马哥教育
最专业的Linux培训机构

安全和加密

本章内容

- 安全机制
- 对称加密
- 不对称加密
- 散列算法
- PKI和CA
- openssl
- 证书管理
- gpg
- ssh服务
- dropbear
- aide
- sudo

加密需要

- 不加密流量的易受攻击性
  - 密码/数据嗅探
  - 数据操作
  - 验证操作
  - 相当于邮寄明信片
- 不安全的传统协议
  - telnet、FTP、POP3等等；不安全密码
  - http、smtp、NFS等等;不安全信息
  - Ldap、NIS、rsh等等；不安全验证

安全机制

- NIST(美国国家标准与技术研究院)定义的安全属性：
  - 保密性：
    - 数据保密性
    - 隐私性
  - 完整性：不可篡改
    - 数据完整性
    - 系统完整性
  - 可用性
- 安全攻击： STRIDE

Spoofing（假冒）、Tampering（篡改）、Repudiation（否认）、Information Disclosure（信息泄漏）、Denial of Service（拒绝服务）和 Elevation of Privilege（提升权限）

安全

- 安全机制：

  加密、数字签名、访问控制、数据完整性、认证交换、流量填充、路由控制、公证

- 安全服务：

  认证

  访问控制

  数据保密性

    连接保密性

    无连接保密性

    选择域保密性

    流量保密性

  数据完整性

  不可否认性

设计基本原则

- 使用成熟的安全系统
- 以小人之心度输入数据
- 外部系统是不安全的
- 最小授权
- 减少外部接口
- 缺省使用安全模式
- 安全不是似是而非
- 从STRIDE思考
- 在入口处检查
- 从管理上保护好你的系统

安全算法

- 常用安全技术
  - 认证
  - 授权
  - 安全通信
  - 审计
- 密码算法和协议：
  - 对称加密
  - 公钥加密
  - 单向加密
  - 认证协议
- Linux系统：OpenSSL，gpg(pgp协议的实现)

对称加密算法

- 对称加密：加密和解密使用同一个密钥
  - DES：Data Encryption Standard，56bits
  - 3DES：
  - AES：Advanced (128，192，256bits)
  - Blowfish，Twofish
  - IDEA，RC6，CAST5
- 特性：
  - 1、加密、解密使用同一个密钥，效率高
  - 2、将原始数据分割成固定大小的块，逐个进行加密
- 缺陷：
  - 1、密钥过多
  - 2、密钥分发
  - 3、数据来源无法确认

非对称加密算法

- 公钥加密：密钥是成对出现
  - 公钥：公开给所有人；public key
  - 私钥：自己留存，必须保证其私密性；secret key
- 特点：用公钥加密数据，只能使用与之配对的私钥解密；反之亦然
- 功能：
  - 数字签名：主要在于让接收方确认发送方身份
  - 对称密钥交换：发送方用对方的公钥加密一个对称密钥后发送给对方
  - 数据加密：适合加密较小数据
- 缺点：密钥长，加密解密效率低下
- 算法：

  RSA（加密，数字签名）,DSA（数字签名）,ELGamal

非对称加密

- 基于一对公钥/密钥对
  - 用密钥对中的一个加密，另一个解密
- 实现加密：
  - 接收者

    生成公钥/密钥对：P和S

    公开公钥P，保密密钥S
  - 发送者

    使用接收者的公钥来加密消息M

    将P(M)发送给接收者
  - 接收者

    使用密钥S来解密：M=S(P(M))

非对称加密

- 实现数字签名：
  - 发送者
    - 生成公钥/密钥对：P和S
    - 公开公钥P，保密密钥S
    - 使用密钥S来加密消息M
    - 发送给接收者S(M)
  - 接收者
    - 使用发送者的公钥来解密M=P(S(M))
- 结合签名和加密
- 分离签名

单向散列

- 将任意数据缩小成固定大小的“指纹”
  - 任意长度输入
  - 固定长度输出
  - 若修改数据，指纹也会改变（“不会产生冲突”）
  - 无法从指纹中重新生成数据（“单向”）
- 功能：数据完整性
- 常见算式

  md5：128bits、sha1：160bits、sha224
  sha256、sha384、sha512
- 常用工具
  - md5sum | sha1sum [--check] file
  - openssl、gpg
  - rpm -V

密钥交换

- 密钥交换：IKE（ Internet Key Exchange ）

  公钥加密：

  DH (Deffie-Hellman)：

- DH：

  1、A：a,p 协商生成公开的整数a，大素数p

  B：a,p

  2、A:生成隐私数据 :x (x<p)，计算得出 a^x%p，发送给B

  B:生成隐私数据 :y,计算得出 a^y%p，发送给A

  3、A:计算得出 （a^y%p）^x = a^xy%p，生成为密钥

  B:计算得出 （a^x%p）^y = a^xy%p，生成为密钥

CA和证书

- PKI: Public Key Infrastructure
  - 签证机构：CA（Certificate Authority）
  - 注册机构：RA
  - 证书吊销列表：CRL
  - 证书存取库：
- X.509：定义了证书的结构以及认证协议标准
  - 版本号
  - 序列号
  - 签名算法
  - 颁发者
  - 有效期限
  - 主体名称
  - 主体公钥
  - CRL分发点
  - 扩展信息
  - 发行者签名

证书获取

- 证书类型：
  - 证书授权机构的证书
  - 服务器
  - 用户证书
- 获取证书两种方法：
  - 使用证书授权机构
    - 生成签名请求（csr）
    - 将csr发送给CA
    - 从CA处接收签名
  - 自签名的证书
    - 自己签发自己的公钥

安全协议

- SSL: Secure Socket Layer

  TLS: Transport Layer Security

  1995：SSL 2.0 Netscape

  1996：SSL 3.0

  1999：TLS 1.0

  2006：TLS 1.1 RFC（ Request For Comments ） 4346

  2008：TLS 1.2 当前使用

  2015：TLS 1.3

  功能：机密性，认证，完整性，重放保护

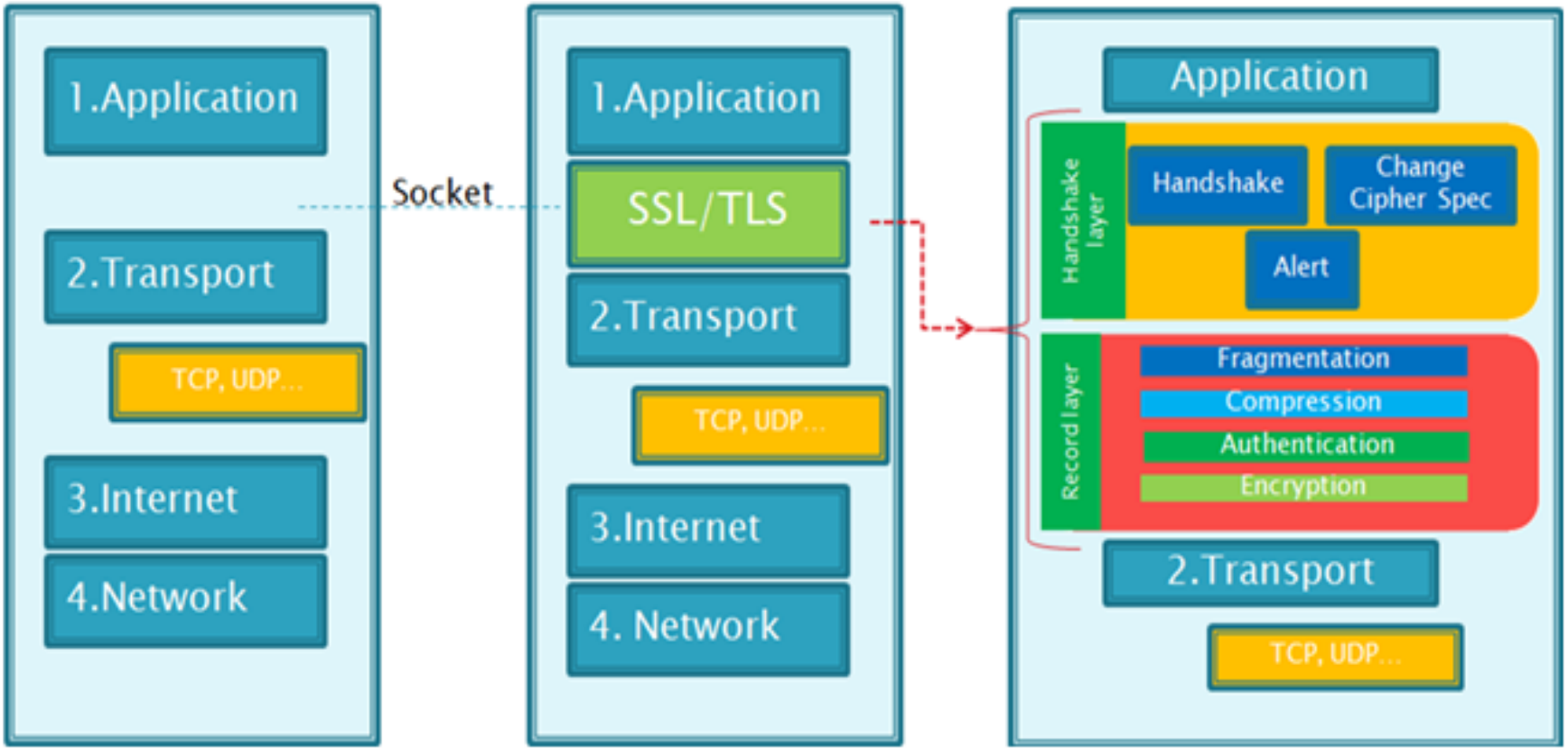
- 两阶段协议，分为握手阶段和应用阶段

  握手阶段(协商阶段):客户端和服务器端认证对方身份（依赖于PKI体系，利用数字证书进行身份认证），并协商通信中使用的安全参数、密码套件以及主密钥。后续通信使用的所有密钥都是通过MasterSecret生成。

  应用阶段:在握手阶段完成后进入，在应用阶段通信双方使用握手阶段协商好的密钥进行安全通信。

TCP/IP Model

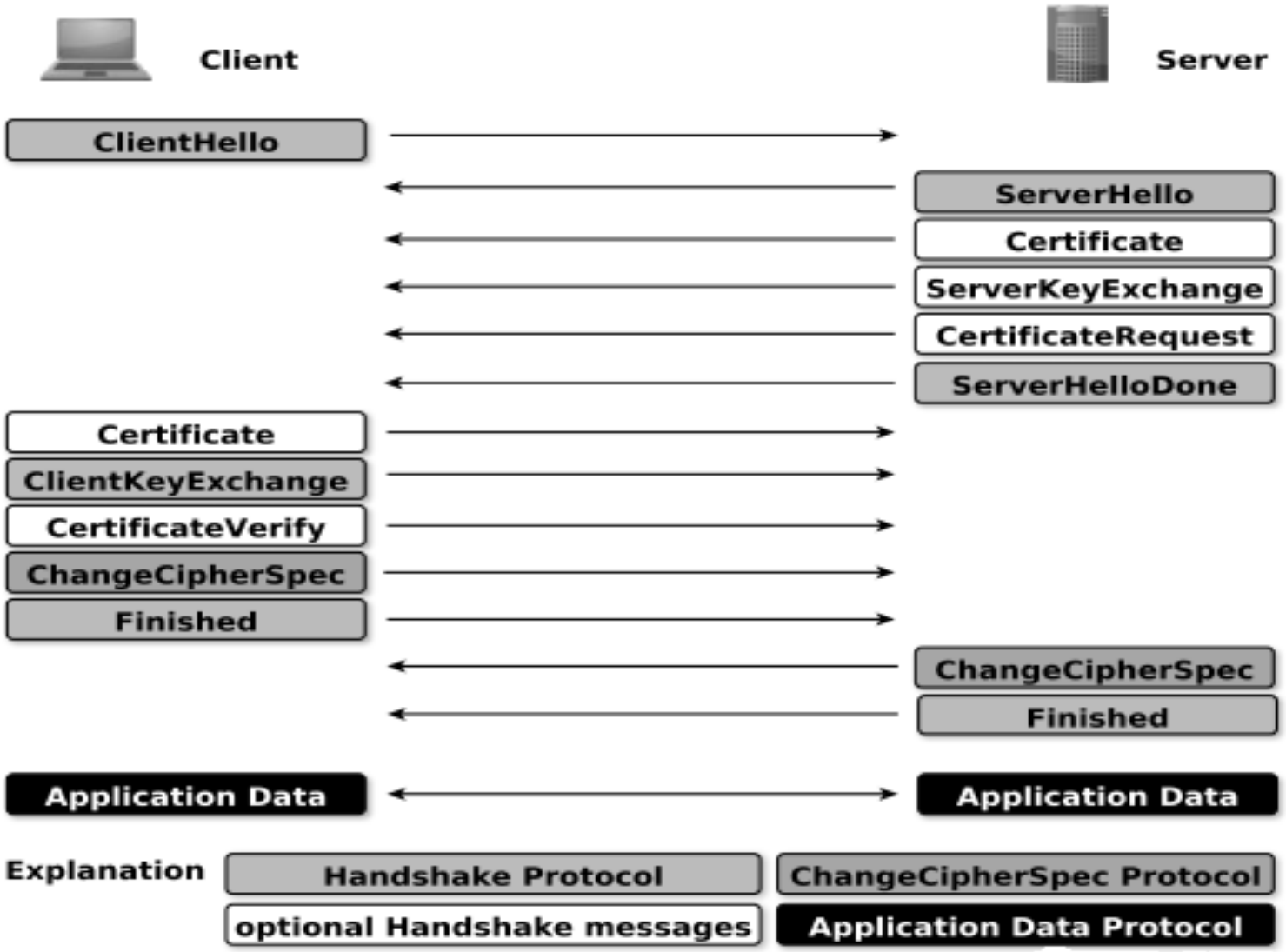
SSL/TLS Protocol



SSL/TLS

- Handshake协议：包括协商安全参数和密码套件、服务器身份认证（客户端身份认证可选）、密钥交换
- ChangeCipherSpec 协议：一条消息表明握手协议已经完成
- Alert 协议：对握手协议中一些异常的错误提醒，分为fatal和warning两个级别，fatal类型错误会直接中断SSL链接，而warning级别的错误SSL链接仍可继续，只是会给出错误警告
- Record 协议：包括对消息的分段、压缩、消息认证和完整性保护、加密等
- HTTPS 协议：就是“HTTP 协议”和“SSL/TLS 协议”的组合。HTTP over SSL”或“HTTP over TLS”，对http协议的文本数据进行加密处理后，成为二进制形式传输

SSL/TLS


Client
Server
ClientHello
ServerHello
Certificate
ServerKeyExchange
CertificateRequest
ServerHelloDone
Certificate
ClientKeyExchange
CertificateVerify
ChangeCipherSpec
Finished
ChangeCipherSpec
Finished
Application Data
Application Data
Explanation
Handshake Protocol
ChangeCipherSpec Protocol
optional Handshake messages
Application Data Protocol

- OpenSSL：开源项目

  三个组件：

  openssl：多用途的命令行工具

  libcrypto：加密算法库

  libssl：加密模块应用库，实现了ssl及tls

- openssl命令：

  两种运行模式：交互模式和批处理模式

  openssl version：程序版本号

  标准命令、消息摘要命令、加密命令

  标准命令：

  enc, ca, req, ...

openssl命令

- 对称加密：

  工具：openssl enc, gpg

  算法：3des, aes, blowfish, twofish

- enc命令：man enc

  加密：

  ```
  openssl enc -e -des3 -a -salt -in testfile
          -out testfile.cipher
  ```

  解密：

  ```
  openssl enc -d -des3 -a -salt -in testfile.cipher
          -out testfile
  ```

  ```
  openssl ?
  ```

openssl命令

- 单向加密：

  工具：md5sum，sha1sum，sha224sum,sha256sum…

  openssl dgst

- dgst命令：man dgst

  openssl dgst -md5 [-hex默认] /PATH/SOMEFILE

  openssl dgst -md5 testfile

  md5sum /PATH/TO/SOMEFILE

- MAC：Message Authentication Code，单向加密的一种延伸应用，用于实现网络通信中保证所传输数据的完整性机制

  CBC-MAC

  HMAC：使用md5或sha1算法

openssl命令

- 生成用户密码：

  passwd命令:man sslpasswd

  openssl passwd -1 -salt SALT(最多8位)

  openssl passwd -1 -salt centos

- 生成随机数：man sslrand

  openssl rand -base64|-hex NUM

  NUM：表示字节数；-hex时，每个字符4位，出现的字符数为NUM*2

openssl命令

- 公钥加密：
  - 算法：RSA，ELGamal
  - 工具：gpg，openssl rsautl（man rsautl）
- 数字签名：
  - 算法：RSA，DSA，ELGamal
- 密钥交换：
  - 算法：dh
  - DSA：Digital Signature Algorithm
  - DSS：Digital Signature Standard
  - RSA：

openssl命令

- 生成密钥对儿：man genrsa
- 生成私钥：

openssl genrsa -out /PATH/TO/PRIVATEKEY.FILE NUM_BITS
(umask 077; openssl genrsa -out key.pri -des 2048)

- 从私钥中提取出公钥：

openssl rsa -in PRIVATEKEYFILE -pubout -out PUBLICKEYFILE

- 随机数生成器：伪随机数字

键盘和鼠标

块设备中断

/dev/random：仅从熵池返回随机数；随机数用尽，阻塞

/dev/urandom：从熵池返回随机数；随机数用尽，会利用软件生成伪随机数,非阻塞

- PKI：Public Key Infrastructure
  - CA
  - RA
  - CRL
  - 证书存取库
- 建立私有CA：
  - OpenCA
  - openssl
- 证书申请及签署步骤：
  - 1、生成申请请求
  - 2、RA核验
  - 3、CA签署
  - 4、获取证书

创建CA和申请证书

- 创建私有CA：

  openssl的配置文件：/etc/pki/tls/openssl.cnf

- (1) 创建所需要的文件

  touch /etc/pki/CA/index.txt

  echo 01 > /etc/pki/CA/serial

- (2) CA自签证书

  生成私钥

  cd /etc/pki/CA/

  (umask 066; openssl genrsa -out /etc/pki/CA/private/cakey.pem 2048)

创建CA和申请证书

- 生成自签名证书

```
openssl req -new -x509 -key
/etc/pki/CA/private/cakey.pem -days 7300
    -out /etc/pki/CA/cacert.pem
```

-new：生成新证书签署请求

-x509：专用于CA生成自签证书

-key：生成请求时用到的私钥文件

-days n：证书的有效期限

-out /PATH/TO/SOMECERTFILE：证书的保存路径

创建CA和申请证书

- (3) 颁发证书
- (a) 在需要使用证书的主机生成证书请求；

  给web服务器生成私钥

  (umask 066; openssl genrsa -out /etc/httpd/ssl/httpd.key 2048)

  生成证书申请文件

  openssl req -new -key /etc/httpd/ssl/httpd.key -days 365 -out /etc/httpd/ssl/httpd.csr
- (b) 将证书请求文件传输给CA
- (c) CA签署证书，并将证书颁发给请求者；

  openssl ca -in /tmp/httpd.csr -out /etc/pki/CA/certs/httpd.crt -days 365

  注意：默认国家，省 ，公司名称必须和CA一致

创建CA和证书管理

- (d) 查看证书中的信息：

  openssl x509 -in /PATH/FROM/CERT_FILE -noout -text|subject|serial|dates

- (4) 吊销证书
- (a) 在客户端获取要吊销的证书的serial

  openssl x509 -in /PATH/FROM/CERT_FILE -noout -serial -subject

- (b) 在CA上，根据客户提交的serial与subject信息，对比检验是否与index.txt文件中的信息一致

  吊销证书：

  openssl ca -revoke /etc/pki/CA/newcerts/SERIAL.pem

创建CA和证书管理

- (c) 生成吊销证书的编号(第一次吊销一个证书时才需要执行)

  echo 01 > /etc/pki/CA/crlnumber

- (d) 更新证书吊销列表

  openssl ca -gencrl -out /etc/pki/CA/crl/ca.crl

  查看crl文件：

  openssl crl -in /etc/pki/CA/crl/ca.crl

  -noout -text

应用程序：RPM

- 文件完整性的两种实施方式
- 被安装的文件
  - MD5单向散列
  - rpm --verify package_name (or -V)
- 发行的软件包文件
  - GPG公钥签名
  - rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat*
  - rpm --checksig pakage_file_name (or -K)

使用gpg实现对称加密

- 对称加密file文件

```
gpg -c file
ls file.gpg
```

- 在另一台主机上解密file

```
gpg -o file -d file.gpg
```

使用gpg工具实现公钥加密

- 在hostB主机上用公钥加密，在hostA主机上解密
- 在hostA主机上生成公钥/私钥对

```
gpg --gen-key
```

- 在hostA主机上查看公钥

```
gpg --list-keys
```

- 在hostA主机上导出公钥到wang.pubkey

```
gpg -a  --export -o wang.pubkey
```

- 从hostA主机上复制公钥文件到需加密的B主机上

```
scp wang.pubkey  hostB:
```

使用gpg工具实现公钥加密

- 在需加密数据的hostB主机上生成公钥/私钥对

```
gpg --list-keys
gpg --gen-key
```

- 在hostB主机上导入公钥

```
gpg --import wang.pubkey
gpg --list-keys
```

- 用从hostA主机导入的公钥，加密hostB主机的文件file,生成file.gpg

```
gpg -e -r wangxiaochun file
file file.gpg
```

使用gpg工具实现公钥加密

- 复制加密文件到hostA主机

```
scp fstab.gpg hostA:
```

- 在hostA主机解密文件

```
gpg -d file.gpg
gpg -o file  -d file.gpg
```

- 删除公钥和私钥

```
gpg --delete-keys  wangxiaochun
gpg --delete-secret-keys  wangxiaochun
```

SSH

- ssh：secure shell，protocol，22/tcp，安全的远程登录
- OpenSSH：ssh协议的开源实现
- dropbear：另一个开源实现
- SSH协议版本
  - v1：基于CRC-32做MAC，不安全；man-in-middle
  - v2：双方主机协议选择安全的MAC方式
  - 基于DH算法做密钥交换，基于RSA或DSA实现身份认证
- 两种方式的用户登录认证：
  - 基于password
  - 基于key

- OpenSSH:
  - C/S
    - C: ssh, scp, sftp
    - Windows客户端:
    - xshell, putty, securecrt, sshsecureshellclient
  - S: sshd

ssh客户端

- 客户端组件：
- ssh，配置文件：/etc/ssh/ssh_config
  Host PATTERN
    StrictHostKeyChecking no 首次登录不显示检查提示
- 格式：ssh [user@]host [COMMAND]
  ssh [-l user] host [COMMAND]
    -p port：远程服务器监听的端口
    -b:指定连接的源IP
    -v:调试模式
    -C：压缩方式
    -X：支持x11转发
    -Y：支持信任x11转发
      ForwardX11Trusted yes
    -t： 强制伪tty分配
      ssh -t remoteserver1 ssh remoteserver2

ssh客户端

- 允许实现对远程系统经验证地加密安全访问
- 当用户远程连接ssh服务器时，会复制ssh服务器/etc/ssh/ssh_host*key.pub（centos7.0默认是ssh_host_ecdsa_key.pub）文件中的公钥到客户机的~./ssh/know_hosts中。下次连接时，会比较两处是否有不同。

基于key认证

- 基于密钥的认证：
- (1) 在客户端生成密钥对

  ssh-keygen -t rsa [-P ''] [-f "/root/.ssh/id_rsa"]

  #ssh-keygen -t rsa -P '' -f "/root/.ssh/id_rsa"
- (2) 把公钥文件传输至远程服务器对应用户的家目录

  ssh-copy-id [-i [identity_file]] [user@]host
- (3) 测试
- (4) 在SecureCRT，Xshell或实现基于key验证

在SecureCRT工具—>创建公钥—>生成Identity.pub文件

转化为openssh兼容格式（适合SecureCRT，Xshell不需要转化格式），并复制到需登录主机上相应文件authorized_keys中，注意权限必须为600，在需登录的ssh主机上执行：

ssh-keygen -i -f Identity.pub >> .ssh/authorized_keys

基于key认证

- (5)重设私钥口令：#ssh-keygen -p
- (6)验证代理（authentication agent）保密解密后的密钥
  - 这样口令就只需要输入一次
  - 在GNOME中，代理被自动提供
  - 否则运行ssh-agent bash
- (7)钥匙通过命令添加给代理

  ssh-add

scp命令

- scp命令：
- scp [options] SRC... DEST/
- 两种方式：

  scp [options] [user@]host:/sourcefile /destpath

  scp [options] /sourcefile [user@]host:/destpath
- 常用选项：

  -C：压缩数据流

  -r：递归复制

  -p：保持原文件的属性信息

  -q：静默模式

  -P PORT：指明remote host的监听的端口

rsync命令

- 基于ssh和rsh服务实现高效率的远程系统之间复制文件
- 使用安全的shell连接做为传输方式
  - rsync -av /etc server1:/tmp 复制目录和目录下文件
  - rsync -av /etc/ server1:/tmp 只复制目录下文件
- 比scp更快，只复制不同的文件
- 选项：

-n 模拟复制过程
-v 显示详细过程
-r 递归复制目录树
-p 保留权限
-t 保留时间戳
-g 保留组信息
-o 保留所有者信息
-l 把符号链接文件做为符号文件进行复制（默认）
-L 将软链接文件指向文件复制
-a 存档模式，相当于 -rlptgoD，但不保留ACL（-A）和SELinux属性（-X）

sftp命令

- 交互式文件传输工具
- 用法能和传统的ftp工具相似
- 利用ssh服务实现安全的文件上传和下载
- 使用ls cd mkdir rmdir pwd get put等指令，可用？获取帮助信息。

```
sftp [user@]host
sftp> help
```

SSH端口转发

❖ 什么是SSH端口转发？

SSH 会自动加密和解密所有 SSH 客户端与服务端之间的网络数据。但是，SSH 还能够将其他 TCP 端口的网络数据通过 SSH 链接来转发，并且自动提供了相应的加密及解密服务。这一过程也被叫做“隧道”（tunneling），这是因为 SSH 为其他 TCP 链接提供了一个安全的通道来进行传输而得名。例如，Telnet，SMTP，LDAP 这些 TCP 应用均能够从中得益，避免了用户名，密码以及隐私信息的明文传输。而与此同时，如果工作环境中的防火墙限制了一些网络端口的使用，但是允许 SSH 的连接，也能够通过将 TCP 端口转发来使用 SSH 进行通讯

❖ SSH 端口转发能够提供两大功能：

- 加密 SSH Client 端至 SSH Server 端之间的通讯数据
- 突破防火墙的限制完成一些之前无法建立的 TCP 连接。

SSH端口转发

- 本地转发：

  -L localport:host:hostport sshserver

  ssh -L 9527:telnetsrv:23 -N sshsrv

  telnet 127.0.0.1 9527

  当访问本机的9527的端口时，被加密后转发到sshsrv的ssh服务，再解密被转发到telnetsrv:23

data ← → localhost:9527 ← → localhost:XXXXX ← → sshsrv:22 ← → sshsrv:YYYYY ← → telnetsrv:23

- 选项：

  -f 后台启用

  -N 不开远程shell

  -g 启用网关功能

SSH端口转发

- 远程转发：

  -R sshserverport:host:hostport sshserver

  ssh -R 9527:telnetsrv:23 -N sshsrv

  让sshsrv侦听9527端口的访问，如有访问，就加密后通过ssh服务转发请求到本机ssh客户端，再由本机解密后转发到telnetsrv:23

Data ← → sshsrv:9527 ← → sshsrv:22 ← → localhost:XXXXX ← → localhost:YYYYY ← → telnetsrv:23

SSH端口转发

- 动态端口转发：
- 当用firefox访问internet时，本机的1080端口做为代理服务器，firefox的访问请求被转发到sshserver上，由sshserver替之访问internet

  在本机firefox设置代理socket proxy:127.0.0.1:1080

  #ssh -D 1080 root@sshserver

X 协议转发

- 所有图形化应用程序都是X客户程序
  - 能够通过tcp/ip连接远程X服务器
  - 数据没有加密机，但是它通过ssh连接隧道安全进行
- ssh -X user@remotehost gedit

  remotehost主机上的gedit工具，将会显示在本机的X服务器上

  传输的数据将通过ssh连接加密

ssh服务器

- 服务器端：
- sshd，配置文件：/etc/ssh/sshd_config
- 常用参数：

  Port

  ListenAddress ip

  PermitRootLogin yes

  ClientAliveInterval 0

  UseDNS yes

  限制可登录用户的办法：

  AllowUsers user1 user2 user3

  DenyUsers

  AllowGroups

  DenyGroups

ssh服务的最佳实践

- 1、不要使用默认端口
- 2、禁止使用protocol version 1
- 3、限制可登录用户
- 4、设定空闲会话超时时长
- 5、利用防火墙设置ssh访问策略
- 6、仅监听特定的IP地址
- 7、基于口令认证时，使用强密码策略

```
tr -dc A-Za-z0-9_ < /dev/urandom | head -c 30 | xargs
```

- 8、使用基于密钥的认证
- 9、禁止使用空密码
- 10、禁止root用户直接登录
- 11、限制ssh的访问频度和并发在线数
- 12、做好日志，经常分析

编译安装dropbear示例

- ssh协议的另一个实现：dropbear
- 安装准备：
- 1、安装开发包组：
- 2、ftp://172.16.0.1/pub/Sources/sources/dropbear
- /dropbear-2013.58.tar.bz2
- 安装：
- 3、tar xf dropbear-2013.58.tar.bz2,
- 4、less INSTALL
- 5、./configure
- 6、make PROGRAMS="dropbear dbclient dropbearkey dropbearconvert scp"
- 7、make PROGRAMS="dropbear dbclient dropbearkey dropbearconvert scp" install

❖ 启动ssh服务：

- 8、ls /usr/local/sbin/ /usr/local/bin/
- 9、/usr/local/sbin/dropbear -h
- 10、mkdir /etc/dropbear
- 11、dropbearkey -t rsa -f /etc/dropbear/dropbear_rsa_host_key -s 2048
- 12、dropbearkey -t dss -f /etc/dropbear/dropbear_dsa_host_key
- 13、dropbear -p :2222 -F -E #前台运行
  dropbear -p :2222 #后台运行

❖ 客户端访问：

- 14、ssh -p 2222 root@127.0.0.1
- 15、dbclient -p 2222 root@127.0.0.1

AIDE

- 当一个入侵者进入了你的系统并且种植了木马，通常会想办法来隐蔽这个木马（除了木马自身的一些隐蔽特性外，他会尽量给你检查系统的过程设置障碍），通常入侵者会修改一些文件，比如管理员通常用ps -aux来查看系统进程，那么入侵者很可能用自己经过修改的ps程序来替换掉你系统上的ps程序，以使用ps命令查不到正在运行的木马程序。如果入侵者发现管理员正在运行crontab作业，也有可能替换掉crontab程序等等。所以由此可以看出对于系统文件或是关键文件的检查是很必要的。目前就系统完整性检查的工具用的比较多的有两款：Tripwire和AIDE，前者是一款商业软件，后者是一款免费的但功能也很强大的工具。

AIDE

- AIDE(Adevanced Intrusion Detection Environment)
- 高级入侵检测环境)是一个入侵检测工具，主要用途是检查文件的完整性，审计计算机上的那些文件被更改过了。
- AIDE能够构造一个指定文件的数据库，它使用aide.conf作为其配置文件。AIDE数据库能够保存文件的各种属性，包括：权限(permission)、索引节点序号(inode number)、所属用户(user)、所属用户组(group)、文件大小、最后修改时间(mtime)、创建时间(ctime)、最后访问时间(atime)、增加的大小以及连接数。AIDE还能够使用下列算法：sha1、md5、rmd160、tiger，以密文形式建立每个文件的校验码或散列号.
- 这个数据库不应该保存那些经常变动的文件信息，例如：日志文件、邮件、/proc文件系统、用户起始目录以及临时目录.

AIDE

- 安装

  yum install aide

- 修改配置文件

  vim /etc/aide.conf (指定对哪些文件进行检测)

  /test/chameleon R
  /bin/ps R+a
  /usr/bin/crontab R+a
  /etc PERMS
  !/etc/mtab # “!”表示忽略这个文件的检查

  R=p+i+n+u+g+s+m+c+md5 权限+索引节点+链接数+用户+组+大小+最后一次修改时间+创建时间+md5校验值

  NORMAL = R+rmd60+sha256

AIDE

- 初始化默认的AIDE的库：

  /usr/local/bin/aide --init

- 生成检查数据库（建议初始数据库存放到安全的地方）

  cd /var/lib/aide

  mv aide.db.new.gz aide.db.gz

- 检测：

  /usr/local/bin/aide --check

- 更新数据库

  aide --update

更改身份

- su 切换身份:su -l username -c 'command'
- sudo 命令
  - 1．sudo能够授权指定用户在指定主机上运行某些命令。如果未授权用户尝试使用 sudo，会提示联系管理员
  - 2．sudo可以提供日志，记录每个用户使用sudo操作
  - 3．sudo为系统管理员提供配置文件，允许系统管理员集中地管理用户的使用权限和使用的主机
  - 4.sudo使用时间戳文件来完成类似“检票”的系统，默认存活期为5分钟的“入场券”
  - 5、通过visudo命令编辑配置文件，具有语法检查功能

- 配置文件：/etc/sudoers，/etc/sudoers.d/
- 时间戳文件：/var/db/sudo
- 日志文件：/var/log/secure
- 配置文件支持使用通配符glob：

  ？:任意单一字符

  * ： 匹配任意长度字符

  [wxc]:匹配其中一个字符

  [!wxc]:除了这三个字符的其它字符

  \x ： 转义

  [[alpha]] :字母 示例： /bin/ls [[alpha]]*

- 配置文件规则有两类；

  1、别名定义:不是必须的

  2、授权规则:必须的

- 授权规则格式：

  用户 登入主机=(代表用户) 命令

- 示例：

  root ALL=(ALL) ALL

- 格式说明：

  user：运行命令者的身份

  host：通过哪些主机

  (runas)：以哪个用户的身份

  command：运行哪些命令

别名

- Users和runas:
  - username
  - #uid
  - %group_name
  - %#gid
  - user_alias|runas_alias
- host:
  - ip或hostname
  - network(/netmask)
  - host_alias
- command:
  - command name
  - directory
  - sudoedit
  - Cmnd_Alias

sudo别名和示例

- 别名有四种类型：User_Alias，Runas_Alias，Host_Alias，Cmnd_Alias
- 别名格式：[A-Z]([A-Z][0-9]_)*
- 别名定义：

Alias_Type NAME1 = item1, item2, item3 : NAME2 = item4, item5

- 示例1：

```
Student    ALL=(ALL)  ALL
 %wheel          ALL=(ALL)  ALL
```

- 示例2：

```
student ALL=(root)     /sbin/pidof,/sbin/ifconfig
%wheel ALL=(ALL)  NOPASSWD: ALL
```

sudo示例

- 示例3

```
User_Alias  NETADMIN= netuser1,netuser2
Cmnd_Alias  NETCMD = /usr/sbin/ip
NETADMIN ALL=（root） NETCMD
```

- 示例4

```
User_Alias SYSADER=wang,mage,%admins
User_Alias DISKADER=tom
Host_Alias SERS=www.magedu.com,172.16.0.0/24
Runas_Alias OP=root
Cmnd_Alias SYDCMD=/bin/chown,/bin/chmod
Cmnd_Alias DSKCMD=/sbin/parted,/sbin/fdisk
SYSADER SERS=   SYDCMD,DSKCMD
DISKADER ALL=(OP)  DSKCMD
```

sudo示例

- 示例4

```
User_Alias ADMINUSER = adminuser1,adminuser2
Cmnd_Alias ADMINCMD = /usr/sbin/useradd,
/usr/sbin/usermod, /usr/bin/passwd [a-zA-Z]*,
!/usr/bin/passwd root
ADMINUSER ALL=(root) NOPASSWD:ADMINCMD,
PASSWD:/usr/sbin/userdel
```

- 示例5

```
Defaults:wang runas_default=tom
wang ALL=(tom,jerry) ALL
```


- 示例6

```
wang 192.168.175.136,192.168.175.138=(root)
/usr/sbin/,!/usr/sbin/useradd
```

- 示例7

```
wang   ALL=(ALL)  /bin/cat /var/log/message*
```

sudo命令

- #ls -l /usr/bin/sudo
- sudo -i -u wang 切换身份
- sudo [-u user] COMMAND

  -V 显示版本信息等配置信息

  -u user 默认为root

  -l,ll 列出用户在主机上可用的和被禁止的命令

  -v 再延长密码有效期限5分钟,更新时间戳

  -k 清除时间戳，下次需要重新输密码

  -K 与-k类似，还要删除时间戳文件

  -b 在后台执行指令

  -p 改变询问密码的提示符号

  如 -p "password on %h for user %p"

关于马哥教育

- 博客：http://mageedu.blog.51cto.com
- 主页：http://www.magedu.com
- QQ：1661815153，113228115
- QQ群：203585050，279599283

马哥教育
最专业的Linux培训机构

Thank You!