一　加密与解密

1 案例1：加密与解密应用

1.1 问题

本案例要求采用gpg工具实现加/解密及软件签名等功能，分别完成以下任务：

检查文件的MD5校验和

使用GPG实现文件机密性保护，加密和解密操作

使用GPG的签名机制，验证数据的来源正确性

1.2 方案

加密算法主要有以下几种分类：

1.为确保数据机密性算法：

a) 对称加密算法(AES,DES)

b) 非对称加密算法（RSA，DSA）

2.为确保数据完整性算法：

a) 信息摘要（MD5，SHA256，SHA512）

1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：检查文件的MD5校验和

1） 查看文件改动前的校验和，复制为新文件其校验和不变

[root@proxy ~]# vim file1.txt

abcdef

123456779

[root@proxy ~]# cp file1.txt file2.txt

[root@proxy ~]# cat file1.txt > file3.txt

[root@proxy ~]# md5sum file?.txt //文件内容一致，则校验和也不变

b92aa0f8aa5d5af5a47c6896283f3536 file1.txt

b92aa0f8aa5d5af5a47c6896283f3536 file2.txt

b92aa0f8aa5d5af5a47c6896283f3536 file3.txt

2） 对文件内容稍作改动，再次检查校验和，会发现校验和已大不相同

[root@proxy ~]# echo "x" >> file1.txt

[root@proxy ~]# md5sum file?.txt

6be3efe71d8b4b1ed34ac45f4edd2ba7 file1.txt

b92aa0f8aa5d5af5a47c6896283f3536 file2.txt

b92aa0f8aa5d5af5a47c6896283f3536 file3.txt

步骤二：使用GPG对称加密方式保护文件

GnuPG是非常流行的加密软件，支持所有常见加密算法，并且开源免费使用。

1）确保已经安装了相关软件（默认已经安装好了）

[root@proxy ~]# yum -y install gnupg2 //安装软件

[root@proxy ~]# gpg --version //查看版本

gpg (GnuPG) 2.0.22

2） gpg使用对称加密算法加密数据的操作

执行下列操作：

[root@proxy ~]# gpg -c file2.txt

.. ..

根据提示输入两次口令，加密后的文件（自动添加后缀 .gpg）就生成了，传递过程中只要发送加密的文件（比如 file2.txt.gpg）就可以了。

[root@proxy ~]# cat file2.txt.gpg //查看加密数据为乱码

3）使用gpg对加密文件进行解密操作

收到加密的文件后，必须进行解密才能查看其内容。

[root@proxy ~]# gpg -d file2.txt.gpg > file2.txt //解密后保存

gpg: 3DES 加密过的数据

.. .. //根据提示输入正确密码

[root@proxy ~]# cat file2.txt //查看解密后的文件

abcdef

123456779

步骤三：使用GPG非对称加密方式保护文件

非对称加密/解密文件时，UserA（192.168.4.100）生成私钥与公钥，并把公钥发送给UserB（192.168.4.5），UserB使用公钥加密数据，并把加密后的数据传给UserA，UserA最后使用自己的私钥解密数据。

实现过程如下所述。

1）接收方UserA创建自己的公钥、私钥对(在192.168.4.100操作)

[root@client ~]# gpg --gen-key //创建密钥对

… …

请选择您要使用的密钥种类：

(1) RSA and RSA (default) //默认算法为RSA

(2) DSA and Elgamal

(3) DSA (仅用于签名)

(4) RSA (仅用于签名)

您的选择？ //直接回车默认(1)

RSA 密钥长度应在 1024 位与 4096 位之间。

您想要用多大的密钥尺寸？(2048) //接受默认2048位

您所要求的密钥尺寸是 2048 位

请设定这把密钥的有效期限。

0 = 密钥永不过期

<n> = 密钥在 n 天后过期

<n>w = 密钥在 n 周后过期

<n>m = 密钥在 n 月后过期

<n>y = 密钥在 n 年后过期

密钥的有效期限是？(0) //接受默认永不过期

密钥永远不会过期

以上正确吗？(y/n)y //输入y确认

You need a user ID to identify your key; the software constructs the user ID

from the Real Name, Comment and Email Address in this form:

"Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

真实姓名：UserA

电子邮件地址：UserA@tarena.com

注释：UserA

您选定了这个用户标识：

“UserA (UserA) <UserA@tarena.com>”

更改姓名(N)、注释(C)、电子邮件地址(E)或确定(O)/退出(Q)？O //输入大写O确认

您需要一个密码来保护您的私钥。

我们需要生成大量的随机字节。这个时候您可以多做些琐事(像是敲打键盘、移动

鼠标、读写硬盘之类的)，这会让随机数字发生器有更好的机会获得足够的熵数。

gpg: 正在检查信任度数据库

gpg: 需要 3 份勉强信任和 1 份完全信任，PGP 信任模型

gpg: 深度：0 有效性： 1 已签名： 0 信任度：0-，0q，0n，0m，0f，1u

pub 2048R/421C9354 2017-08-16

密钥指纹 = 8A27 6FB5 1315 CEF8 D8A0 A65B F0C9 7DA6 421C 9354

uid UserA (UserA) <UserA@tarena.com>

sub 2048R/9FA3AD25 2017-08-16

注意：生产密钥后当前终端可能会变的无法使用，执行reset命令即可，或者关闭后再开一个终端。

2）UserA导出自己的公钥文件(在192.168.4.100操作)

用户的公钥、私钥信息分别保存在pubring.gpg和secring.gpg文件内：

[root@client ~]# gpg --list-keys //查看公钥环

/root/.gnupg/pubring.gpg

------------------------------

pub 2048R/421C9354 2017-08-16

uid UserA (User A) <UserA@tarena.com>

sub 2048R/9FA3AD25 2017-08-16

使用gpg命令结合--export选项将其中的公钥文本导出：

[root@client ~]# gpg -a --export UserA > UserA.pub

//--export的作用是导出密钥，-a的作用是导出的密钥存储为ASCII格式

[root@client ~]# scp UserA.pub 192.168.4.5:/tmp/

//将密钥传给Proxy

3）UserB导入接收的公钥信息（在192.168.4.5操作）

使用gpg命令结合--import选项导入发送方的公钥信息，以便在加密文件时指定对应的公钥。

[root@proxy ~]# gpg --import /tmp/UserA.pub

gpg: 密钥 421C9354：公钥“UserA (UserA) <UserA@tarena.com>”已导入

gpg: 合计被处理的数量：1

gpg: 已导入：1 (RSA: 1)

4) UserB使用公钥加密数据，并把加密后的数据传给UserA（在192.168.4.5操作）

[root@proxy ~]# echo "I love you ." > love.txt

[root@proxy ~]# gpg -e -r UserA love.txt

无论如何还是使用这把密钥吗？(y/N)y //确认使用此密钥加密文件

//-e选项是使用密钥加密数据

//-r选项后面跟的是密钥，说明使用哪个密钥对文件加密

[root@proxy ~]# scp love.txt.gpg 192.168.4.100:/root //加密的数据传给UserA

4）UserA以自己的私钥解密文件（在192.168.4.100操作）

[root@client ~]# gpg -d love.txt.gpg > love.txt

您需要输入密码，才能解开这个用户的私钥：“UserA (UserA) <UserA@tarena.com>”

2048 位的 RSA 密钥，钥匙号 9FA3AD25，建立于 2017-08-16 (主钥匙号 421C9354)

//验证私钥口令

gpg: 由 2048 位的 RSA 密钥加密，钥匙号为 9FA3AD25、生成于 2017-08-16

“UserA (UserA) <UserA@tarena.com>”

[root@client ~]# cat love.txt //获得解密后的文件内容

I love you.

**步骤四：使用GPG的签名机制，检查数据来源的正确性**

使用私钥签名的文件，是可以使用对应的公钥验证签名的，只要验证成功，则说明这个文件一定是出自对应的私钥签名，除非私钥被盗，否则一定能证明这个文件来自于某个人！

1）在client(192.168.4.100)上，UserA为软件包创建分离式签名

将软件包、签名文件、公钥文件一起发布给其他用户下载。

[root@client ~]# tar zcf log.tar /var/log //建立测试软件包

[root@client ~]# gpg -b log.tar //创建分离式数字签名

[root@client ~]# ls -lh log.tar\*

-rw-rw-r--. 1 root root 170 8月 17 21:18 log.tar

-rw-rw-r--. 1 root root 287 8月 17 21:22 log.tar.sig

[root@client ~]# scp log.tar\* 192.168.4.5:/root //将签名文件与签名传给UserB

2）在192.168.4.5上验证签名

[root@proxy ~]# gpg --verify log.tar.sig log.tar

gpg:于2028年06月07日 星期六 23时23分23秒 CST 创建的签名，使用 RSA，钥匙号 421C9354

gpg: 完好的签名，来自于“UserA (UserA) <UserA@tarena.com>”

.. ..

二　AIDE入侵检测系统

2.1 问题

本案例要求熟悉Linux主机环境下的常用安全工具，完成以下任务操作：

安装aide软件

执行初始化校验操作，生成校验数据库文件

备份数据库文件到安全的地方

使用数据库执行入侵检测操作

2.2 方案

Aide通过检查数据文件的权限、时间、大小、哈希值等，校验数据的完整性。

使用Aide需要在数据没有被破坏前，对数据完成初始化校验，生成校验数据库文件，在被攻击后，可以使用数据库文件，快速定位被人篡改的文件。

2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：部署AIDE入侵检测系统

1）安装软件包

[root@proxy ~]# yum -y install aide

2) 修改配置文件

确定对哪些数据进行校验，如何校验数据

[root@proxy ~]# vim /etc/aide.conf

@@define DBDIR /var/lib/aide //数据库目录

@@define LOGDIR /var/log/aide //日志目录

database\_out=file:@@{DBDIR}/aide.db.new.gz //数据库文件名

//一下内容为可以检查的项目（权限，用户，组，大小，哈希值等）

#p: permissions

#i: inode:

#n: number of links

#u: user

#g: group

#s: size

#md5: md5 checksum

#sha1: sha1 checksum

#sha256: sha256 checksum

DATAONLY = p+n+u+g+s+acl+selinux+xattrs+sha256

//以下内容设置需要对哪些数据进行入侵校验检查

//注意：为了校验的效率，这里将所有默认的校验目录与文件都注释

//仅保留/root目录，其他目录都注释掉

/root DATAONLY

#/boot NORMAL //对哪些目录进行什么校验

#/bin NORMAL

#/sbin NORMAL

#/lib NORMAL

#/lib64 NORMAL

#/opt NORMAL

#/usr NORMAL

#!/usr/src //使用[!]，设置不校验的目录

#!/usr/tmp

步骤二：初始化数据库，入侵后检测

1）入侵前对数据进行校验，生成初始化数据库

[root@proxy ~]# aide --init

AIDE, version 0.15.1

AIDE database at /var/lib/aide/aide.db.new.gz initialized.

//生成校验数据库，数据保存在/var/lib/aide/aide.db.new.gz

2）备份数据库，将数据库文件拷贝到U盘（非必须的操作）

[root@proxy ~]# cp /var/lib/aide/aide.db.new.gz /media/

3）入侵后检测

[root@proxy ~]# cd /var/lib/aide/

[root@proxy ~]# mv aide.db.new.gz aide.db.gz

[root@proxy ~]# aide --check //检查哪些数据发生了变化

三　扫描与抓包