什么是审计

做了与规则匹配的操作,生成日志记录人,行为

只能记录,不能起保护作用

日志内容:

日期,事件,事件结果,触发事件的用户

所有认证机制的使用都可被操作,如ssh

对关键文件的修改行为

需:audit软件包

Ausearch工具

Audit

主配置文件: /etc/audit/auditd.conf

日志文件: /var/log/audit/audit.log (配置文件中定义的)

1.部署Audit

]#yum -y install audit //安装

]#systemctl start audit**d** //启动,不能用restart

]#一直启动, 手动关不掉, 不可以停,停掉无法记录审计规则

2.查看审计规则

auditctl -l 查看规则

auditctl -s 查看状态

auditctl -D 删除所有规则

3.定义审计规则

+命令行定义(立即生效)

auditctl -w 文件|目录 -p 权限 -k key\_name

权限:

a: 属性变化 r,w,x

-k: 给审计规则起个名字

+写入配置文件(永久)

]# vim /etc/audit/rules.d/audit.rules **#不是主配置文件**

4.Ausearch工具

]# ausearch -k sshd\_config -i

//默认搜索/var/log/audit/audit.log

//根据key搜索日志，-i选项表示以

5.日志内容详解

Time->时间

Success=yes/no 是否执行成功

Comm=执行的命令

Exe=命令的绝对路径

Uid=

Gid=

Uuid=父进程

Tty=从哪个终端执行的命令

#######################################################################################

(服务安全)Nginx优化

1.自动索引(删除不需要的模块)

介绍:

默认安装自动检索模块

--with-http\_autoindex\_module

Server {

autoindex on 开启自动检索

*不安全: 若客户端网址没输入完整,会显示网址末尾下的所有文件,类似于ftp共享*

禁用:

]#./configure --without-http\_autoindex\_module

2.隐藏版本号

*有些黑客精通nginx,若暴露nginx版本号会存在风险*

源码目录下

]#vim nginx-1.12.2/src/http/ngx\_http\_header\_filter\_module.c

删除蓝色内容

49 static u\_char ngx\_http\_server\_string[] = "Server: nginx" CRLF;

50 static u\_char ngx\_http\_server\_full\_string[] = "Server: " NGINX\_VER CRLF;

51 static u\_char ngx\_http\_server\_build\_string[] = "Server: " NGINX\_VER\_BUILD CRLF;

验证 : ]# curl -I <http://192.168.4.5>

3.限制并发

默认会安装

http {

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

同一ip每秒处理一次请求 定义使用10m内存,内存名为one

server {

limit\_req zone=one burst=5;

剩下的请求取5个放到one内存里

一共能处理6个请求,剩下的不处理

1m可存8千个ip地址

Curl -i

1. 拒绝非法请求

实际应用中一般仅需要get和post

#这里，!符号表示对正则取反，~符号是正则匹配符号

#如果用户使用非GET或POST方法访问网站，则retrun返回444的错误信息

server {

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

return 444;

}

]# curl -i -X GET http://192.168.4.5 //正常

]# curl -i -X HEAD http://192.168.4.5 //不正常

1. 防止buffer溢出

当客户端连接服务器时，服务器会启用各种缓存，用来存放连接的状态信息。

如果攻击者发送大量的连接请求，而服务器不对缓存做限制的话，内存数据就有可能溢出（空间不足）。

修改Nginx配置文件，调整各种buffer参数，可以有效降低溢出风险。

http {

client\_body\_buffer\_size 1K;

client\_header\_buffer\_size 1k;

client\_max\_body\_size 16k;

large\_client\_header\_buffers 4 4k;

################################################################################

数据库安全

Mariadb

初始化脚本

匿名账号登录:

首次启动, 任何一个用户都能登录数据库,不管用户存不存在

能看到虚拟库和text库

虽然对所有库表没有权限,但对text库有完全权限

1.初始化安全脚本

]# mysql\_secure\_installation #初始化脚本

输入新密码

Remove anonymous users（删除匿名账户）

Disallow root login remotely?（禁止root远程登录）

Remove test database（删除测试数据库）

Reload privilege（刷新权限）

1. 密码安全

修改mysql密码的若干方法(至少3中)

]#mysqladmin -hlocalhost -p password “密码”

>set password for root@’localhost’ identified by “密码”

>update mysql.user set password=password(“密码”)

删除历史命令

**里面记录了明文密码**

删除记录mysql历史命令的文件: /root/.mysql\_history

删除记录系统命令行历史命令的文件: /root/.bash\_history

删除历史命令保存: ]#history -c

]# grep 1000 /etc/profile

HISTSIZE=1000

另外数据库还有一个binlog日志里也有明文密码（5.6版本后修复了）。

解决: 选择合适的版本5.6以后的版本，日志，行为审计（找到行为人），使用防火墙从

TCP层设置ACL（禁止外网接触数据库）

3.数据安全: 备份数据、添加授权用户访问数据、tcpdump抓包

数据传输是没有加密的

为什么: 数据在内网

#############################################################################

Tomcat优化

安装jdk环境

]# yum -y install java-1.8.0-openjdk

安装

]#mv apache-tomcat-8.0.30 /usr/local/tomcat

测试

]# curl 192.168.4.59:8080/text.html

一共有三个端口, grep java :

8005 8009 8080

1. 隐藏版本信息

server.xml ---->命令行curl -**I** 信息显示

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

connectionTimeout="20000" redirectPort="8443" **server="jacob"** />

lib/org/apache/catalina/util/ServerInfo.properties --->网页版本信息显示

server.info=版本

server.number=1.0.1.0

server.built=时间

jar解压得到org

]#yum -y install **java-1.8.0-openjdk-devel**

]# cd /usr/local/tomcat/lib/

]#jar -xf catalina.jar

META-INF org

/dev/random

2.删除默认的主页面

*打开会显示tom猫,一眼可知为tomcat*

]# rm -rf /usr/local/tomcat/webapps/\*

1. 降权运行

*以root身份运行tomcat, 若黑客有了java控制权就有了root权限*

*]#ps aux | root*

*root ............*

]#useradd tomcat #创建tomcat用户

]#chown tomcat:tomcat /var/local/tomcat/ #修改tomcat工作目录所有文件u:g

]#su - tomcat -c “/usr/local/tomcat/bin/startup.sh” #tomcat身份运行服务

3个端口

###############################################################################

安全之打补丁

存放版本1和版本2之间差别信息的文件为补丁文件

**.**patch: 补丁文件

补丁文件里记录了路径,如何操作

创建补丁

diff逐行比较

–告诉我们怎么修改旧版本文件得到新版本文件

选项:

-u:

-r: 递归比较目录中的所有资源(比较目录时使用)

-a:所有文件视为文本(比如有二进制文件,其他类型的文件等等)

-N:无文件视为空文件(旧目录里没有新目录里的文件)

]# diff -u 文件1 文件2 > 补丁文件名.patch

]# diff -urNa 旧版本目录 新版本目录 > 补丁文件名.patch

树型目录结构

]# yum -y install tree

打补丁

patch -p数字 < 补丁文件

数字: 路径重复的数,(当前路径与补丁记录的路径作对比,重复几个)

还原旧版本,反向修复

patch -RE < 补丁文件

R:反向修复(reverse)

E:修复后若文件为空,则删除

打补丁时所在的路径+补丁文件内记录的路径

补丁目录 客户端放置目录的路径必须与源一致

必须进入被补丁文件所在???

当给文件打补丁时,补丁文件里没有路径,需将被补丁的文件及补丁文件放在一起,并进入到两者所在位置 p0

当给目录打补丁时,补丁文件里有路径, 需进入被补丁文件的路径范围,重复几个路径就减几