## 系统状态查询

netstat -n | awk '/^tcp/ {++S[$NF]} END {for(a in S) print a, S[a]}'

显示服务器上所有的每个IP多少个连接数

netstat -ntu | awk '{print $5}' | cut -d: -f1 | sort | uniq -c | sort -n

查询系统端口及服务状态  
netstat -tnl  
查看端口对应server  
lsof -i :22 //查看22端口对应sshd服务  
查询服务运行级别  
chkconfig --list  
GUI设置服务命令  
ntsysv  
调整服务运行级别  
以kudzu服务为例//检测硬件更换服务  
chkconfig --level 3 kzdzu on  
chkconfig --level 2345 kzdzu off

搭建配置Nagios对服务器服务进行全面的监控

刚发现 /var/log/btmp 这个文件有好几个 G 的大小，搜了下关于 btmp 文件的信息：

/var/log/btmp这个文件记录错误的登录尝试，执行lastb命令就可以查看到最后一个不成功的登录尝试.

如果文件太大可删除了再重新生成一个新的空文件。

大多数使用iptables软件防火墙  
Iptables –A INPUT –p imcp –j   
Drop(丢弃)  
ACCPET(接受)  
REJECT(弹回)  
LOG(日志)  
IPTABLES –t  
表明  
分三类nat filteter(默认) mangle(服务质量等)  
Iptables   
-A 增加一个规则  
-D 删除规则  
-R 替换(指定行上替换)  
-I 插入  
-L 显示所有规则  
-F 删除所有规则  
-P 默认策略  
--line-numbers显示行号  
-p 指定使用的协议  
！号排除  
--src 源IP地址  
--dst 目的地址  
--in-interface 选择网卡  
--fragment 数据包分段  
--sport 源端口  
--dport 目的端口  
--state 状态(RELATED,ESTABLISHED)  
demo::防ping  
Iptables –A INPUT –p imcp –j DROP  
   
   
demo::限制某端口  
iptables –A INPUT –p tcp –d 192.168.0.1 –dport 21 –j DROP

chattr +i 防删除  
chattr -i 取消防删  
可安装第三方app防止root取消反删除属性  
lsattr 查询属性

## ICMP限制

我们可以阻断ICMP echo请求。但是，如果有自己的自治系统，就应该允许从因特网上ping你。不能ping通会使ISP或技术支持团队（如果有的话）丧失某些故障排解能力。

## 链接数控制

## syn攻击

查看linux默认的syn配置：

[root@web ~]# sysctl -a | grep \_syn

net.ipv4.tcp\_max\_syn\_backlog = 1024

net.ipv4.tcp\_syncookies = 1

net.ipv4.tcp\_synack\_retries = 5

net.ipv4.tcp\_syn\_retries = 5

tcp\_max\_syn\_backlog是SYN队列的长度，加大SYN队列长度可以容纳更多等待连接的网络连接数。tcp\_syncookies是一个开关，是否打开SYN Cookie 功能，该功能可以防止部分SYN攻击。tcp\_synack\_retries和tcp\_syn\_retries定义SYN 的重试连接次数，将默认的参数减小来控制SYN连接次数的尽量少。

#sysctl -w net.ipv4.tcp\_max\_syn\_backlog=2048

#sysctl -w net.ipv4.tcp\_syncookies=1

#sysctl -w net.ipv4.tcp\_synack\_retries=3

#sysctl -w net.ipv4.tcp\_syn\_retries=3

也有的人喜欢用访问控制列表来防止SYN的攻击，在一定程度上减缓了syn的攻击：

Syn 洪水攻击

#iptables -A INPUT -p tcp --syn -m limit --limit 1/s -j ACCEPT

--limit 1/s 限制syn并发数每秒1次

防端口扫描

# iptables -A FORWARD -p tcp --tcp-flags SYN,ACK,FIN,RST RST -m limit --limit 1/s -j ACCEPT

死亡之ping

# iptables -A FORWARD -p icmp --icmp-type echo-request -m limit --limit 1/s -j ACCEPT

#>iptables-save >/etc/sysconfig/iptables

再次进行查看syn连接：

[root@web ~]# netstat -an | grep SYN | awk '{print $5}' | awk -F: '{print $1}' | sort | uniq -c | sort -nr | more

     20 10.92.10.220

      1 125.43.36.199

明显SYN连接数已经下来了。

UDP洪水，就是各种各样的假冒攻击利用简单的TCP/IP服务，如Chargen和Echo来传送毫无用处的占满带宽的数据。通过伪造与某一主机的Chargen服务之间的一次的UDP连接，回复地址指向开着Echo服务的一台主机，这样就生成在两台主机之间的足够多的无用[数据流](http://zhidao.baidu.com/search?word=%E6%95%B0%E6%8D%AE%E6%B5%81&fr=qb_search_exp&ie=utf8" \t "_blank)，如果足够多的[数据流](http://zhidao.baidu.com/search?word=%E6%95%B0%E6%8D%AE%E6%B5%81&fr=qb_search_exp&ie=utf8" \t "_blank)就会导致带宽的服务攻击。

## HTTP单IP连接限制

<http://bbs.chinaunix.net/thread-3665515-1-1.html>

防简单攻击iptables策略

# 如果同时在80端口的连接数大于10，就Drop掉这个ip

netstat -an | grep :80 | awk -F: '{ print $8 }' | sort | uniq -c | awk -F\   '$1>10 && $2!="" { print $2 }' >> /etc/fw.list

less /etc/fw.list | sort | uniq -c | awk -F\   '$2!="" { print $2 }' > /etc/fw.list2

less /etc/fw.list2 > /etc/fw.list

while read line

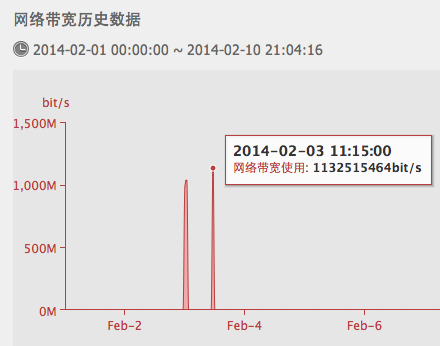
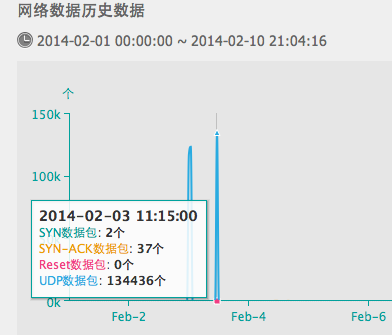
       do

       t=`echo "$line"`

       $IPTABLES -A INPUT -p tcp -s $t -j DROP

done < /etc/fw.list2

## DDoS攻击

分布式拒绝服务攻击（DDoS）是目前黑客经常采用而难以防范的攻击手段。DoS的攻击方式有很多种，最基本的DoS攻击就是利用合理的服务请求来占用过多的服务资源，从而使合法用户无法得到服务的响应。

通常，被攻击的服务器有以下症状：

1、被攻击主机上有大量等待的TCP连接；

2、网络中充斥着大量的无用的数据包，源地址为假；

3、制造高流量无用数据，造成网络拥塞，使受害主机无法正常和外界通讯；

4、利用受害主机提供的服务或传输协议上的缺陷，反复高速的发出特定的服务请求，使受害主机无法及时处理所有正常请求；

5、严重时会造成系统死机……

## 致命的大流量攻击

### 攻击特点

这种攻击通常以tcp syn，icmp和UDP（尤其是UDP包，单UDP的数据包可以很大）方式为主。客服系统遭遇到的最大的一次为16G的攻击流量，整个机房都被影响到。攻击者通常控制大量肉鸡或者直接勾结IDC里的服务器和带宽资源，对目标进行流量打击。此时流量会快速占满服务器的网络带宽，导致无法响应任何用户请求。

## 硬防

mysql, memcache服务设置白名单访问机制

# iptables -F

# iptables -P INPUT DROP

# iptables -A INPUT -p tcp -s 192.168.0.2 --dport 11211 -j ACCEPT

# iptables -A INPUT -p udp -s 192.168.0.2 --dport 11211 -j ACCEPT

## Linux系统开发安全管理

root密码的控制(php,nginx如何热启动)

开发人员gzwq账号

开发环境建设

mysql不开放远程访问，只开放内网IP

监控服务器的HTTP的返回值。

**CC攻击**

{http}

limit\_conn\_zone $binary\_remote\_addr zone=conn\_zone:64m;

limit\_req\_zone $binary\_remote\_addr zone=req\_zone:64m rate=100r/m;

{server}

#limit

limit\_conn conn\_zone 32;

limit\_req zone=req\_zone burst=5 nodelay;

limit\_req\_zone: 请求数控制，平均处理的请求频率不能超过每分钟100次

同时允许超过频率限制的请求数不多于5个。

如果不希望超过的请求被延迟，可以用nodelay参数,

设置对应的共享内存限制域和允许被处理的最大请求数阈值。 如果请求的频率超过了限制域配置的值，请求处理会被延迟，所以所有的请求都是以定义的频率被处理的。 超过频率限制的请求会被延迟，直到被延迟的请求数超过了定义的阈值，这时，这个请求会被终止，并返回503 (Service Temporarily Unavailable) 错误。这个阈值的默认值为0i

limit\_conn\_zone: 限制连接数，limit\_conn conn\_zone 32;限制并发的最大连接数位32.

参数req,conn都可以使用白名单机制，大致逻辑，对于白名单IP地址处理位空字符””，白名单之外的IP显示真实的IP地址。为空字符“”的数据就直接跳过nginx限制设定。

参考：

limit白名单机制

http://netsecurity.51cto.com/art/201306/400210.htm

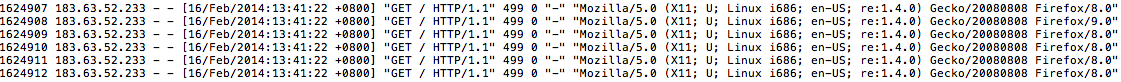
#允许单个IP的最大连接数为 50

iptables -I INPUT -p tcp --dport 80 -m connlimit  --connlimit-above 50 -j REJECT

控制单个IP在一定的时间（比如60秒）内允许新建立的连接数

iptables -A INPUT -p tcp --dport 80 -m recent --name BAD\_HTTP\_ACCESS --update --seconds 60 --hitcount 30 -j REJECT iptables -A INPUT -p tcp --dport 80 -m recent --name BAD\_HTTP\_ACCESS --set -j ACCEPT #单个IP在60秒内只允许最多新建30个连接

nginx UA限制设定:



if ( $http\_user\_agent ~\* "Mozilla/5.0\ \(X11;\ U;\ Linux\ i686;\ en-US;\ re:1.4.0\)\ Gecko/20080808\ Firefox/8.0" )

{

return 503;

}

以下脚本通过netstat命令列举所有连接，将连接数最高的一个IP如果连接数超过150，则通过 iptables阻止访问：

01.#!/bin/sh

02.status=`netstat -na|awk '$5 ~ /[0-9]+:[0-9]+/ {print $5}' |awk -F ":" -- '{print $1}' |sort -n|uniq -c |sort -n|tail -n 1`

03.NUM=`echo $status|awk '{print $1}'`

04.IP=`echo $status|awk '{print $2}'`

05.result=`echo "$NUM > 150" | bc`

06.if [ $result = 1 ]

07.then

08.echo IP:$IP is over $NUM, BAN IT!

09./sbin/iptables -I INPUT -s $IP -j DROP

10.fi

运行crontab -e，将上述脚本添加到crontab每分钟自动运行：

1.\* \* \* \* \* /root/xxxx.sh

## 登录接口改进：

http://www.99yx.com/member/mem\_login\_ajax?account=happy12&passwd=19218fcc8f5894fcd95531f269978f8a&remember=N&jsoncallback=jQuery172009848513104952872\_1392941970266&\_=1392941990996

加入ts, passwd = md5/sha1(account+ts+ md5\_passwd)

ts有效期不超过3分钟

## 系统操作日志记录

/etc/profile.d/history.sh

USER=`whoami`

USER\_IP=`who -u am i 2>/dev/null| awk '{print $NF}'|sed -e 's/[()]//g'`

if [ "$USER\_IP" = "" ]; then

USER\_IP=`hostname`

fi

if [ ! -d /var/log/history ]; then

mkdir /var/log/history

chmod 777 /var/log/history

fi

export HISTSIZE=4096

DT=`date +"%Y%m%d%H%M%S"`

export HISTFILE="/var/log/history/$DT.${USER}@${USER\_IP}\_history"

chmod 600 /var/log/history/\*history\* 2>/dev/null