**Morning：**

1、加/解密概述

**-加密算法**

对称加密：加/解密用同一个密钥。DES、AES

非对称加密：公钥、私钥。RSA、DSA

单向加密：根据文件生成位数固定的散列值。MD5、SHA

**-MD5完整性检验**

[root@proxy ~]# md5sum file1

[root@proxy ~]# iptables -nL | md5sum

2、GPG加/解密工具

**-对称加/解密**

[root@proxy ~]# gpg -c data1 #文件加密(输入密码)

[root@proxy ~]# ls data1\*

data1 data1.gpg #加密后的文件

[root@proxy ~]# gpg -d data1.gpg > data2 #文件解密

Tips：对称加密一般用于本机文件加密

**-非对称加/解密**

1）基本过程

1. UserA生成密钥对
2. UserA传递公钥给UserB
3. UserB使用公钥加密数据，并把加密数据传给UserA
4. UserA使用私钥解密

2）基本用法

###密钥对的生成与传递###

[root@proxy ~]# gpg --gen-key

#创建密钥对，需填写密钥对信息、个人信息、私钥使用密码

[root@proxy ~]# gpg --list-keys #查看密钥信息

[root@proxy ~]# gpg -a --export jerry > jerry.pub

#-a使用ASCII格式、--export导出密钥

#jerry为密钥姓名(也可以使用邮箱、唯一编码)

[root@proxy ~]# scp jerry.pub 192.168.4.66:/tmp/ #传递公钥

[root@client ~]# gpg --import /tmp/jerry.pub #导入密钥

###数据加密、传递与解密###

[root@client ~]# gpg -e -r jerry data1.txt

#-e或--encrypt加密数据，-r或--recipient指定密钥

[root@client ~]# ls data1.txt\*

data1.txt data1.txt.gpg #加密后的文件

[root@client ~]# scp data1.txt.gpg 192.168.4.1:/tmp/

#发送加密数据

[root@proxy ~]# gpg -d /tmp/data1.txt.gpg > data1.txt

#-d或--decrypt数据解密

Tips：一般用于网络文件传输的加密

3）软件签名与验证

发文件前(私钥签名) --> 打开前(公钥验证签名) --> 打开

[root@proxy ~]# gpg -b data.tar.gz #创建软件签名

[root@proxy ~]# ls data.tar.gz\*

data.tar.gz data.tar.gz.sig #签名文件与签名

[root@proxy ~]# scp data.tar.gz\* 192.168.4.66:/root/

#发送2个文件(签名文件与签名)

[root@client ~]# gpg --verify data.tar.gz.sig data.tar.gz

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

#验证签名

Tips：用于打开前验证文件的安全性，防止病毒

**Afternoon：**

1、AIDE入侵检测系统

**步骤1：部署AIDE系统**

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

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

@@define DBDIR /var/lib/aide #校验文件目录

database\_out=file:@@{DBDIR}/aide.db.new.gz #新校验文件

database=file:@@{DBDIR}/aide.db.gz #旧校验文件

DATAONLY = p+n+u+g+s+...+sha256 #定义检查项目

/root/ DATAONLY #定义哪些目录进行校验

#批量注释：Ctrl+v -> 下键 -> shift+i -> 输入# -> ESC

#批量去注释：Ctrl+v -> 下键 -> x

**步骤2：初始化校验文件**

[root@proxy ~]# aide --init

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

**步骤3：入侵后检测**

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

[root@proxy ~]# aide --check

2、NMAP扫描

支持多种探测技术：ping扫描、多端口扫描、TCP/IP指纹校验

**nmap [扫描类型] [选项] <扫描目标 ...>**

-sS，TCP SYN扫描(两次握手)

-sT，TCP连接扫描(三次握手、四次断开)

-sU，UDP扫描

-sP，ICMP扫描

-A，目标系统全面分析

[root@proxy ~]# nmap -sP -n 192.168.4.0/24

#扫描是否ping通(可跳过防火墙)，-n跳过dns警告

[root@proxy ~]# nmap -sT 或 -sS 192.168.4.1

#扫描TCP端口，-sS速度更快

[root@proxy ~]# nmap -sU 192.168.4.1

#扫描UDP端口(速度很慢)

[root@proxy ~]# nmap -A 192.168.4.1

3、tcpdump抓包工具

**步骤1：准备vsftpd服务**

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

[root@proxy ~]# systemctl restart vsftpd

[root@client ~]# yum -y install ftp

[root@client ~]# ftp 192.168.4.1

**步骤2：启用tcpdump抓包与分析**

**tcpdump [选项] [过滤条件]**

-i，指定监控的网卡(默认eth0)

-A，转换为ACSII码，方便阅读

-w，将抓包信息保存到文件

-r，从文件读取抓包信息

tcpdump过滤条件：

类型：host、net、port、portrange

方向：src、dst

协议：tcp、udp、ip、wlan、arp、......

多个条件组合：and、or、not

[root@proxy ~]# tcpdump -A -i eth0 -w a.txt tcp port 22 and src host 1.1.1.1

#抓取来自网卡eth0源地址为1.1.1.1且为ssh的包，保存到a.txt

[root@client ~]# ftp 192.168.4.1

Name (192.168.4.1:root):gjq

Password:

[root@proxy ~]# tcpdump -r abc.bao | egrep -i "user | pass"

FTP: USER gjq

FTP: PASS 123456

#分析数据包

**【拓展】抓取Nginx明文账户认证信息**

1）设置nginx用户认证

[root@web\_33 ~]# vim /usr/local/nginx/conf/nginx.conf

auth\_basic "Input Password:";

auth\_basic\_user\_file "/usr/local/nginx/pass";

[root@web\_33 ~]# htpasswd -c /usr/local/nginx/pass gjq

[root@web\_33 ~]# nginx -s reload

2）抓包与分析

[root@web\_33 ~]# tcpdump -A tcp port 80 -w abc.txt

[root@client ~]# firefox 192.168.4.33

[root@web\_33 ~]# tcpdump -r abc.txt | grep -i auth

Authorization: Basic dG9tOjEyMzQ1Ng==

3）使用base64解释编码

[root@web\_33 ~]# echo "dG9tOjEyMzQ1Ng==" | base64 -d

tom:123456

[root@web\_33 ~]# echo "tom:123456" | base64

dG9tOjEyMzQ1Ngo=