

这个人在加班

图解SSH原理

- 本文目录
- 1. 初见SSH
 - 2. SSH工作原理
 - 3. SSH实践
 - 4. 总结



1. 初见SSH

SSH是一种协议标准，其目的是实现安全远程登录以及其它安全网络服务。

SSH仅仅是一协议标准，其具体的实现有很多，既有开源实现的OpenSSH，也有商业实现方案。使用范围最广泛的当然是开源实现OpenSSH。

2. SSH工作原理

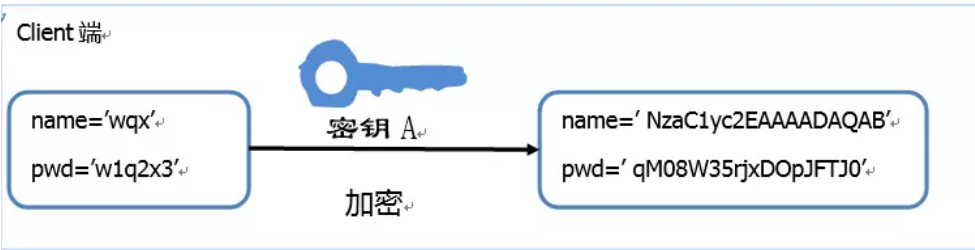
在讨论SSH的原理和使用前，我们需要分析一个问题：为什么需要SSH？

从1.1节SSH的定义中可以看出，SSH和telnet、ftp等协议主要的区别在于安全性。这就引出下一个问题：如何实现数据的安全呢？首先想到的实现方案肯定是对数据进行加密。加密的方式主要有两种：

- 对称加密（也称为秘钥加密）
- 非对称加密（也称公钥加密）

所谓对称加密，指加密解密使用同一套秘钥。如下图所示：

Client:



Server:

公告



访问总量:
AmazingCounters.com

昵称： 这个人在加班
园龄： 1年
粉丝： 3
关注： 4
+加关注

积分与排名

积分 - 32841
排名 - 19835

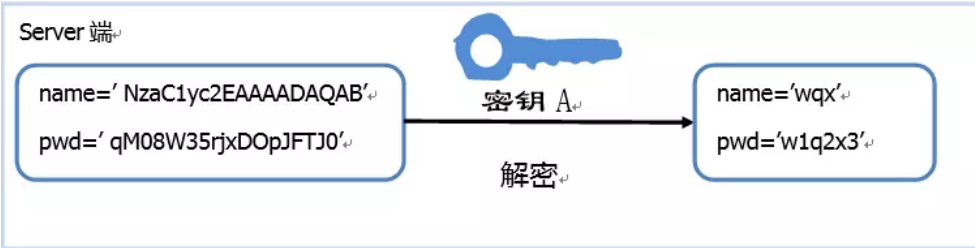
随笔分类 (78)

- ad(2)
- Android(3)
- book
- design pattern(4)
- git(3)
- IDE(6)
- Java(4)
- JS(6)
- Life(1)
- Linux(9)
- mybatis
- MySQL(7)
- network(2)
- Other(1)
- python(1)
- test(2)
- TODO(2)
- tools(3)
- Web(18)
- Windows(4)

随笔档案 (74)

- 2019年8月(2)
- 2019年7月(3)
- 2019年6月(4)
- 2019年5月(1)
- 2019年4月(5)
- 2019年3月(9)
- 2019年2月(3)
- 2019年1月(1)
- 2018年12月(3)
- 2018年11月(6)
- 2018年10月(7)
- 2018年9月(13)
- 2018年8月(17)

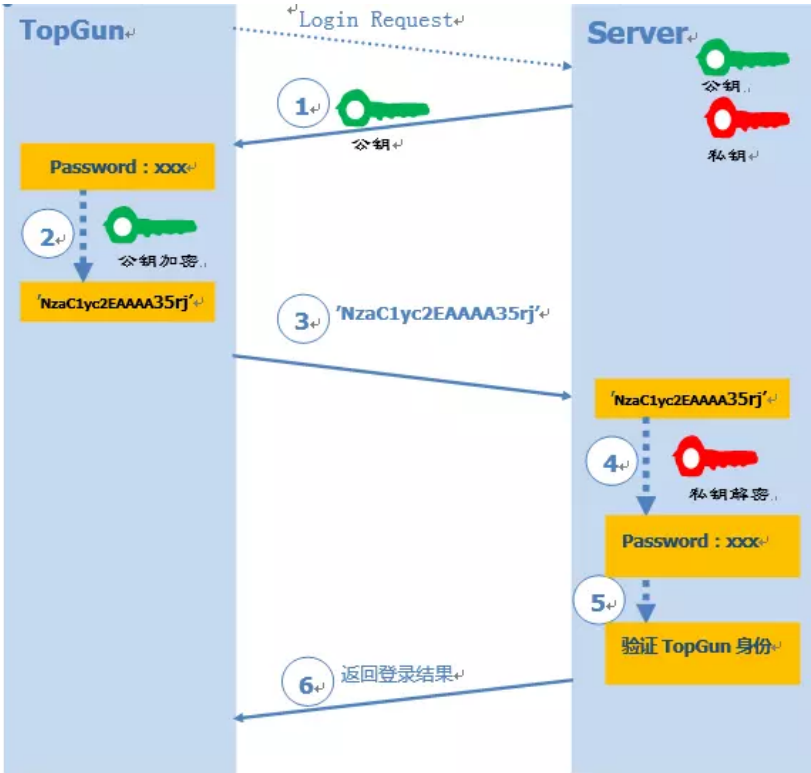
最新评论



对称加密的加密强度高，很难破解。但是在实际应用过程中不得不面临一个棘手的问题：如何安全的保存密钥呢？尤其是考虑到数量庞大的Client端，很难保证密钥不被泄露。一旦一个Client端的密钥被窃据，那么整个系统的安全性也就不复存在。为了解决这个问题，非对称加密应运而生。非对称加密有两个密钥：“公钥”和“私钥”。

两个密钥的特性：公钥加密后的密文，只能通过对应的私钥进行解密。而通过公钥推理出私钥的可能性微乎其微。

下面看下使用非对称加密方案的登录流程：



- 1.远程Server收到Client端用户TopGun的登录请求，Server把自己的公钥发给用户。
- 2.Client使用这个公钥，将密码进行加密。
- 3.Client将加密的密码发送给Server端。
- 4.远程Server用自己的私钥，解密登录密码，然后验证其合法性。
- 5.若验证结果，给Client相应的响应。

私钥是Server端独有，这就保证了Client的登录信息即使在网络传输过程中被窃据，也没有私钥进行解密，保证了数据的安全性，这充分利用了非对称加密的特性。

这样就一定安全了吗？

上述流程会有一个问题：**Client端**如何保证接受到的公钥就是目标**Server端**的？，如果一个攻击者中途拦截Client的登录请求，向其发送自己的公钥，Client端用攻击者的公钥进行数据加密。攻击者接收到加密信息后再用自己的私钥进行解密，不就窃取了Client的登录信息了吗？这就是所谓的中间人攻击

1. Re:Error:(1, 1) java: 非法字符: '\ufeff'
这波操作完美
--陈晨飞抵
2. Re:MSBUILD : error MSB3428: 未能加载 Visual C++ 组件"VCBuild.exe"
问题已经解决，刚开始遇到这个问题，一直以为缺少.NET和VS组件，下载完安装以后还是没解决。后来降低了nodejs的版本之前nodejs10版本，现在降成nodejs8.11.3
--一盏清茶
3. Re:MSBUILD : error MSB3428: 未能加载 Visual C++ 组件"VCBuild.exe"
是什么问题呢？
--一盏清茶
4. Re:Error:(1, 1) java: 非法字符: '\ufeff'
@ 云中欧龙it's nothing...
--这个人在加班
5. Re:Error:(1, 1) java: 非法字符: '\ufeff'
感谢老铁
--云中欧龙

阅读排行榜

1. check the manual that corresponds to your MySQL server version for the right syntax to use near(8731)
2. Error:(1, 1) java: 非法字符: '\ufeff'(7754)
3. maven的pom文件报错: must be "pom" but is "jar"(3107)
4. 图解SSH原理(1951)
5. Please restart this script from an administrative PowerShell(1576)

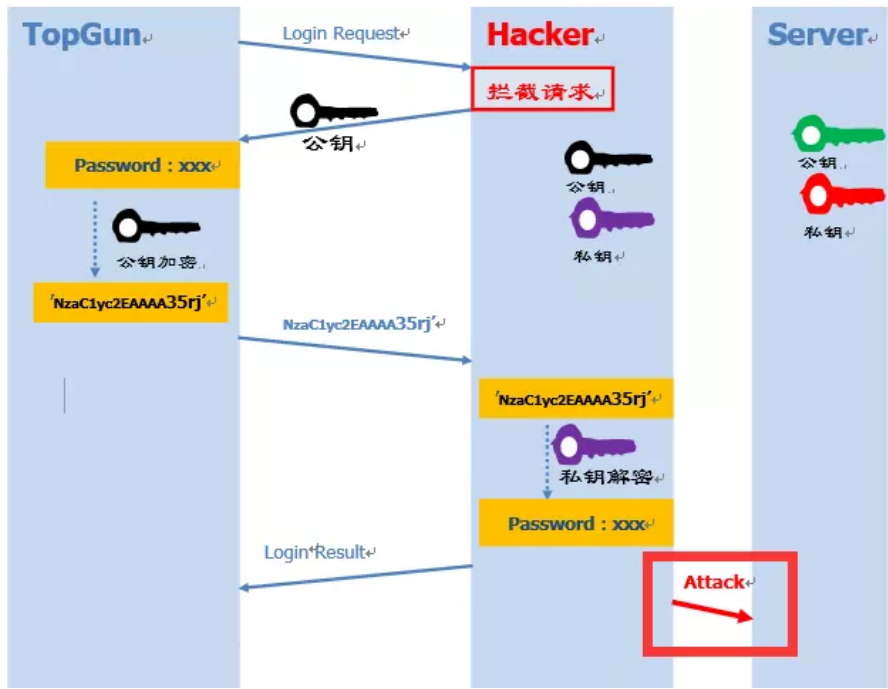
评论排行榜

1. Error:(1, 1) java: 非法字符: '\ufeff'(5)
2. Spring boot读取application.properties中文乱码(2)
3. MSBUILD : error MSB3428: 未能加载 Visual C++ 组件"VCBuild.exe"(2)

推荐排行榜

1. Error:(1, 1) java: 非法字符: '\ufeff'(3)
2. Spring boot读取application.properties中文乱码(2)
3. MSBUILD : error MSB3428: 未能加载 Visual C++ 组件"VCBuild.exe"(2)
4. windows下pwd、ls、tail-f命令使用(1)
5. org.springframework.mail.MailSendException: Failed messages: javax.mail.SendFailedException: Invalid Addresses(1)





SSH中是如何解决这个问题的？

1. 基于口令的认证

从上面的描述可以看出，问题就在于如何对Server的公钥进行认证？在https中可以通过CA来进行公证，可是SSH的publish key和private key都是自己生成的，没法公证。只能通过Client端自己对公钥进行确认。通常在第一次登录的时候，系统会出现下面提示信息：

```
The authenticity of host 'ssh-server.example.com (12.18.429.21)' can't be established.  
RSA key fingerprint is 98:2e:d7:e0:de:9f:ac:67:28:c2:42:2d:37:16:58:4d.  
Are you sure you want to continue connecting (yes/no)?
```

上面的信息说的是：无法确认主机ssh-server.example.com（12.18.429.21）的真实性，不过知道它的公钥指纹，是否继续连接？

之所以用fingerprint代替key，主要是key过于长（RSA算法生成的公钥有1024位），很难直接比较。所以，对公钥进行hash生成一个128位的指纹，这样就方便比较了。

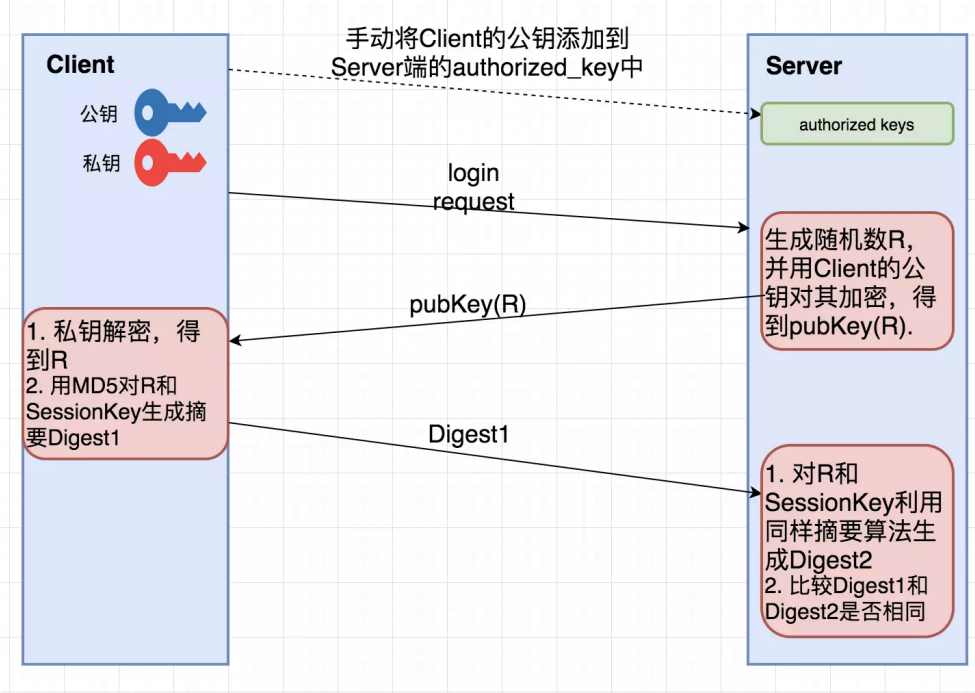
如果输入yes后，会出现下面信息：

```
Warning: Permanently added 'ssh-server.example.com,12.18.429.21' (RSA) to the list of known hosts.  
Password: (enter password)
```

该host已被确认，并被追加到文件known_hosts中，然后就需要输入密码，之后的流程就按照图1-3进行。

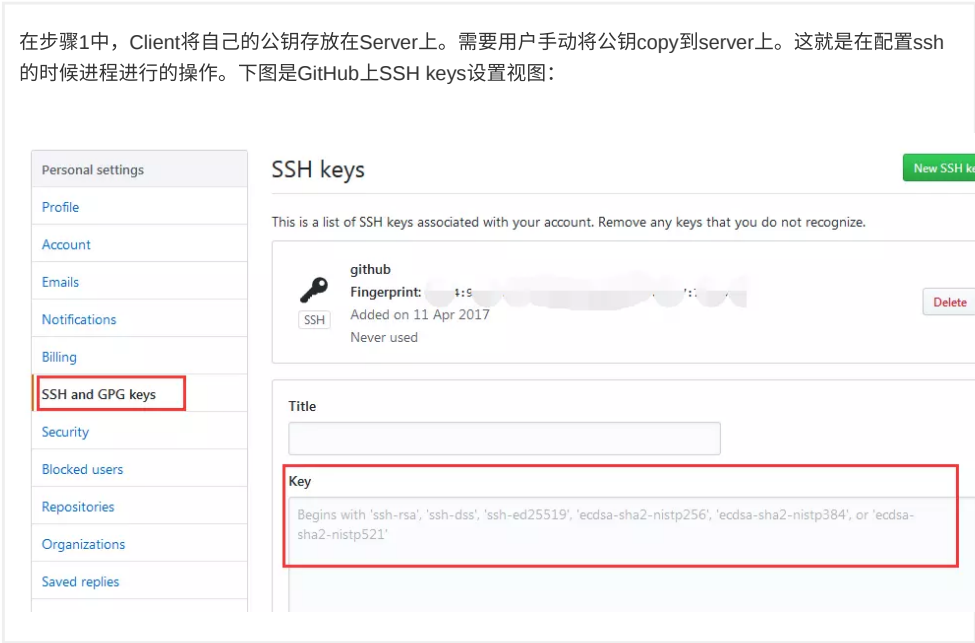
2.基于公钥认证

在上面介绍的登录流程中可以发现，每次登录都需要输入密码，很麻烦。SSH提供了另外一种可以免去输入密码过程的登录方式：公钥登录。流程如下：



- 1.Client将自己的公钥存放在Server上，追加在文件authorized_keys中。
- 2.Server端接收到Client的连接请求后，会在authorized_keys中匹配到Client的公钥pubKey，并生成随机数R，用Client的公钥对该随机数进行加密得到pubKey(R)，然后将加密后信息发送给Client。
- 3.Client端通过私钥进行解密得到随机数R，然后对随机数R和本次会话的SessionKey利用MD5生成摘要Digest1，发送给Server端。
- 4.Server端会也会对R和SessionKey利用同样摘要算法生成Digest2。
- 5.Server端会最后比较Digest1和Digest2是否相同，完成认证过程。

在步骤1中，Client将自己的公钥存放在Server上。需要用户手动将公钥copy到server上。这就是在配置ssh的时候进程进行的操作。下图是GitHub上SSH keys设置视图：



3. SSH实践

生成密钥操作





经过上面的原理分析，下面三行命令的含义应该很容易理解了：

```
$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
$ chmod 0600 ~/.ssh/authorized_keys
```

ssh-keygen是用于生产密钥的工具。

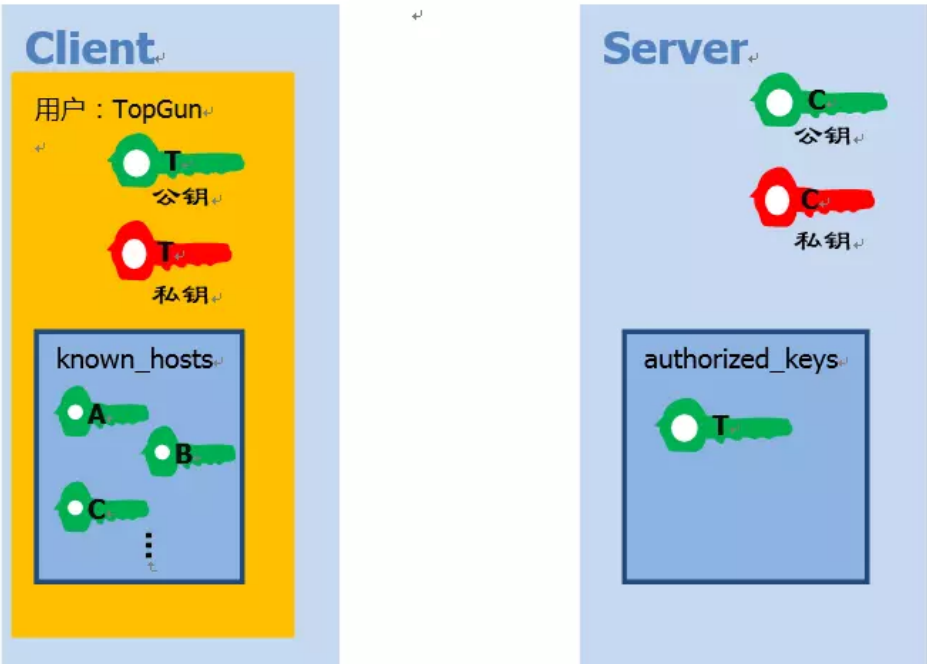
- -t: 指定生成密钥类型 (rsa、dsa、ecdsa等)
- -P: 指定passphrase，用于确保私钥的安全
- -f: 指定存放密钥的文件（公钥文件默认和私钥同目录下，不同的是，存放公钥的文件名需要加上后缀.pub）

首先看下面~/.ssh中的四个文件：

	authorized_keys	2017/4/13 22:11	文件	1 KB
	id_rsa	2017/4/11 9:11	文件	2 KB
	id_rsa.pub	2017/4/11 9:11	Microsoft Publis...	1 KB
	known_hosts	2017/4/16 22:04	文件	1 KB

- 1.id_rsa：保存私钥
- 2.id_rsa.pub：保存公钥
- 3.authorized_keys：保存已授权的客户端公钥
- 4.known_hosts：保存已认证的远程主机ID（关于known_hosts详情，见文末更新内容）

四个角色的关系如下图所示：



需要注意的是：一台主机可能既是Client，也是Server。所以会同时拥有authorized_keys和known_hosts。

登录操作

```
# 以用户名user，登录远程主机host
$ ssh user@host

# 本地用户和远程用户相同，则用户名可省去
$ ssh host

# SSH默认端口22，可以用参数p修改端口
$ ssh -p 2017 user@host
```

4 总结

本文以图文方式对SSH原理进行解析（主要指远程登录，没有涉及端口转发等功能）。同时分析了非对称加密的特性，以及在实际过程中如何对加密操作进行改进。

1. known_hosts中存储的内容是什么？

known_hosts中存储是已认证的远程主机host key，每个SSH Server都有一个secret, unique ID, called a host key。

2. host key何时加入known_hosts的？

当我们第一次通过SSH登录远程主机的时候，Client端会有如下提示：

```
Host key not found from the list of known hosts.
Are you sure you want to continue connecting (yes/no)?
```

此时，如果我们选择yes，那么该host key就会被加入到Client的known_hosts中，格式如下：

```
# domain name+encryption algorithm+host key
example.hostname.com ssh-rsa AAAAB4NzaC1yc2EAAAABIwAAAQEA。。。
```

3. 为什么需要known_hosts？

最后探讨下为什么需要known_hosts，这个文件主要是通过Client和Server的双向认证，从而避免中间人（man-in-the-middle attack）攻击，每次Client向Server发起连接的时候，不仅仅Server要验证Client的合法性，Client同样也需要验证Server的身份，SSH client就是通过known_hosts中的host key来验证Server的身份的。

这中方案足够安全吗？当然不，比如第一次连接一个未知Server的时候，known_hosts还没有该Server的host key，这不也可能遭到中间人攻击吗？这可能只是安全性和可操作性之间的折中吧。

转自：<https://www.jianshu.com/p/33461b619d53>

作者：[DiffX](#) —— 这个人在加班

出处：<http://www.cnblogs.com/diffx/>

本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接，否则保留追究法律责任的权利。

分类： Web

标签： 加密

好文要顶

关注我

收藏该文

[这个人在加班](#)
关注 - 4
粉丝 - 3
[+加关注](#)

[这个人在加班](#)

10

« 上一篇：[监听Google Player下载并获取包名等信息](#)

» 下一篇：[设计模式：生产者消费者模式](#)

posted @ 2018-08-29 12:35 这个人在加班 阅读(1951) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) 网站首页。

【推荐】超50万C++/C#源码: 大型实时仿真组态图形源码

【推荐】零基础轻松玩转云上产品，获赠礼加返百元大礼

【推荐】华为IoT平台开发者套餐9.9元起，购买即送免费课程

- 相关博文：
- [ssh原理图解](#)
 - [SSH登录过程](#)
 - [SSH学习笔记](#)
 - [linux上ssh免密登录原理及实现](#)
 - [SSH原理和使用](#)



最新 IT 新闻:

- 科学家发现镍氧化物超导体
 - 月亮盈亏关联情绪波动
 - 我国正在制定AVS3编码标准 视频容量可压缩千倍
 - 发热的衣服材料还能防止蚊子叮咬？石墨烯材料在防蚊上有保护作用
 - NASA在火星车装了直升机，将在火星上飞行
- » 更多新闻...

