**Morning：**

1、系统审计

**-What’s审计**

可以发现并记录违反安全策略的用户及行为；

属于事后行为，不会为系统提供额外的安全保护；

日志记录：日期事件、用户、输用户名密码行为、关键文件的修改行为。

**-常用软件**

jumpserver、audit(系统自带)、公司自主开发boss系统

2、audit审计

**步骤1：部署auditd**

[root@proxy ~]# cat /etc/audit/auditd.conf

log\_file = /var/log/audit/audit.log #审计日志

[root@proxy ~]# systemctl status auditd

Active: active (running)

#auditd.service服务无法停止

**步骤2：定义审计规则**

1）命令语法

auditctl -s #查询服务状态

auditctl -l #查看规则

auditctl -D #删除所有规则

auditctl -w path -p permission [-k keyname] #定义临时规则

[root@proxy ~]# vim /etc/audit/rules.d/audit.rules

-w path -p permission [-k keyname] #定义永久规则

#-w指定文件/目录、-p指定r,w,x,a(属性被修改)，-k定义别名方便查找

2）案例

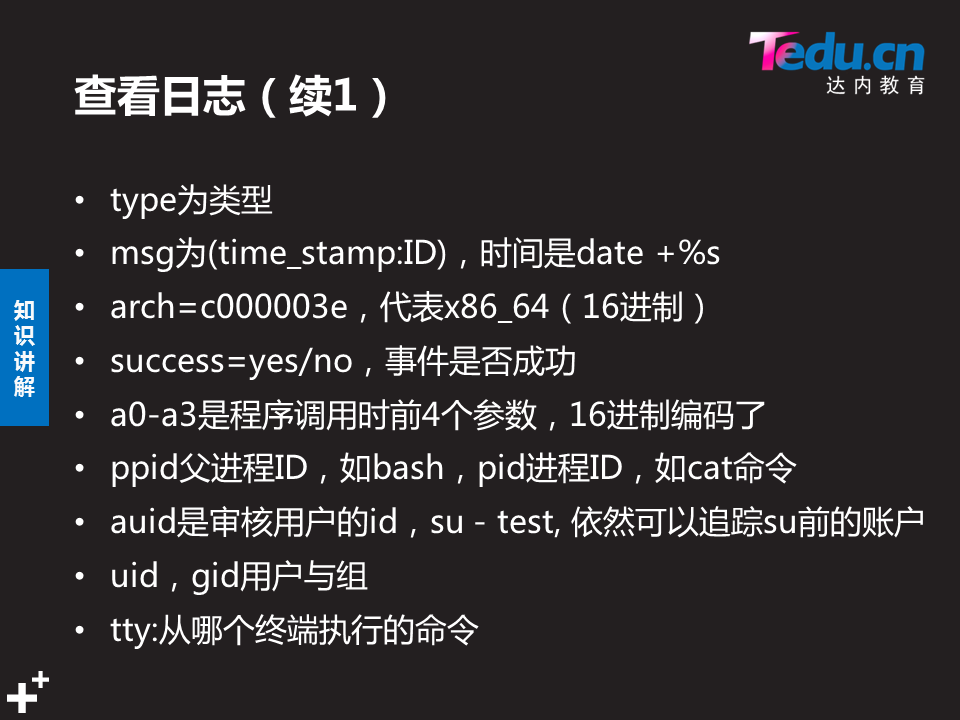
[root@proxy ~]# auditctl -w /root/nb.txt -p rwxa -k nb\_change

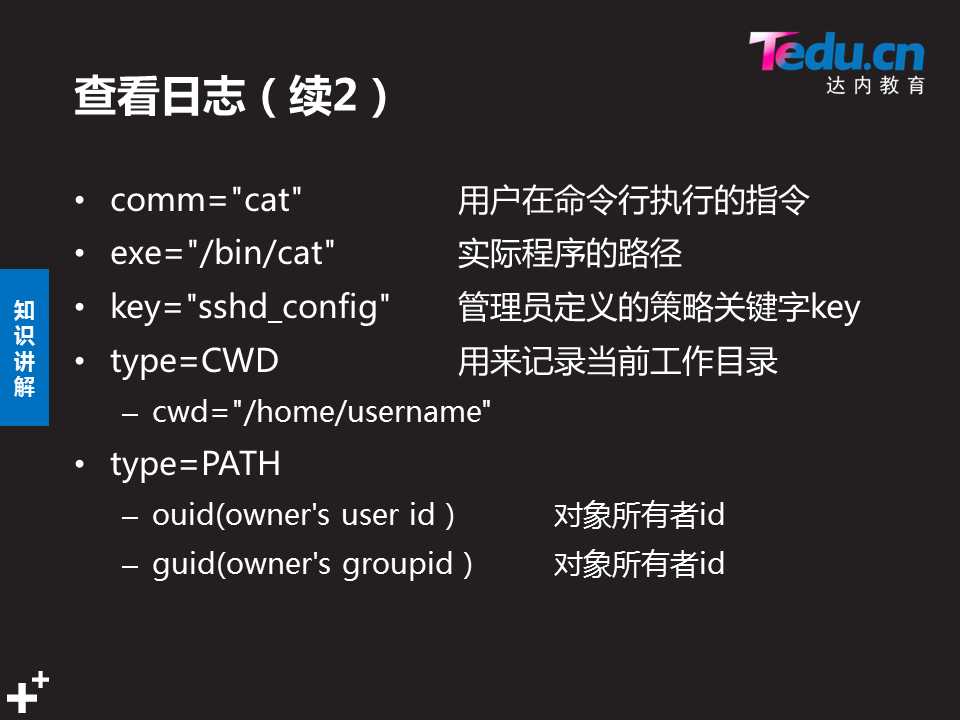
**步骤3：审计日志**

[root@proxy ~]# ausearch -k nb\_change -i

type=SYSCALL msg=audit(2018年12月10日 09:31:35.618:144) : arch=x86\_64 syscall=open success=yes exit=3 a0=0x7ffe8ae8c681 a1=O\_RDONLY a2=0x1fffffffffff0000 a3=0x7ffe8ae8a040 items=1 ppid=1525 pid=1564 auid=root uid=root gid=root euid=root suid=root fsuid=root egid=root sgid=root fsgid=root tty=pts1 ses=4 comm=cat exe=/usr/bin/cat subj=unconfined\_u:unconfined\_r:unconfined\_t:s0-s0:c0.c1023 key=nb\_change

#-i，将日志内容转化可读





Tips：企业一般作二次开发，将日志数据整理成表格、图形等等可视化元素。

**Afternoon：**

1、Nginx安全

**-最小化安装(卸载模块)**

[root@proxy nginx-1.12.2]# ./configure --help | grep "\-\-without\-"

--without-http\_ssi\_module #执行shell脚本

--without-http\_autoindex\_module #无首页，自动列出网页文件

--without-http\_fastcgi\_module #执行php脚本

--without-http\_uwsgi\_module #执行python脚本

--without-mail\_pop3\_module #与邮件服务相关模块

--without-mail\_imap\_module

--without-mail\_smtp\_module

**-修改版本号信息**

浏览器按F12 -> Network -> 访问页面 -> Headers(显示Server信息)

[root@proxy nginx]# vim /usr/local/nginx/conf/nginx.conf

http {

server\_tokens off; #隐藏版本号

...

}

[root@proxy nginx-1.12.2]# vim +48 src/http/ngx\_http\_header...c

static u\_char ngx\_http\_server\_string[] = "Server: Gjq" ...

static u\_char ngx\_http\_server\_full\_string[] = "Server: Gjq" ...

static u\_char ngx\_http\_server\_build\_string[] = "Server: Gjq" ...

#修改Server显示信息，修改安装文件的c源码

[root@proxy nginx-1..]# ./configure && make

[root@proxy sbin]# mv nginx nginx.bak #备份

[root@proxy nginx-1..]# cp objs/nginx /usr/local/nginx/sbin/nginx

#升级安装

**-限制某IP并发量**

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

http{

limit\_req\_zone $binary\_remote\_addr zone=one:10m rate=1r/s;

server{

limit\_req zone=one burst=5;

...

}

}

#limit\_req\_zone分配共享内存，$binary\_remote\_addr用于记录客户IP；

#zone=one存储空间命名为one，:10m空间大小为10M；

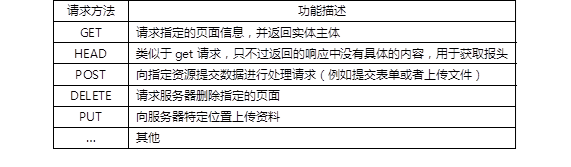
#1M可以存8000个IP信息，10M存80000个；

#rate=1r/s，每秒仅接收1个请求，多余的放入漏斗(排队处理)；

#burst=5漏斗超过5个则报错。

**-拒绝非法请求**

1. 常见HTTP请求方法



1. 仅允许GET | POST

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

http{

server{

...

if ($request\_method !~ ^(GET|POST)$){

return 444;

}

}

}

[root@proxy ~]# curl -i -X GET http://192.168.4.1

[root@client ~]# curl -i -X HEAD http://192.168.4.1

#-i显示所有网页信息(包括header)、-X指定请求服务器的方法

#-X HEAD返回为空

**-防止buffer溢出**

#调整存放客户连接状态信息的缓存大小

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

http{

client\_body\_buffer\_size 1k;

client\_header\_buffer\_size 1k;

client\_max\_body\_size 1k;

large\_client\_header\_buffers 2 1k;

… …

}

2、MySQL数据库安全

**-初始化安全脚本**

[root@proxy ~]# mysql\_secure\_installation

#初始化root密码，清除匿名用户，禁止root远程登陆，删除test数据库

**-密码安全**

[root@proxy ~]# mysqladmin -uroot -p123456 password 'mysql'

mysql> set password for root@'localhost'=password('mysql');

[root@proxy ~]# history -c #清空命令行历史

[root@proxy ~]# > .bash\_history

[root@proxy ~]# > .mysql\_history #清空mysql历史记录

如何解决明文登陆带来的安全问题？

#生产环境一般用navicat、phpMyAdmin图形化软件访问数据库

#管理好历史记录，选择新版本的mysql

#日志、行为审计、使用防火墙设置ACL（仅允许web服务器，其他拒绝）

**-数据备份与还原**

参考：阶段3 -> 1.DBA1 -> DAY05

新方法：对数据库手动加锁 --> 快照(ceph、LVM逻辑卷) ---> rsync/cp将快照数据进行备份 #可以减少锁表时间

**-数据安全**

使用mysql命令远程连接数据库，传输的数据是明文，不安全！

如何解决？

1. 使用SSH登陆服务器，再通过localhost登陆
2. 使用ssl对mysql服务器加密（类似于http+ssl）

3、Tomcat安全

**-隐藏版本信息**

1）修改源码

[root@proxy ~]# yum -y install java-1.8.0-openjdk-devel

[root@proxy ~]# cd /usr/local/tomcat/lib/

[root@proxy lib]# jar -xf catalina.jar

[root@proxy lib]# vim org/apache/catalina/util/ServerInfo...

#修改版本信息，然后重启tomcat服务

[root@proxy ~]# curl 192.168.4.1:8080/test

#访问错误页面，验证header是否隐藏版本信息

2）修改配置文件

[root@proxy ~]# vim /usr/local/tomcat/conf/server.xml

<Connector port="8080" protocol="HTTP/1.1"

...server="Gjq V1.0" />

#重启服务并验证

**-降级启动**

[root@proxy ~]# useradd tomcat

[root@proxy ~]# chown -R tomcat:tomcat /usr/local/tomcat/

[root@proxy ~]# su - tomcat -c "/usr/local/tomcat/bin/startup.sh"

[root@proxy ~]# ps aux | grep tomcat

tomcat 16446 26.2 7.9 ...

#默认使用root启动tomcat，应尽量使用普通用户启动服务

#可以将启动指令添加到/etc/rc.local，开机自启

**-删除默认测试页面**

[root@web1 ~]# rm -rf /usr/local/tomcat/webapps/\*

1. Linux安全之打补丁

**-diff制作补丁文件**

原理：告诉我们怎么修改第一个文件，成为第二个文件

[root@proxy ~]# diff -Nura /root/dir1/ /root/dir2/ > test.patch

#-u用于打补丁(默认只对文本)，让计算机知道哪个文件需要修改

#-r递归对比目录中的所有资源

#-a所有文件视为文本（包括二进制程序）

#-N无文件视为空文件（空文件怎么变成第二个文件）

**-patch打补丁**

[root@proxy ~]# cd ~

[root@proxy ~]# patch -p2 < test.patch #打补丁

[root@proxy ~]# patch -R -p2 < test.patch #还原

Tips：注意制作补丁时的绝对/相对路径，-p2表示将补丁文件的2级目录删掉，/为第一级目录