值H的签名
  - **握手消息完整性**：散列值H的输入包含客户端和服务器的版本信息、SSH_MSG_KEXINIT消息（包含Cookie字段）、热键、双方的D-H公开值以及共享密钥。
  - 散列值H同时用于计算会话ID，可用来防止重放攻击。

SSH_MSG_KEXDH_REPLY消息格式



二 SSH传输协议

- **2.3 计算密钥**：SSH使用4个密钥：通信双方用来加密和认证的密钥（Ecs,Esc,Mcs,Msc），以及分组加密中使用的IV（IVcs,IVsc）
  - IVcs = HASH(K|H|"A"|会话ID)
  - IVsc = HASH(K|H|"B"|会话ID)
  - Ecs = HASH(K|H|"C"|会话ID)
  - Esc = HASH(K|H|"D"|会话ID)
  - Mcs = HASH(K|H|"E"会话ID)
  - Msc = HASH(K|H|"F"|会话ID)

K1=HASH(K|H|X|会话ID)

K2=HASH(K|H|K1)

K3=HASH(K|H|K1|K2)

… …

Key=K1 | K2 | K3

二 SSH传输协议

- **2.4 密钥再交换：**
  - 通信双方都可以发起密钥再交换协商，与首次协商过程相同。
  - 密钥交换完成后，通信双方交互SSH_MSG_NEWKEYS消息以通告对等端随后的通信流使用新的算法和密钥保护。

二 SSH传输协议

（3）服务请求/响应。

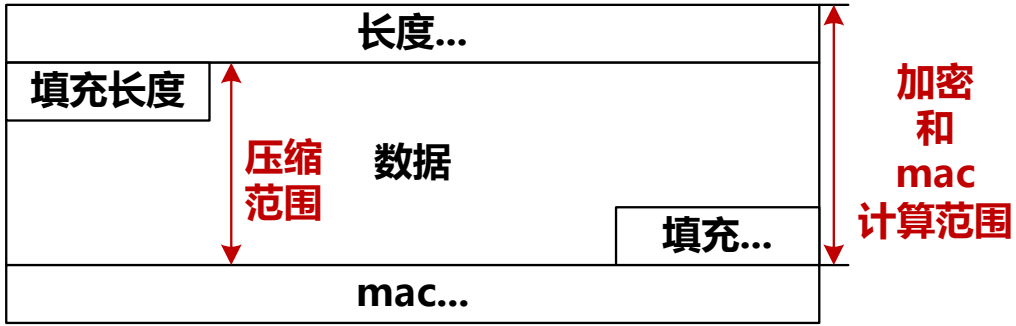
- **客户端发送**：SSH_MSG_SERVICE_REQUEST消息，其中包含了所请求的服务名称。
- **服务器发送**：

  如果同意提供客户端提出服务，则返回SSH_MSG_SERVICE_ACCEPT

  否则返回SSH_MSG_DISCONNECT
- **SSH定义两种服务**：ssh-userauth（用户认证）和ssh-connection（连接请求）

二 SSH传输协议

- **报文格式**：Mac=MAC(Mcs/Msc，序号｜未加密的报文)
  - MAC：消息验证码算法
  - 未加密的报文：除MAC以外的所有报文数据
- 加密算法应用于除MAC外的其他部分。
- 压缩算法应用于数据部分。



提纲

一、SSH概述

二、SSH传输协议

三、SSH身份认证协议

三 SSH身份认证协议

- 客户利用传输层协议向服务器提出**用户身份认证服务请求**，若服务器接受请求，则双方开始执行SSH身份认证协议。
  - 身份认证协议在传输层协议提供的安全通道上运行。(即身份认证协议的消息封装在传输层报文的数据区中)
  - 一次身份认证失败，客户端可再次提出认证请求，但重试的时间间隔和次数是有限制的。

三 身份认证过程

- 身份认证过程主要由三种消息完成：
  - 客户端所发出的认证请求
    - SSH_MSG_USERAUTH_REQUEST
  - 服务器应答
    - 服务器接受认证的应答：SSH_MSG_USERAUTH_SUCCESS
    - 服务器拒绝认证的应答：SSH_MSG_USERAUTH_FAILURE

三

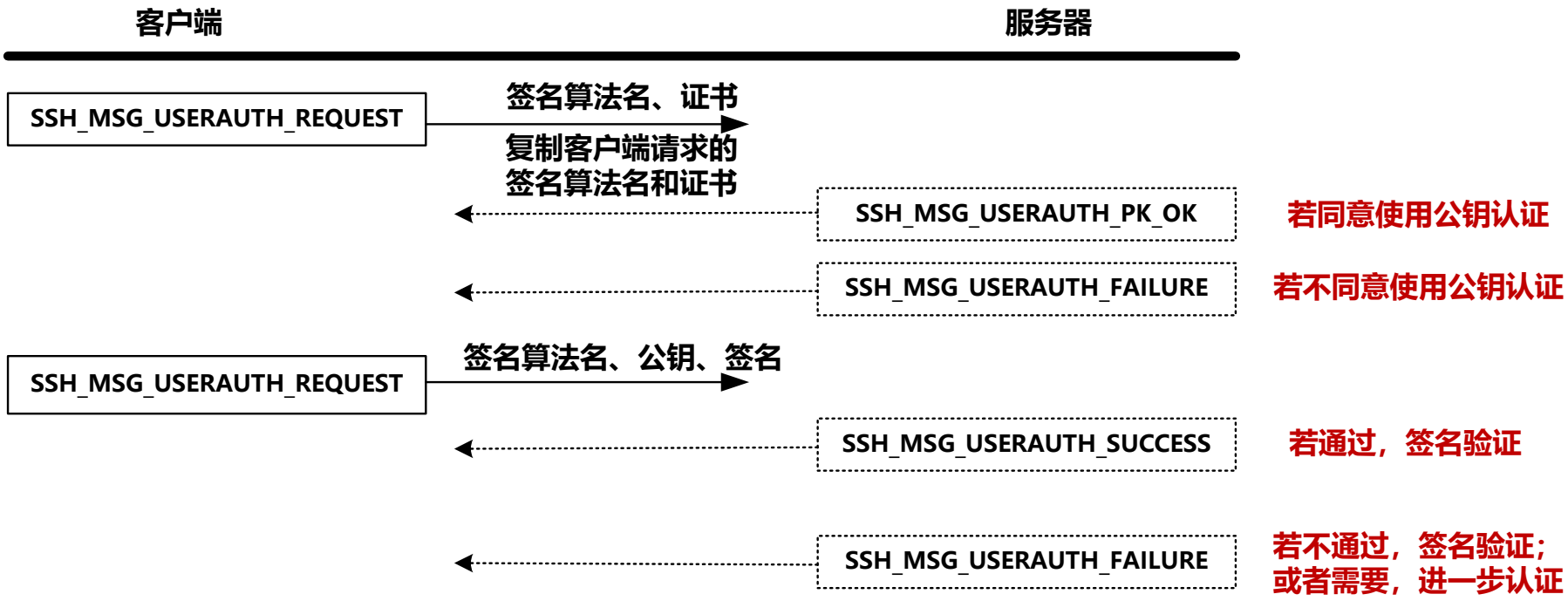
身份认证过程

- SSH支持4种用户身份认证方法：
  - Publickey（公钥认证）：客户端向服务器提供签名
  - Password（口令认证）：客户端向服务器提供口令
  - Hostbased （基于主机的认证）：由用户宿主机代理用户完成身份认证，用于主机有多个客户的情形。
  - None（不使用认证）：用于客户端身份不敏感和查询服务器支持的认证方法的情形下。

三 公钥认证方法

- 第一轮：通告签名算法和证书
  - SSH_MSG_USERAUTH_REQUEST消息：“认证方法”名字段被设置为“publickey”，“与认证方法相关的字段”，包含标志（设置为FLASE）签名算法名、证书
- 第二轮：身份认证
  - SSH_MSG_USERAUTH_REQUEST消息：“认证方法”名字段被设置为“publickey”，“与认证方法相关的字段”，包含标志（设置为TRUE）签名算法名、公钥和签名

三 公钥认证方法


客户端
服务器
SSH_MSG_USERAUTH_REQUEST
签名算法名、证书
复制客户端请求的
签名算法名和证书
SSH_MSG_USERAUTH_PK_OK
若同意使用公钥认证
SSH_MSG_USERAUTH_FAILURE
若不同意使用公钥认证
SSH_MSG_USERAUTH_REQUEST
签名算法名、公钥、签名
SSH_MSG_USERAUTH_SUCCESS
若通过，签名验证
SSH_MSG_USERAUTH_FAILURE
若不通过，签名验证；
或者需要，进一步认证

三 口令认证方法

- 客户端发送的请求消息包含“password”认证方法、标志字段设置为FALSE、**口令字符串**。
- 服务器的响应可能为：
  - 成功
  - 失败
  - 若口令过期：发送SSH_MSG_USERAUTH_PASSWD_CHANGEREQ
    - 客户端再次发出请求消息：其中标志字段设置为TRUE，包含原口令和新口令
    - 服务器响应

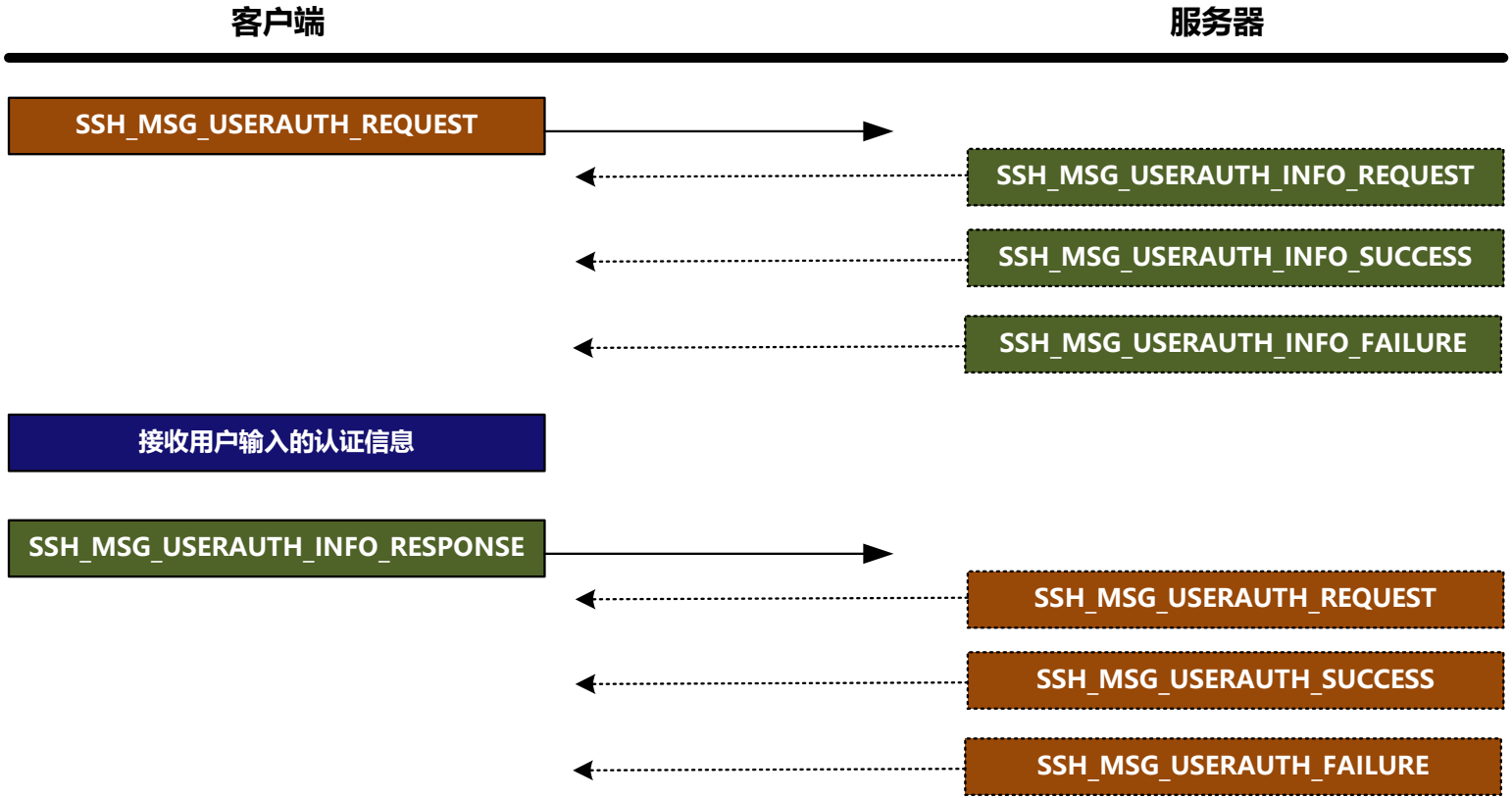
三 基于主机认证方法

- 使用数字签名验证身份，流程与公钥认证相似
- 和公钥认证的方法的不同之处：
  - 用客户端主机的私钥（公钥认证使用用户的私钥）进行签名
  - 服务器用客户端主机的公钥（公钥认证使用用户的公钥）验证签名。
- 前提：主机已通过用户身份验证，客户端主机充当用户身份认证代理。
- 使用场合：主机有多个客户

三 基于主机认证方法

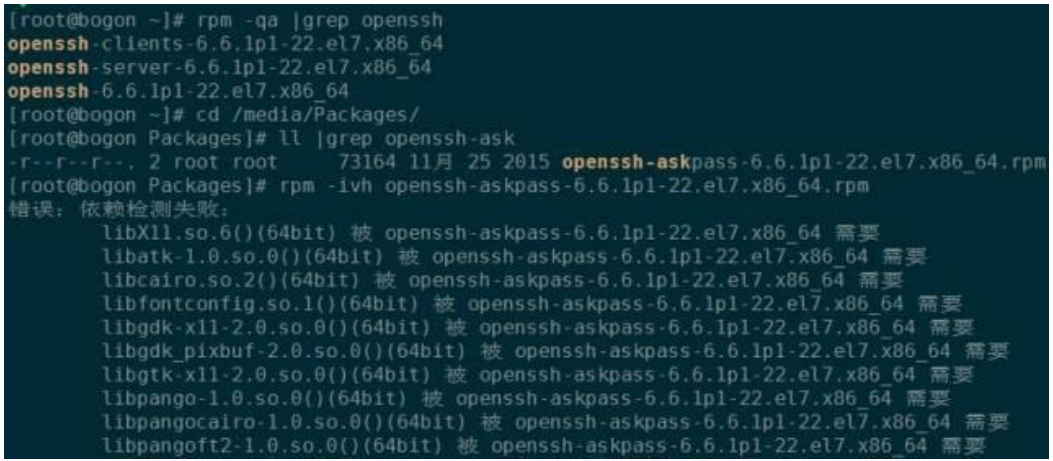
- 客户端的请求消息包含：“认证方法名”字段设置为hostbased、签名算法、证书、用户端主机的FQDN、用户名、签名。
- 服务器根据以下内容验证客户端身份：
  - 证书是否合法
  - 给出的用户是否允许在这台主机登录
  - 签名是否正确
  - 客户端主机的合格域名是否与报文的源地址一致。
- 服务器给出成功或者失败的应答

三 交互式认证方法


客户端
服务器
SSH_MSG_USERAUTH_REQUEST
SSH_MSG_USERAUTH_INFO_REQUEST
SSH_MSG_USERAUTH_INFO_SUCCESS
SSH_MSG_USERAUTH_INFO_FAILURE
接收用户输入的认证信息
SSH_MSG_USERAUTH_INFO_RESPONSE
SSH_MSG_USERAUTH_REQUEST
SSH_MSG_USERAUTH_SUCCESS
SSH_MSG_USERAUTH_FAILURE

SSH

- SSH（Secure Shell)
  - SSH可以将所有的传输数据加密，这样“中间人”这种攻击方式就不可能实现了，而且也可以防止DNS和IP欺骗。
  - SSH是由服务端和客户端的软件组成，服务端是一个守护进程，它在后台运行并响应来自客户端的连接请求。



办公地点：理科大楼B1209

联系方式：17621203829

邮箱：liuhongler@foxmail.com