监控与服务安全

监控的目的

----报告系统运行状况

每一部分必须同时监控

内容包括吞吐量,反应时间,使用率等

---提前发现问题

进行服务器性能调整前,知道调整什么

找出系统的瓶颈在什么地方

监控的资源类别

---公开数据

Web FTP SSH 数据库等应用服务

TCP或UDP端口

---私有数据

CPU 内存 磁盘 网卡流量等使用信息

用户 进程等运行信息

自动化监控系统

Cacti 基于SNMP协议的监控软件,强大的绘图能力

Nagios

基于Agent监控,强大的状态检查与报警机制

插件极多,自己写监控脚本潜入到Nagios非常方便

Zabbix 基于多种监控机制,支持分布式监控

Zabbix简介

-Zabbix是一个高度集成的监控解决方案

-可以实现企业级的开源分布式监控

-Zabbix通过C/S模式采集监控数据

-Zabbix通过B/S模式实现Web管理(B:浏览器)

监控拓扑

---监控服务器

监控服务器可以通过SNMP或Agent采集数据

数据可以写入MySQL Oracle等数据库中

服务器使用LNMP实现Web前端的管理

---被监控主机

被监控主机需要安装Agent

常见的网络设备一般支持SNMP

主配置文件 /usr/local/etc/zabbix\_server.conf

默认端口 10051

日志文件 /tmp/zabbix\_server.log

"/usr/local/nginx/html/conf/zabbix.conf.php" zabbix保存配置信息的文件

整体步骤

1部署Nginx (开启php模块 添加fastcgi参数)

2安装mariadb 创建zabbix用户 zabbix库 字符集为utf8 (安装zabbix时涉及到--with-mysql=/usr/bin/mysql\_config)

3安装zabbix 修改 server.conf 配置文件 (将frontends/php/下的文件全部拷贝到html目录下 导入/database/mysql/下的3张表 依次是schema images data )

4.安装php php-fpm php-mysql php-gd php-xml php-ldap php-bcmath php-mbstring 相关软件包,修改 /etc/php.ini相关参数

5.创建zabbix用户 开启服务

部署监控服务

1.安装LNMP环境(Zabbix监控管理控制台需要通过Web页面展示出来，并且还需要使用MySQL来存储数据，因此需要先为Zabbix准备基础LNMP环境。)

[root@zabbixserver ~]# yum -y install gcc pcre-devel openssl-devel

[root@zabbixserver ~]# tar -xf nginx-1.12.2.tar.gz

[root@zabbixserver ~]# cd nginx-1.12.2

[root@zabbixserver nginx-1.12.2]# ./configure --with-http\_ssl\_module

[root@zabbixserver nginx-1.12.2]# make && make install

[root@zabbixserver ~]# yum -y install php php-mysql mariadb mariadb-devel mariadb-server

2.修改Nginx配置文件(配置Nginx支持PHP动态网站，因为有大量PHP脚本需要执行，因此还需要开启Nginx的各种fastcgi缓存，加速PHP脚本的执行速度)

!!fastcgi缓存不能写在server中!!

vim /usr/local/nginx/conf/nginx.conf

… …

http{

… …

fastcgi\_buffers 8 16k; //缓存php生成的页面内容，8个16k

fastcgi\_buffer\_size 32k; //缓存php生产的头部信息

fastcgi\_connect\_timeout 300; //连接PHP的超时时间

fastcgi\_send\_timeout 300; //发送请求的超时时间

fastcgi\_read\_timeout 300; //读取请求的超时时间

location ~ \.php$ {

root html;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

include fastcgi.conf;

}

3）启动服务(启动Nginx、PHP-FPM、MariaDB服务，关闭SELinux与防火墙。)

[root@zabbixserver ~]# systemctl start mariadb

[root@zabbixserver ~]# systemctl start php-fpm

[root@zabbixserver ~]# ln -s /usr/local/nginx/sbin/nginx /sbin/nginx

[root@zabbixserver ~]# nginx

[root@zabbixserver ~]# firewall-cmd --set-default-zone=trusted

[root@zabbixserver ~]# setenforce 0

4)部署监控服务器Zabbix Server(1）源码安装Zabbix Server 多数源码包都是需要依赖包的，zabbix也一样，源码编译前需要先安装相关依赖包。)

[root@zabbixserver lnmp\_soft]# yum -y install net-snmp-devel curl-devel //安装相关依赖包

[root@zabbixserver lnmp\_soft]# yum -y install libevent-devel-2.0.21-4.el7.x86\_64.rpm //注意libevent-devel这个软件包在lnmp\_soft目录下有提供

[root@zabbixserver lnmp\_soft]# tar -xf zabbix-3.4.4.tar.gz

[root@zabbixserver lnmp\_soft]# cd zabbix-3.4.4/

[root@zabbixserver zabbix-3.4.4]# ./configure --enable-server // --enable-server安装部署zabbix服务器端软件

--enable-proxy // --enable-proxy安装部署zabbix代理相关软件

--enable-agent // --enable-agent安装部署zabbix被监控端软件

--with-mysql=/usr/bin/mysql\_config // --with-mysql配置mysql\_config路径

--with-net-snmp // --with-net-snmp允许zabbix通过snmp协议监控其他设备

--with-libcurl // --with-libcurl安装相关curl库文件，这样zabbix就可以通过curl连接http等服务，测试被监控主机服务的状态

[root@zabbixserver zabbix-3.4.4]# make && make install

5)初始化Zabbix 创建数据库，上线Zabbix的Web页面

[root@zabbixserver ~]# mysql

mysql> create database zabbix character set utf8; //创建数据库，支持中文字符集(必须是支持中文字符集 否则导入不进去)

mysql> grant all on zabbix.\* to zabbix@'localhost' identified by 'zabbix'; //创建可以访问数据库的账户与密码

[root@zabbixserver ~]# cd lnmp\_soft/zabbix-3.4.4/database/mysql/

[root@zabbixserver mysql]# mysql -uzabbix -pzabbix zabbix < schema.sql

[root@zabbixserver mysql]# mysql -uzabbix -pzabbix zabbix < images.sql

[root@zabbixserver mysql]# mysql -uzabbix -pzabbix zabbix < data.sql

//刚刚创建是空数据库，zabbix源码包目录下，有提前准备好的数据

//使用mysql导入这些数据即可（注意导入顺序）

[root@zabbixserver ~]# cd lnmp\_soft/zabbix-3.4.4/frontends/php/

[root@zabbixserver php]# cp -r \* /usr/local/nginx/html/

[root@zabbixserver php]# chmod -R 777 /usr/local/nginx/html/\*

//将web页面放入Nginx页面目录

6)安装PHP相关依赖 并设置相关配置

[root@zabbixserver ~]# cd /root/

[root@zabbixserver ~]# yum -y install php-gd php-xml php-ldap php-bcmath php-mbstring

[root@zabbixserver ~]# vim /etc/php.ini

date.timezone = Asia/Shanghai //设置时区

max\_execution\_time = 300 //最大执行时间,秒

post\_max\_size = 32M //POST数据最大容量

max\_input\_time = 300 //服务器接受数据的时间限制

[root@zabbixserver ~]# systemctl restart php-fpm

7)开启服务并初始化配置

[root@zabbixserver ~]#zabbix\_server

[root@zabbixserver ~]#firefox 192.168.2.5/index.php //web初始化配置

部署被监控主机Zabbix Agent

1)源码安装Zabbix agent软件

[root@zabbixclient\_web1 ~]# yum -y install gcc pcre-devel

[root@zabbixclient\_web1 ~]# tar -xf zabbix-3.4.4.tar.gz

[root@zabbixclient\_web1 ~]# cd zabbix-3.4.4/

[root@zabbixclient\_web1 zabbix-3.4.4]# ./configure --enable-agent //只开启此功能即可

[root@zabbixclient\_web1 zabbix-3.4.4]# make && make install

2）修改agent配置文件，启动Agent

[root@zabbixclient\_web1 ~]#vim /usr/local/etc/zabbix\_agentd.conf

Server=127.0.0.1,192.168.2.5 //谁可以监控本机（被动监控模式）

ServerActive=127.0.0.1,192.168.2.5 //谁可以监控本机（主动监控模式）

[root@zabbixclient\_web1 ~]# useradd -s /sbin/nologin zabbix

[root@zabbixclient\_web1 ~]# zabbix\_agentd //启动agent服务

配置及使用Zabbix监控系统(添加被监控的服务器)

1.web登录zabbix 使用管理员admin登录

2.通过Configuration（配置）-->Hosts（主机）-->Create Host（创建主机)

添加被监控主机时，需要根据提示输入被监控Linux主机的主机名称（最好与电脑的主机名一致，但也允许不一致）、主机组、IP地址

3.为被监控主机添加监控模板

Zabbix通过监控模板来对监控对象实施具体的监控功能，根据模板来定义需要监控哪些数据，对于Linux服务器的监控，Zabbix已经内置了相关的模板（Template OS Linux），选择模板并链接到主机即可

4.查看监控数据

查看监控数据，登录Zabbix Web控制台，点击Monitoring(监控中)—> Latest data(最新数据)，正过滤器中填写过滤条件，根据监控组和监控主机选择需要查看哪些监控数据

自定义Zabbix监控项目(zabbix支持自己写的监控脚本添加到模板中)

1）被监控主机创建自定义key

自定义key语法格式为：UserParameter=自定义key名称,命令

[root@zabbixclient\_web1 ~]# vim /usr/local/etc/zabbix\_agentd.conf

UnsafeUserParameters=1 //开启自定义功能

# Include=/usr/local/etc/zabbix\_agentd.userparams.conf

#Include=/usr/local/etc/zabbix\_agentd.conf.d/

# Include=/usr/local/etc/zabbix\_agentd.conf.d/\*.conf

//加载配置目录 3选1

[root@zabbixclient\_web1 ~]# cd /usr/local/etc/zabbix\_agentd.conf.d/ //进入对应目录

[root@zabbixclient\_web1 zabbix\_agentd.conf.d]# vim count.line.passwd

UserParameter=count.line.passwd,wc -l /etc/passwd | awk ' {print $1} '

////自定义key语法格式:UserParameter=自定义key名称,命令

////UserParameter=key[\*],<command> (一般用于脚本)key里的所有参数，都会传递给后面命令的位置变量

////UserParameter=ping[\*],echo $1 $1=[\*]

////严格区分大小写! U P必须大写

2）测试自定义key是否正常工作

[root@zabbixclient\_web1 ~]# killall zabbix\_agentd

[root@zabbixclient\_web1 ~]# zabbix\_agentd //重启agent服务

[root@zabbixclient\_web1 ~]# zabbix\_get -s 127.0.0.1 -k count.line.passwd

///若自定义key语法格式 UserParameter=key[\*],<command> 则 -k 'key[\*]' 'nginx.status[accepts]'

3)创建监控模板

登录Zabbix Web监控控制台，通过Configuration(配置)-->Template(模板)-->Create template(创建模板)，填写模板名称，新建模板群组

创建完成模板后，默认模板中没有任何应用、项目、触发器、图形等资源。这里需要点击模板后面的Application（应用集）链接打开创建应用的页面

依次创建应用集 监控项

!!! 监控项 中的key值 与 被监控的上自定义创建的必须一致!!!

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

Zabbix报警机制

概念介绍

--自定义的监控项默认不会自动报警

--首页也不会提示错误

--需要配置触发器与报警动作才可以自动报警

触发器

--表达式,如内存不足300M,用户超过30个等

--当触发条件发生后,会导致一个触发事件

--触发事件会执行某个动作

动作

--触发器的条件被触发后的行为

--可以是发送邮件,也可以是重启某个服务等

1）创建触发器

通过Configuration--> Templates，找到我们之前创建的count.line.passwd模板，点击模板后面的triggers

2）触发器表达式

创建触发器时需要定义表达式，触发器表达式（Expression）是触发异常的条件，触发器表达式格式如下：

{<server>:<key>.<function>(<parameter>)}<operator><constant>

{主机：key.函数(参数)}<表达式>常数

在如图-2所示的蓝色方框中编写触发器表达式，可以直接手写，也可以通过add选择表达式模板。

{web1:system.cpu.load[all,avg1].last(0)}>5 //0为最新数据

如果web1主机最新的CPU平均负载值大于5，则触发器状态Problem

{vfs.fs.size[/,free].max(5m)}<10G //5m为最近5分钟

根分区，最近5分钟的最大容量小于10G，则状态进入Problem

{vfs.file.cksum[/etc/passwd].diff(0)}>0 //0为最新数据

最新一次校验/etc/passwd如果与上一次有变化，则状态进入Problem

大多数函数使用秒作为参数，可以使用#来表示其他含义（具体参考表-1）。

avg, count, last, min and max 等函数支持额外的第二个参数time\_shift（时间偏移量），这个参数允许从过去一段时间内引用数据。

3）配置触发器

设置触发器名称 选择触发器报警级别 点击add添加表达式，填写表达式：监控项为账户数量，最近300秒账户数量大于26

4)设置邮件 创建Media

通过Administration（管理）-->Media Type（报警媒体类型）-->选择Email（邮件）

5)为用户添加Media

在Administration（管理）-->Users（用户）中找到选择admin账户

在弹出的界面中选择Media（报警媒介）菜单-->点击Add(添加)报警媒介

6)创建Action动作

Action（动作）是定义当触发器被触发时的时候，执行什么行为。

通过Configuration（配置）-->Actions（动作）-->Create action（创建动作）

配置Action动作的触发条件

填写Action动作的名称，配置什么触发器被触发时会执行本Action动作（账户数量大于26）

配置Action动作的具体行为

配置动作的具体操作行为（发送信息或执行远程命令），无限次数发送邮件，60秒1次(操作中步骤持续时间使用=60)，发送给Admin用户

概述

自动发现 :当zabbix需要监控的设备越来越多,手动添加监控设备越来越有挑战,此时,可以考虑使用自动发现功能

自动发现可以实现: 发现主机,添加主机,添加主机到组,连接模板等

自动发现流程

--创建自动发现规则

--创建Action动作(发现主机后自动执行什么动作)

--通过动作,执行添加主机,链接模板到主机等操作

主动监控

主动和被动都是对 被监控端 主机而言的

默认zabbix采用的是被动监控

--被动监控:server向Agent发起连接,发送监控key,Agent接受请求,响应监控数据(简单的来说就是服务端主动与被监控段发送请求连接)

--主动监控:Agent向Server发起连接,Agent请求需要检测的监控项目列表,Server响应Agent发送一个items列表,Agent确认收到监控列表,TCP连接完成,会话关闭,Agent开始周期性的收集数据

区别:

server不用每次需要数据都连接Agent,Agent会自己收集数据并处理数据,server仅需要保存数据即可

优势:

当监控主机达到一定量级后,zabbix服务器会越来越慢

此时,可以考虑使用主动监控,释放服务器的压力

另外,zabbix也支持分布式监控,也是可以考虑的方案

主动模式 只有进程没有端口

添加主机时,名称必须是被监控配置文件中设置的hostname值

被监控端配置

[root@zabbixclient\_web2 ~]# vim /usr/local/etc/zabbix\_agentd.conf

#Server=127.0.0.1,192.168.2.5 //注释该行，允许谁监控本机

StartAgents=0

//被动监控时启动多个进程

//设置为0，则禁止被动监控，不启动zabbix\_agentd服务

ServerActive=192.168.2.5 //允许哪些主机监控本机（主动模式），一定要取消127.0.0.1

Hostname=zabbixclient\_web2

//告诉监控服务器，是谁发的数据信息

//一定要和zabbix服务器配置的监控主机名称一致（后面设置）

RefreshActiveChecks=120 //默认120秒检测一次

UnsafeUserParameters=1 //允许自定义key

Include=/usr/local/etc/zabbix\_agentd.conf.d/

[root@zabbixclient\_web2 ~]# killall zabbix\_agentd //关闭服务

[root@zabbixclient\_web2 ~]# zabbix\_agentd //启动服务

web监控端配置

通过Configuration（配置）-->Templates（模板）-->选择Template OS Linux -->全克隆，克隆该模板，新建一个新的模板

修改模板中的监控项目的监控模式

将模板中的所有监控项目全部修改为主动监控模式，通过Configuration（配置）-->Templates（模板）-->选择新克隆的模板，点击后面的Items（监控项）-->点击全选，选择所有监控项目，点击批量更新，将类型修改为：Zabbix Agent（Active主动模式）

禁用部分监控项目

批量修改监控项的监控模式后，并非所有监控项目都支持主动模式，批量修改后，会发现有几个没有修改主动模式成功，说明，这些监控项目不支持主动模式，关闭即可。

在Zabbix监控服务器，添加被监控的主机（主动模式），设置主机名称：zabbixclient\_web2 （必须与被监控端的配置文件Hostname一致），将主机添加到Linux servers组，IP地址修改为0.0.0.0，端口设置为0

为主机添加监控模板，选择刚刚创建的模板（主动模式），添加链接模板到主机

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

linux基本防护

用户账号安全

设置账号有效期

chage -d 0 ,强制修改密码

chage -E yyyy-mm-dd,指定失效日期(-1取消)

账号的锁定/解锁

passwd -S user //查看用户密码状态

passwd -l user //锁定账号

passwd -u user //解锁

强制定期修改密码

配置文件/etc/login.defs

对新建的用户有效

主要控制属性

PASS\_MAX\_DAYS

PASS\_MIN\_DAYS

PASS\_WARN\_AGE

伪装登录提示

配置文件/etc/issue /etc/issue.net

分别用于本地和远程登录

默认会提示内核,系统版本信息

文件系统安全

程序和服务控制

禁用非必要的系统服务

使用systemctl工具 rh7

service 服务 start rh6

chkconfig 服务 on/off rh6

锁定/解锁保护文件

ext3/ext4文件属性控制

chattr (+-=) i//不可变 a//仅可追加重定向写入

lsattr 文件 //查看文件特殊属性

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

用户切换与提权

su - user

su - -C '命令' user

安全日志/var/log/secure

用户提权:让普通用户登录系统后有执行root命令的权限

主配置文件/etc/sudoers

配置提权 visudo或者vim /etc/sudoers(需要wq!强制保存退出)

格式

用户 主机列表=命令列表

root ALL=(ALL) ALL //(ALL)是目标身份,缺省时表示root

root用户 所有主机名 所有命令 //命令要用绝对路径,多个命令用逗号分隔,命令前加!表示禁止使用改命令

%wheel ALL=(ALL) NOPASSWD:ALL //在wheel组里的所有用户可以无需验证使用所有命令

执行提权命令 sudo 提权命令

查看提权命令 sudo -l

修改全局配置,启用日志

Defaults logfile="/var/log/sudo" //文件不存在会自动创建

sudo别名设置

提高可重用性,易读性

简化配置,使记录更有条理

visudo

User\_Alias MYUSER=yaya,jing,plj //定义用户别名 别名名称必须大写

Host\_Alias MYSER=localhost,pxcnode71 //定义主机别名 别名名称必须大写

Cmnd\_Alias MYCMD=命令1,命令2,.... //定义命令别名 别名名称必须大写

调用别名做sudo提权

MYUSER MYSER=MYCMD

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

ssh防护

概述

存在的安全隐患

密码嗅探,键盘记录

暴力枚举账号,猜解密码

常见的防护措施

用户限制,黑白名单

更改验证方式(密码>密钥对)

防火墙

ssh基本安全配置

配置文件/etc/ssh/sshd\_config

Port 3389 //改用非标准端口

Protocol 2 //启用SSH V2版本

ListenAddress 192.168.

PermitRootLogin no //禁止root登录

UseDNS no //不解析客户机地址

LoginGraceTime 1m //登录限时

MaxAuthTries 3 //每连接最多认证次数

ssh黑白名单配置

DenyUsers user1 user2 ...

AllowUsers user1@host user2@host...

DenyGroup group1 group2 ...

AllowGroup group1 group2 ...

ssh密钥对认证

sshd验证方式控制

口令验证

检查登录用户的口令是否一致

密钥验证

检查客户端私钥与服务器上的公钥是否匹配

PasswordAuthentication yes //支持口令

PubkeyAuthentication yes //支持密钥

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

SElinux策略集

SELINUXTYPR=targeted

推荐,仅保护最常见/关键的网络服务,其他不限制

主要软件包:

selinux-policy,selinux-policy-targeted,libselinux-utils,liselinux-utils,coreutils,policycoreutils

sestatus //查看selinux状态

SElinux模式控制

修改kernel启动参数

添加selinux=0禁用

添加enforcing=0设置SELinux为允许模式

修改/etc/selinux/config配置文件 //要先创建/.autorelabel 文件

设置SELINUX=disabled禁用

设置SELINUX=permissive 宽松

设置SELINUX=enforcing 强制

安全上下文

ls -Z 文件

ls -ldZ 文件

用户:角色:访问类型:选项.... //只需要关注访问类型

修改安全上下文

chcon

-t 指定访问类型

-R 递归修改

移动的文件,原有的上下文属性不变

复制的文件,自动继承目标位置的上下文

创建新的文件或目录 自动继承父目录的上下文

重置安全上下文

使用restorecon工具

恢复为所在位置的默认上下文属性

-R 递归修改

/.autorelabel 文件

下次重启后全部重置

调整SELinux布尔值

设置服务功能的开关

查看布尔值 getsebool -a

修改布尔值 setsebool -P 选项 值

服务端运行ftp服务

yum -y install vsftpd

vim /etc/vsftpd/vsftpd.conf

anonymous\_enable=YES

mkdir /var/ftp/share

chmod 777 /var/ftp/share

systemctl start vsftpd

cp /etc/passwd /var/ftp/share

setenforce 1

setsebool -P ftpd\_anon\_write on

setsebool -P ftpd\_full\_access on

客户端连接ftp服务

yum -y install ftp

ftp 192.168.4.71

用户名:ftp

密码:"空"

下载:get

查看:ls

切换目录:cd

切换本地目录:lcd

上传:put

退出:bye

日志排错

安装selinux日志软件

yum -y install setroubleshoot-server

reboot //重启系统

/var/log/messages //日志文件

grep -i 'setroubleshoot' /var/log/messages | tail -1 //查看报错内容

sealert -l 0c0c6daf-163c-4e07-b4c6-c55f86241796 //运行日志里让你运行的命令

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

加密与解密

目的:

确保数据的机密性

对称加密:加密/解密用同一个密钥

DES AES

非对称加密:加密/解密用不同的密钥

RSA DSA

保护信息的完整性

信息摘要:基于输入的信息生成长度较短,位数固定的散列值

Hash散列技术

MD5 SHA

只要源文本不变,生成的散列值不变

md5sum校验工具

md5sum /文件/

GPG加/解密工具

简介

GnuPG,GNU Privacy Guard

http://www.gnupg.org/

最流行的数据加密,数字签名工具软件

gpg --version //查看信息

gpg 软件包: gnupg

对称加/解密

基本用法

加密:--symmetric 或 -c

gpg -c /文件/

file /文件\*

解密:--decrypt 或 -d

gpg -d 文件 > 文件

非对称加/解密

基本用法

B:创建密钥对:--gen-key

为了加快创建的速度,要把/dev/random 改个名

ln -s /dev/urandom /dev/random 用urandom 替换random

B:导出公钥:--export,--armor 或 -a

gpg --list-keys //查看公钥信息

gpg --export -a userb > /tmp/userb.pub //导出公钥文件

A:导入公钥:--import

gpg --import /tmp/userb.pub //导入公钥

A:加密:--encrypt 或 -e

gpg -e -r userb /文件1/ //用公钥加密文件

A:指定目标用户:--recipient 或 -r

B:解密:--decrypt 或 -d

gpg -d /文件1.gpg > /文件2 //解密文件1并保存为文件2

gpg软件签名与验证

过程

软件官方用私钥对软件包执行数字签名

用户下载软件包,软件官方的公钥

用官方公钥验证软件包签名,确保数据来源正确

建立签名文件

分离式签名:--detach-sign或-b

gpg -b /tmp/软件包 //签名

验证签名

--verify

gpg --import /tmp/userb.pub //导入公钥

gpg --verify /tmp/软件包 //验证签名

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

AIDE入侵检测系统

安装软件

AIDE(Advanced intrusion detection environment)

用于检测入侵

yum -y install aide

修改配置文件

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

FIPSR = p+i+n+u+g+s+m+c+acl+selinux+xattrs+sha256 //检测规则(不只这一种规则)

/boot/ CONTENT\_EX // 目录 调用的检测规则

/bin/ CONTENT\_EX

/sbin/ CONTENT\_EX

/lib/ CONTENT\_EX

/lib64/ CONTENT\_EX

/opt/ CONTENT //检测项(!开头的是不检测项)

初始化检测

在没有被攻击入侵前

根据配置文件,对数据进行校验操作 //用于和之后每次检测作对比

aide --inite

备份数据库

在被入侵前,将校验的数据库文件被分到安全的地方

mv /var/lib/aide/aide.db.new.gz /media/

执行入侵检查

将之前备份的校验数据库文件还原

cp /media/aide.db.new.gz /var/lib/aide/aide.db.gz

根据数据库执行入侵检测

aide --check

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

扫描与抓包

扫描 Scan,主动探测 Sniff,被动监听/嗅探 Capture,数据包捕获(抓包)

NMAP 简介

一款强大的网络探测利器工具

支持多种探测技术

ping扫描

多端口扫描

tcp/ip指纹校验...

NMAP基本用法

nmap [扫描类型] [选项] <扫描目的...>

常用的扫描类型

-- -sS TCP SYN扫描(半开)

-- -sT TCP连接扫描(全开)

-- -sU UDP扫描

-- -sP ICMP扫描 (ping的时候使用icmp协议)

-- -A 目标系统全面分析

-- -p 端口号

-- -n 检查主机存活状态

抓包 tcpdump

基本用法

tcpdump [选项] [过滤条件]

常见的监控选项

-i 指定监控的网络接口

-A 转换为ASCII码,以方便阅读

-W 将数据包信息保存到指定文件

-r 从指定文件读取数据包信息

-c 定义抓包个数

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

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

审计的目的是基于事先配置的规则生成日志，记录可能发生在系统上的事件（正常或非正常行为的事件），审计不会为系统提供额外的安全保护，但她会发现并记录违反安全策略的人及其对应的行为。

审计能够记录的日志内容：

日期与事件以及事件的结果

触发事件的用户

所有认证机制的使用都可以被记录，如ssh等

对关键数据文件的修改行为等都可以被记录

软件包 audit

配置文件 /etc/audit/auditd.conf

日志文件 /var/log/audit/audit.log

常用命令

auditctl -s //查询状态

auditctl -l //查看规则

auditctl -D //删除所有规则

ausearch -k sshd\_config -i //根据key搜索日志，-i选项表示以交互式方式操作

auditctl -w path -p permission -k key\_name

aureport //生成审计报告

定义临时文件系统规则：

#语法格式：auditctl -w path -p permission -k key\_name

# path为需要审计的文件或目录

# 权限可以是r,w,x,a(文件或目录的属性发生变化) a=属性

# Key\_name为可选项，方便识别哪些规则生成特定的日志项

如果需要创建永久审计规则，则需要修改规则配置文件： etc/audit/rules.d/audit.rules

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

优化Nginx服务的安全配置

开启自动索引

server{

autoindex on; //开启自动索引

禁用自动索引

# ./configure \

>--without-http\_autoindex\_module \ //禁用自动索引文件目录模块

修改版本信息，并隐藏具体的版本号

显示了使用的软件为nginx的相关信息

[root@proxy nginx-1.12]# curl -I http://192.168.4.5 //查看版本信息验证

修改源码

[root@proxy nginx-1.12]# vim +48 src/http/ngx\_http\_header\_filter\_module.c

//注意：vim这条命令必须在nginx-1.12源码包目录下执行！！！！！！

//该文件修改前效果如下：

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

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

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

//下面是我们修改后的效果：

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

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

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

//修改完成后，再去编译安装Nignx，版本信息将不再显示为Nginx，而是Jacob

限制并发量

http{

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

server {

limit\_req zone=one burst=5;

此设置:当前处理一个连接 其余5个放入漏斗,最多处理6个连接请求 其他失败

//备注说明：

//limit\_req\_zone语法格式如下：

//limit\_req\_zone key zone=name:size rate=rate;

//上面案例中是将客户端IP信息存储名称为one的共享内存，内存空间为10M

//1M可以存储8千个IP信息，10M可以存储8万个主机连接的状态，容量可以根据需要任意调整

//每秒中仅接受1个请求，多余的放入漏斗(burst)

//漏斗超过5个则报错

拒绝非法的请求

网站使用的是HTTP协议，该协议中定义了很多方法，可以让用户连接服务器，获得需要的资源。但实际应用中一般仅需要get和post。

server {

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

return 444;

}

修改服务器配置后，客户端使用不同请求方法测试：

[root@client ~]# curl -i -X GET http://192.168.4.5 //正常

[root@client ~]# curl -i -X HEAD http://192.168.4.5 //报错

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 1k;

large\_client\_header\_buffers 2 1k;

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

数据库安全

安装完MariaDB或MySQL后，默认root没有密码，并且提供了一个任何人都可以操作的test测试数据库。

有一个名称为mysql\_secure\_installation的脚本，该脚本可以帮助我们为root设置密码，并禁止root从远程其他主机登陆数据库，并删除测试性数据库test。

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

//执行初始化安全脚本

密码安全

手动修改MariaDB或MySQL数据库密码的方法：

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

//修改密码，旧密码为redhat，新密码为mysql

MariaDB [(none)]>set password for root@'localhost'=password('redhat');

//使用账户登录数据库，修改密码

update mysql.user set password=password('654321') where user='root'and host='localhost';

//更新mysql.user表中password字段

删除记录历史命令文件

[root@proxy ~]# rm -rf ~/.bash\_history //记录命令行修改的密码

//通过命令行修改的密码，bash会自动记录历史，历史记录中记录了明文密码

[root@proxy ~]# rm -rf ~.mysql\_history //记录mysql命令来获取密码

//通过mysql命令修改的密码，mysql也会有所有操作指令的记录，这里也记录了明文密码

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

Tomcat安全性

隐藏版本信息、修改tomcat主配置文件（隐藏版本信息）

curl -I 192.168.4.72:8080/test2.html

<!DOCTYPE html><html><head><title>Nginx/9.0.30 //访问一个不存在网页时出现的提示信息

//修改 vim org/apache/catalina/util/ServerInfo.properties 可以实现

//根据自己的需要，修改版本信息的内容,此处控制访问一个不存在网页时出现的提示信息

网页存在 或 访问ip

curl -I 192.168.4.72:8080/test.html

curl -I 192.168.4.72:8080 //显示信息

[root@host72 util]# curl -I 192.168.4.72:8080

HTTP/1.1 200 OK

Content-Type: text/html;charset=UTF-8

Transfer-Encoding: chunked

Date: Tue, 24 Sep 2019 09:01:11 GMT

Server: IIS

//可以修改 vim +69 /usr/local/tomcat/conf/server.xml

redirectPort="8443" server="IIS" /> //添加server="IIS"

来控制显示信息

降级启动

默认tomcat使用系统高级管理员账户root启动服务，启动服务尽量使用普通用户。

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

diff

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

防火墙

一道保护性的安全屏障;保护,隔离

RHEL7默认使用firewalld作为防火墙,但firewalld底层还是调用 包过滤防火墙iptables

主机/网络型防火墙

根基保护对象(本机,其他主机)区分,保护本机为主机型, 其他主机为网络型

iptables框架

1）iptables的4个表（区分大小写）：

iptables默认有4个表:

filter表（数据过滤表）(默认表)

nat表（地址转换表）

raw表（状态跟踪表）

mangle表（包标记表）

2）iptables的5个链（区分大小写）：

INPUT链（入站规则）

OUTPUT链（出站规则）

FORWARD链（转发规则）

PREROUTING链（路由前规则）

POSTROUTING链（路由后规则）

包过滤匹配流程

规则链内的匹配顺序

顺序比对(1,2,3..依次往下),匹配即停止(LOG除外)

若无任何匹配,则按该链的默认策略处理

注意事项/整体规律

//可以不指定表，默认为filter表

//可以不指定链，默认为对应表的所有链

//如果没有找到匹配条件，则执行防火墙默认规则

//选项/链名/目标操作用大写字母，其余都小写

iptables命令的基本使用方法 (-p 必须写在 --dport前s)

iptables [-t 表名] 选项 [链名] [条件] [-j 目标操作]

常用的管理选项

添加规则: -A 在链末尾追加一条规则

-I 在链的开头(或指定序号,则在序号前)插入一条规则

查看规则:-n 以数字形式显示地址,端口等信息

-L 列出所有的规则条目

--line-numbers 查看规则时,显示规则的序号

删除规则:-D 删除链内指定序号(或内容)的一条规则

-F 清空所有的规则

默认策略:-P 为指定的链设置默认规则

基本的匹配条件

通用匹配

可直接使用,不依赖于其他条件或扩展

包括网络协议,IP地址,网络接口等条件

隐含匹配

要求以特定的协议匹配作为前提(-p 写在 --dport前)

包括端口,TCP标记,ICMP类型等条件

通用匹配: -p 协议匹配

地址匹配: -s 源地址

-d 目标地址

接口匹配:-i 收数据的网卡地址

-o 发数据的网卡

基本的目标操作

ACCEPT : 允许通过/放行

DROP : 直接丢弃,不给出任何回应

REJECT : 拒绝通过,必要时会给出提示 (即回包)

LOG : 记录日志(不回包),然后传给下一条规则 !!匹配即停止规律的唯一例外,记录完之后继续往下匹配

iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT //只允许本机ping其他主机,其他主机无法ping本机

--icmp-type echo-reply //类型为 只允许回复