[浅谈DDos攻击攻击与防御 7](#_Toc1580748536)

[小型商务网站如何应对DDoS攻击 10](#_Toc1986150405)

linux面试题:删除一个目录下的所有文件,但保留一个指定文件

find /xx -type f ! -name "file10"|xargs rm -f

find /xx -type f ! -name "file10" -exec rm -f {} /;

rsync -az --delete --exclude "file10" /null/ /xx/

快速截取时间段的日志做分析

awk '$1 >="16/May/2015:11:52:22" &;&; $1 <="16/May/2015:11:55:54"' access.log

awk '{ip[$3]++}END {for (s in ip) print s,ip[s]}' access.log

shell脚本之批量处理文件

Varnish 获取真实IP

tcpflow -cp host IP and port 80

由上图我们可以看出来是使用了X-Forwarded-For变量

修改varnish的配置文件

vim /etc/init.d/varnishncsa

修改为这样的%{X-Forwarded-For}i

DAEMON\_OPTS="-m TxHeader:X-Cache: HIT -F /"%{X-Forwarded-For}i %{%Y-%m-%dT%H:%M:%S+08:00}t %m %U %q - %s %b %D %{User-Agent}i/" -P $pidfile -a -w $logfile -D"

查询库文件属于哪个rpm包

yum install XXX --enablerepo=YYY

yum provides libname

增加swap的大小

在生产环境中要注意,先将服务器切下来,然后停掉服务,添加swap

Too many authentication failures for root (code2)解决办法

找到这个选项:MaxAuthTries=2

Linux被DDOS及CC攻击解决方案

netstat -anlp|grep 80|grep tcp|awk '{print $5}'|awk -F: '{print $1}'|sort|uniq -c|sort -nr|head -n20 | netstat -ant |awk '/:80/{split($5,ip,":");++A[ip[1]]}END{for(i in A) print A[i],i}' |sort -rn|head -n20

iptables -I INPUT -s 222.127.94.0/16 -j DROP

iptables -I INPUT -s 247.27.128.0/16 -j DROP

iptables -I INPUT -s 145.27.133.0/16 -j DROP

iptables -I INPUT -s 193.1.0.0/8 -j DROP 【慎用封禁整个段】

运行下面这行命令,来查看谁的访问量最高(需要服务器安装tcpdump)

tcpdump -i eth0 -tnn dst port 80 -c 1000 | awk -F"." '{print $1"."$2"."$3"."$4}' | sort | uniq -c | sort -nr |head -20

如果出现误封可以参考下面这行解封命令

iptables -D INPUT -s 222.142.2.0/16 -j DROP

关于换行和回车其实平时我们不太在意,所以关于两者的区别也不太清楚,在平时开发时可能会遇到一些文件处理的问题,放到不同的操作系统上出现各种坑。那么回车和换行到底有哪些区别呢?今天咱们就来总结一下。

1. 由来

在计算机还没有出现之前,有一种叫做电传打字机(Teletype Model 33)的机械打字机,每秒钟可以打10个字符。但是它有一个问题,就是打完一行换行的时候,要用去0.2秒,正好可以打两个字符。要是在这0.2秒里面,又有新的字符传过来,那么这个字符将丢失。

于是,研制人员想了个办法解决这个问题,就是在每行后面加两个表示结束的字符。一个叫做“回车”,告诉打字机把打印头定位在左边界,不卷动滚筒;另一个叫做“换行”,告诉打字机把滚筒卷一格,不改变水平位置。

这就是“换行”和“回车”的由来。

2. 使用

后来,计算机发明了,这两个概念也就被般到了计算机上。那时,存储器很贵,一些科学家认为在每行结尾加两个字符太浪费了,加一个就可以。于是,就出现了分歧。

回车 /r 本义是光标重新回到本行开头,r的英文return,控制字符可以写成CR,即Carriage Return

换行 /n 本义是光标往下一行(不一定到下一行行首),n的英文newline,控制字符可以写成LF,即Line Feed

符号ASCII码 意义

/n10换行NL

/r13回车CR

在不同的操作系统这几个字符表现不同,比如在WIN系统下,这两个字符就是表现的本义,在UNIX类系统,换行/n就表现为光标下一行并回到行首,在MAC上,/r就表现为回到本行开头并往下一行,至于ENTER键的定义是与操作系统有关的。通常用的Enter是两个加起来。

不同操作系统下的含义:

/n: UNIX 系统行末结束符

/n/r: window 系统行末结束符

/r: MAC OS 系统行末结束符

我们经常遇到的一个问题就是,Unix/Mac系统下的文件在Windows里打开的话,所有文字会变成一行;而Windows里的文件在Unix/Mac下打开的话,在每行的结尾可能会多出一个^M符号。

3. 软回车和硬回车

再扩展一下回车的一些知识。

硬回车就是普通我们按回车产生的,它在换行的同时也起着段落分隔的作用。

软回车是用 Shift + Enter 产生的,它换行,但是并不换段,即前后两段文字在 Word 中属于同一“段”。在应用格式时你会体会到这一点。

软回车能使前后两行的行间距大幅度缩小,因为它不是段落标记,要和法定的段落标记——硬回车区别出来。硬回车的html代码是..,段落的内容就夹在里面,而软回车的代码很精悍:  
。网页的文字如果复制到word中,则硬回车变为弯曲的箭头,软回车变为向下的箭头。

摘要：以下内容转载自http://blog.csdn.NET/yazhouren/article/details/7356746现在研究CI(持续集成)链,公司要使用jenkins,刚开始接触很多不懂,现在经过一个多月的查找资料,终于明白了一些现在网上的资料特别的混乱,有的特别的基础,学了还是什么都没搞明白;有的是很深,一点基础都不讲,怎么也看不明白,中间层很欠缺。也许使用这些技术的公司,对于员工都进行了很好的培训,基础的东西一听都明白了,但是对于自学使用这些工具的人来说,如果不能

以下内容转载自http://blog.csdn.NET/yazhouren/article/details/7356746

现在研究CI(持续集成)链,公司要使用jenkins,刚开始接触很多不懂,现在经过一个多月的查找资料,终于明白了一些

现在网上的资料特别的混乱,有的特别的基础,学了还是什么都没搞明白;有的是很深,一点基础都不讲,怎么也看不明白,中间层很欠缺。也许使用这些技术的公司,对于员工都进行了很好的培训,基础的东西一听都明白了,但是对于自学使用这些工具的人来说,如果不能提纲挈领,抓住要领,学起来真费劲。下面,我就给根据我的理解,来说一说,希望对看到这篇文章的人有帮助,错误之处,敬请指正。

1. Make工具

这个工具是最最原始的工具了,在Linux下编过程序,看过比较大的c/c++代码的人应该都听说过这个工具(Windows下对应的工具为nmake)。它负责组织构建的过程,也就是负责指挥编译器如何编译,连接器如何连接,最后生成一个可用的文件。

2. Ant工具

但是有人觉得make工具的很难使用,他的语法很难理解,就发明了ant

3. Maven工具

Maven工具是对ant工具的进一步改进(这么说不太准确,但是可以这么理解)。

在make工具中,如果我们要编译某些源文件,我们肯定首先要安装编译器等工具。但是有时候需要不同版本的编译器,在Java的编译器需要不同的各种包的支持,如果把每个包都下载下来,在makefile中进行配置制定,当需要的包非常多时,很难管理。

[plain]view plaincopy

例如一个web项目,可能使用到很多技术,面向对象、泛型、or-mapping、依赖注入(spring-framework)、全文检索(lucene)、数据库、集群、工作流、webservice等等。

由于使用了多种技术,这些技术可能是JDK提供的,也可能是第三方开源组织提供的,或者不同的商业公司提供的。

于是出现了一个新的难题,就是包依赖复杂性。以前,你很难想象你的代码依赖数十个不同开源组织、商业公司提供的库。例如,我们经常使用的log4j、junit、easymock、ibatis、springframework,每个组件都有悠久的历史,存在不同的版本,他们之间版本还有依赖关系。

项目依赖的复杂性,经常的,一个较大部门有10-30个项目是常事,项目之间有不同版本的依赖关系,部门与部门之间的项目也存在复杂的版本依赖关系。

Eclipse本身提供Project的依赖,但是简单的依赖显然解决不了问题。例如ProjectB依赖ProjectA,ProjectA依赖第三方的jar包abc-1.0.jar,那么需要在两个项目的lib下都存放abc-1.0.jar,这产生冗余,当Project数量多起来,这个冗余就产生了管理问题,如果需要将abc-1.0.jar升级为abc-1.1.jar,则需要在两个Project中同时修改,如果Project数量达到10个以上,而且是不同项目组维护的项目,这个就是非常麻烦的事情。而且ProjectA修改依赖,为啥需要ProjectB也作相应的修改呢?

需要解决此问题,就需要在ProjectA的包中描述其依赖库的信息,例如在META-INFO记录所以来的abc-1.0.jar等。Eclipse的plug-in拥有类似的方案,但是这样一来,就使得开发ProjectB的项目组,需要把ProjectA的代码从源代码库中checkout出来。在依赖链末端的项目组是很惨的。

由于Project数量众多,关系复杂,dailybuild的ant脚本编写成了很麻烦的事情,使用Project依赖的方式,更加使得编写dailybuildantscript是非常痛苦的事情。

参考:http://www.cnblogs.com/jobs/archive/2007/09/24/903731.html

这样,于是人们发明了Maven工具。Maven使用配置文件pom.xml对环境进行配置,例如设定编译器的版本,设定所需支持包的URL,这样maven就可以自动去下载所需的包。这样如果需要对构建环境进行改变时,直接改变pom文件就可以了,maven会自动网络上下载配置的包。

Maven像make一样,是个构建(build)工具,它如何调用各种不同的编译器连接器等呢? 这就是Maven plugin (maven 插件)。maven插件是为了使maven能够实用各种工具。不同的工具有对应的插件。

Maven比make的更加大的地方是,可以利用一些别的工具,实现对编译结果的统计,对源代码的检查,对于代码的测试等。例如checkstyle,cobertura等,都有对应的Maven插件。

4. Jenkins工具

Maven已经很强大了,那Jenkins是个什么东西?

其实Maven还是不够强大。Maven可以控制编译,控制连接,可以生成各种报告,可以进行代码测试。

可是如何控制这个流程呢? 编译还是先连接?先进行代码测试,还是先生成报告??

可以使用脚本来对maven进行控制,实现这些流程的控制。但是,很不方便。即使能够走完整个流程,但是查看结果(例如代码静态分析的结果)时,也不方便。设想我们查看了代码覆盖率的结果,又想查看代码style的结果,又要对这些结果进行评价,还要发送给开发人员........太罗嗦了,有没有自动化图形界面实现这些过程的工具????

有,这个工具就是Jenkins。

Jenkins能够对流程进行控制,对能够对各个阶段生成的各种结果进行综合,以图表的形式呈现出来。

但是,不同的工具生成不同格式的结果,Jenkins如何让这些结果呈现出来??

这就是jenkins插件,Jenkins插件使maven中用到的各种工具生成的结果能够在Jenkins中呈现出来。

5. 一些特殊的东东

有些工具,不能被maven使用,却可以通过Jenkins调用脚本来生成结果,并有对应的Jenkins插件,将结果Jenkins中显示出来,例如cppcheck,CCCC等工具

6. Jenkins项目中对于项目配置中工具的配置与workspace中pom文件的该工具的配置的关系

例如,使用该cobertura工具,需要在jenkins中安装对应的插件,在项目设置中,制定cobertura-result所在的目录

但是使用cobertura工具的哪个版本?obertura如何找到需要检查的源文件?cobertura工具会把检查的结果放到哪个目录?

这些设置需要使用pom文件来配置的。

7. pom.xml与makefile

maven中使用的pom与make中使用的makefile作用差不多

但是在maven中,如果想编译某个文件夹下所有文件(包含子文件夹的文件)该怎么办呢?

我现在的方法是使用aggregation 和 inheritance ,详见pom reference(http://maven.apache.org/guides/introduction/introduction-to-the-pom.html)

Inheritance:

If you have several Maven projects, and they all have similar configurations, you can refactor your projects by pulling out those similar configurations and making a parent project. Thus, all you have to do is to let your Maven projects inherit that parent project, and those configurations would then be applied to all of them.

Aggregation:

And if you have a group of projects that are built or processed together, you can create a parent project and have that parent project declare those projects as its modules. By doing so, you'd only have to build the parent and the rest will follow.

8. 总结:怎么样,看了这篇文章是不是思路逐渐的清晰起来了!!!

我讲的可能不是很严谨,但大体是这么个意思。

再说,明白这些也没有多大的意义,我们的终极目的是使用这些工具,来对代码进行分析,提高代码的质量。

如果您通过读这篇文章,明白了这些工具大体的框架,那么剩下的就是仔细研究各个分析工具的具体使用,希望能够通过使用分析工具提高大家的代码质量。

忙了一个月,看了好多的英文html,才学到这点知识,希望能给后来人一点点帮助!

LINUX目录

/bin存放执行命令

/boot 存放操作系统启动时所要用到的程序,如grub

/devlinux系统使用的所有外部设备

/etc存放了系统管理用到的各种配置文件和子目录,如网络配置文件、文件系统、用户信息等。

/etc/rc.d LINUX系统启动和关闭时要用到的脚本文件

/etc/rc.d/initLINUX服务默认启动脚本

/home用户工作目录

/lib 系统动态链接共享库

/losf+found 系统产生异常时,会将一些遗失的片段放在此目录下。

/media光驱和软驱的挂载点

/proc 系统核心与执行程序所需的一些信息,这些信息是在内存中由系统产生的,故不占用磁盘空间。

/root 超级用户主目录

/sbin 存放系统管理员的常用的系统管理程序

/tmp存放不同程序执行时产生的临时文件

/usr相当于windows下的Program Files目录

/usr/bin系统用户使用的应用程序

/usr/sbin超级用户使用的比较高级的管理程序和系统守护程序

/usr/src内核源代码默认的放置目录

/srv 存放一些服务启动之后需要提取的数据

/sys 将内核的一些信息映射,可供应用程序

/var 服务日志信息存放

# 浅谈DDos攻击攻击与防御

在前几天，我们运营的某网站遭受了一次ddos攻击，我们的网站是一个公益性质的网站，为各个厂商和白帽子之间搭建一个平台以传递安全问题等信息，我们并不清楚因为什么原因会遭遇这种无耻的攻击。

作者：佚名来源：80sec|2012-11-30 14:54

[收藏](javascript:favorBox('open');" \o "一键收藏，随时查看，分享好友！" \t "/root/文档\\x/_self)

[分享](javascript:;)

[技术沙龙 | 邀您于8月25日与国美/AWS/转转三位专家共同探讨小程序电商实战](http://mdsa.51cto.com/act/Tech/Tech22" \t "/root/文档\\x/_blank)

一 背景

在前几天，我们运营的某网站遭受了一次ddos攻击，我们的网站是一个公益性质的网站，为各个厂商和白帽子之间搭建一个平台以传递安全问题等信息，我们并不清楚因为什么原因会遭遇这种无耻的攻击。因为我们本身并不从事这种类型的攻击，这种攻击技术一般也是比较粗糙的，所以讨论得比较少，但是既然发生了这样的攻击我们觉得分享攻击发生后我们在这个过程中学到得东西，以及针对这种攻击我们的想法才能让这次攻击产生真正的价值，而并不是这样的攻击仅仅浪费大家的时间而已。

另外，我们发现大型的企业都有遭受攻击的案例，但是大家遭受攻击之后的应对措施及学到的经验却分享都比较少，这导致各家都是自行的摸索经验，依然停留在一家企业对抗整个互联网的攻击的局面，而对于攻击者却是此次攻击针对你，下次攻击却是针对他了，而且攻击之后无论是技术还是资源都没有任何的损耗，这也是导致这种攻击频繁并且肆无忌惮的原因。

我们来尝试做一些改变：）

二 应急响应

在攻击发生后，第一个现象是我们的网站上不去了，但是依然可以访问到管理界面，我们登陆上去简单执行了命令：

netstat -antp

我们看到有大量的链接存在着，并且都是ESTABLISHED状态，正常状态下我们的网站访问量没有这么高，如果有这么高我们相信中国的信息安全就有希望了，对于这样的情况其实处理就比较简单，这是一次四层的攻击，也就是所有ip都是真实的，由于目前为止只是消耗了webserver的网络连接资源，所以我们只需要简单的将这些ip在网络层封禁就可以，很简单，用下面的命令即可：

for i in `netstat -an | grep -i ‘:80 ‘|grep ‘EST’ | awk ‘{print $5}’ | cut -d : -f 1 | sort | uniq -c | awk ‘{if($1 > 50) {print $2}}’`

echo $i

echo $i >> /tmp/banip

/sbin/iptables -A INPUT -p tcp -j DROP -s $i

done

然后作为计划任务一分钟执行一次即可，很快，iptables的封禁列表就充斥了大量的封禁ip，我们简单的统计了下连接数最大的一些ip发现都来自韩国。为了保证系统的性能，我们调大了系统的可接受的连接数以及对Nginx进行了每个连接能够进行的请求速率，系统于是恢复了正常的运行。

正常状态一直持续到第二天，但是到中午之后我们发现访问又出现了问题，网络很慢，使用ping发现大概出现了70%左右的丢包，在艰难的登陆到系统上之后，发现系统已经很少有TCP的正常连接，为了查明原因，我们对系统进行了抓包：

tcpdump -w tmp.pcap port not 22

tcpdump -r tmp.pcap -nnA

我们发现攻击已经从应用层的攻击调整到了网络层的攻击，大量的目标端口是80的udp和icmp包以极快的速度充满了网络，一个包大小大概在1k左右，这次占据的资源纯粹是带宽资源了，即使在系统上做限制也解决不了这个问题，不过也没有关系，对于网络层的问题我们可以在网络层上做限制，我们只需要在网络上把到达我们ip的非TCP的所有包如UDP和ICMP等协议都禁止掉即可，但是我们没有自己的服务器也缺乏对网络设备的控制权，目前是由工信部CERT提供支持的，由于临时无法协调进行相应的操作，后果如大家看到，我们的服务很慢，基本上停止了服务，在一段时间之后攻击者停止了攻击，服务才进行了恢复，很憋屈是么？但是同时我们得到了很多热心朋友的帮助，得到了更好的网络和服务器资源，在网络资源方面的能力得到了很大的提升，缓解了这方面的问题，这里对他们表示感谢。

三 常见ddos攻击及防御

继续秉承80sec的”Know it then hack it”，这里简单谈一下ddos攻击和防御方面的问题。ddos的全称是分布式拒绝服务攻击，既然是拒绝服务一定是因为某些原因而停止服务的，其中最重要的也是最常用的原因就是利用服务端方面资源的有限性，这种服务端的资源范围很广，可以简单的梳理一个请求正常完成的过程：

1 用户在客户端浏览器输入请求的地址

2 浏览器解析该请求，包括分析其中的dns以明确需要到达的远程服务器地址

3 明确地址后浏览器和服务器的服务尝试建立连接，尝试建立连接的数据包通过本地网络，中间路由最终艰苦到达目标网络再到达目标服务器

4 网络连接建立完成之后浏览器根据请求建立不同的数据包并且将数据包发送到服务器某个端口

5 端口映射到进程，进程接受到数据包之后进行内部的解析

6 请求服务器内部的各种不同的资源，包括后端的API以及一些数据库或者文件等

7 在逻辑处理完成之后数据包按照之前建立的通道返回到用户浏览器，浏览器完成解析，请求完成。

上面各个点都可以被用来进行ddos攻击，包括：

1 某些著名的客户端劫持病毒，还记得访问百度跳搜狗的事情么？：）

2 某个大型互联网公司发生的dns劫持事件，或者直接大量的dns请求直接攻击dns服务器，这里可以使用一些专业的第三方dns服务来缓解这个问题，如Dnspod

3 利用建立网络连接需要的网络资源攻击服务器带宽使得正常数据包无法到达如udp的洪水攻击，消耗前端设备的cpu资源以使得数据包不能有效转发如icmp和一些碎片包的洪水攻击，消耗服务器方建立正常连接需要的资源如syn flood或者就是占用大量的连接使得正常的连接无法发起，譬如这次的TCP flood

4 利用webserver的一些特点进行攻击，相比nginx来说，apache处理一个请求的过程就比较笨重。

5 利用应用程序内部的一些特性攻击程序内部的资源如mysql，后端消耗资源大的接口等等，这也就是传统意义上的CC攻击。

这里涉及到攻防的概念，但是实际上如果了解对方的攻击点和攻击手法，防御会变成简单的一个拼资源的过程，不要用你最弱的地方去抗人家最强的地方，应该从最合适的地方入手把问题解决掉，譬如在路由器等设备上解决应用层攻击就不是一个好的办法，同理，在应用层尝试解决网络层的问题也是不可能的，简单来说，目标是只让正常的数据和请求进入到我们的服务，一个完善的防御体系应该考虑如下几个层面：

1 作为用户请求的入口，必须有良好的dns防御

2 与你的价值相匹配的带宽资源，并且在核心节点上布置好应用层的防御策略，只允许你的正常应用的网络数据包能够进入，譬如封杀除了80以外的所有数据包

3 有支持你的服务价值的机器集群来抵抗应用层的压力，有必要的话需要将一个http请求继续分解，将连接建立的过程压力分解到其他的集群里，这里似乎已经有一般的硬件防火墙能做这个事情，甚至将正常的http请求解析过程都进行分解，保证到达后端的是正常的请求，剔除掉畸形的请求，将正常的请求的请求频度等行为进行记录和监控，一旦发生异常就在这里进行应用层的封杀

每个公司都有自己对自己价值的评估从而决定安全投入上的大小，每一次攻击也会涉及到利益的存在，正如防御因为种种原因譬如投入上的不足和实施过程中的不完美，有着天生的弱点一样，攻击也是有着天生的弱点的，因为每一次攻击涉及到不同的环节，每个环节都可能由不同水平的人完成，他所拥有的资源，他使用的工具和技术都不会是完美的，所以才有可能进行防御，另外，我相信进行DDOS攻击的人是一个固定的行业，会有一些固定的人群，对于其中使用的技术，工具，资源和利益链都是比较固定的，与之相对的是各个企业却缺乏相应的沟通，以个人企业对抗一个产业自然是比较困难，而如果每一个企业都能将自己遭受攻击时的经验分享出来，包括僵尸网络的大小及IP分布，攻击工具的特征，甚至有能力的可以去分析背后的利益点及操作者，那么每一次攻击都能让大家的整体防御能力上升，让攻击者的攻击能力有损失，我们很愿意来做这个事情。

四 根源及反击

我困惑的是一点，攻击我们并不能得到实际的好处为什么还是有人来攻击，而且听说其他公司都有被攻击的情况，我觉得有一点原因就是攻击我们的确得不到什么好处，但是实际上攻击者也并不损失什么，无论是资源上还是法律风险上，他不会因为一次攻击而损失太多，而相比之下，服务提供者损失的东西却太多了，这从经济学角度来讲就是不平衡的，我们处于弱势。

一般而言，的确对于作恶者是没有什么惩罚措施，但是这次，我们觉得我们是可以做一些事情的，我们尝试挖掘背后的攻击者，甚至清除这个僵尸网络。

首先这次攻击起源于应用层的攻击，所以所有的ip都是真实的，经过与CERT沟通，也发现这些ip都是韩国的，并且控制端不在国内，因为期间没有与国内有过通讯，即使在后面换成了udp+icmp的flood，但是依然是那些韩国的ip，这很有意思，正常情况下udp+icmp的数据包是可以伪造的，但是这里居然没有伪造，这在后面大概被我们证实了原因。

这些ip是真实存在的ip，而且这些ip肯定在攻击完我们之后一定依然跟攻击者保持着联系，而一般的联系方式因为需要控制的方便都是dns域名，既然如此，如果我们能挖掘到这个dns域名我们就可能间接的挖掘出真正幕后黑手在哪里。首先，我们迅速的找出了这次攻击ip中开放了80端口的机器，因为我们对80端口上的安全问题比较自信，应该很快可以获知这些ip背后的细节（80sec名称由来），我们发现大部分是一些路由器和一些web的vpn设备，我们猜测这次攻击的主要是韩国的个人用户，而个人用户的机器操作系统一般是windows所以在较高版本上发送数据包方面可能有着比较大的限制，这也解释了为什么即使是udp+icmp的攻击我们看到的大都是真实ip。发现这些路由设备之后我们尝试深入得更多，很快用一些弱口令譬如admin/admin登陆进去，果然全世界的网民都一样，admin/admin是天生的入口。

登陆进去一些路由之后我们发现这些路由器里面存在一个功能是设置自己的dns，这意味着这下面的所有dns请求都可以被定向到我们自己设置的dns服务器，这对于我们去了解内部网络的细节会很有用，于是我们建立了一个自己的dns服务器，并且开启了dns请求的日志功能以记录所有请求的细节。我们大约控制了20台路由器的dns指向，并且都成功重定向到我们自己的服务器。

剩下的就是简单的数据分析，在这之前我们可以对僵尸网络的控制域名做如下的猜测：

1 这个dns应该为了灵活的控制域名的缓存时间TTL一般不会特别长

2 这个dns应该是定期的被请求，所以会在dns请求里有较大的出现比例

3 这个dns应该是为了控制而存在的，所以域名不应该在搜索引擎以及其他地方获得较高的访问指数，这与2中的规则配合起来会比较好确定，是一个天生的矛盾。

4 这个dns应该在各个路由下面都会被请求

这些通过简单的统计就很容易得出答案，我们发现了一些3322的通用恶意软件域名但是发现它并不是我们需要的，因为只有少数机器去访问到，经过一些时间之后最后我们发现一个域名访问量与naver（韩国的一个门户）的访问量持平，workgroup001.snow\*\*\*\*.net，看起来似乎对自己的僵尸网络管理很好嘛，大概有18台机器访问过这个域名，这个域名的主机托管在新加坡，生存时间TTL在1800也就是半小时，这个域名在所有的搜索引擎中都不存在记录，是一个韩国人在godady一年前才注册的，同时我们访问这个域名指向主机的3389，简单的通过5下shift就判断出它上面存在着一个典型的windows后门，似乎我们找到它了，不是么？经过后续的观察，一段时间后这个域名指向到了127.0.0.1，我们确信了我们的答案,workgroup001.snow\*\*\*\*.net，看起来似乎对自己的僵尸网络管理很好嘛：）

这是一次典型的ddos攻击，攻击之后我们获得了参与攻击的主机列表和控制端的域名及ip，相信中国和韩国的cert对于清理这次的攻击源很有兴趣，我们是有一些损失，但是攻击者也有损失了（大概包括一个僵尸网络及一个控制端域名，甚至可能包括一次内部的法律调查），我们不再是不平等的了，不是么？

五 总结

正如一个朋友所讲的，所有的防御是不完美的正如攻击是不完美的一样，好的防御者在提升自己的防御能力趋于完美的同时也要善于寻找攻击者的不完美，寻找一次攻击中的漏洞，不要对攻击心生恐惧，对于Ddos攻击而言，发起一次攻击一样是存在漏洞的，如果我们都能够擅长利用其中的漏洞并且抓住后面的攻击者那么相信以后的ddos攻击案例将会减少很多，在针对目标发起攻击之前攻击者也会做更多的权衡，损失，利益和法律。

# 小型商务网站如何应对DDoS攻击

由于商业竞争、打击报复和网络敲诈等多种因素，导致很多IDC托管机房、商业站点、游戏服务器、聊天网络等网络服务商长期以来一直被DDoS攻击所困扰，随之而来的是客户投诉、法律纠纷、商业损失等一系列问题。

作者：佚名来源：mgddos|2012-11-27 16:35

[收藏](javascript:favorBox('open');" \o "一键收藏，随时查看，分享好友！" \t "/root/文档\\x/_self)

[分享](javascript:;)

[技术沙龙 | 邀您于8月25日与国美/AWS/转转三位专家共同探讨小程序电商实战](http://mdsa.51cto.com/act/Tech/Tech22" \t "/root/文档\\x/_blank)

由于商业竞争、打击报复和网络敲诈等多种因素，导致很多IDC托管机房、商业站点、游戏服务器、聊天网络等网络服务商长期以来一直被DDoS攻击所困扰，随之而来的是客户投诉、法律纠纷、商业损失等一系列问题，因此，解决DDoS攻击问题成为商务网站必须考虑的头等大事。DDoS攻击很狡猾，也很难阻止，但是面对攻击，只要快速地响应和采取正确的措施，就可以及时发现攻击数据流并将其挡掉，减轻这种攻击对网络的影响。

DDoS(Distributed Denial of Service，分布式拒绝服务攻击)，俗称洪水攻击。是在传统的DoS攻击基础之上产生的新的破坏力更强的攻击方式。分布式拒绝服务攻击是指借助于客户/服务器技术，将多个甚至几十万个计算机联合起来作为攻击平台，对一个或多个目标发动DoS攻击，从而成倍地提高了攻击的威力。DDoS带来的破坏是巨大的，你无法阻止黑客对你的网站发动DDoS攻击，除非主动断开Internet连接。如果我们无法防止这种攻击，那么，怎样做才能最大限度地保护我们的企业网络呢?

1、了解DDoS攻击当前主要有三种流行的DDoS攻击：

1.1SYN/ACKFlood攻击

这种攻击方法是经典最有效的DDoS方法，可攻击各种系统的网络服务，主要是通过向受害主机发送大量伪造源IP和源端口的SYN或ACK包，导致主机的缓存资源被耗尽或忙于发送回应包而造成拒绝服务，由于源IP和源端口都是伪造的，故追踪起来比较困难。其缺点是实施起来有一定难度，需要高带宽的僵尸主机支持。

1.2TCP全连接攻击

TCP全连接攻击就是通过许多僵尸主机不断地与受害服务器建立大量的TCP连接，直到服务器的内存等资源被耗尽而被拖跨，从而造成拒绝服务，这种攻击的特点是可绕过一般防火墙的防护而达到攻击目的，缺点是需要找很多僵尸主机，并且由于僵尸主机的IP是暴露的，因此容易被追踪。

1.3刷Script脚本攻击

这种攻击主要是针对存在ASP、JSP、PHP、CGI等脚本程序，并调用MSSQLServer、MySQLServer、Oracle等数据库的网站系统而设计的，特征是和服务器建立正常的TCP连接，并不断地向脚本程序提交查询、列表等大量耗费数据库资源的调用，典型的以小博大的攻击方法。常见的现象就是网站慢如蜗牛、ASP程序失效、PHP连接数据库失败、数据库主程序占用CPU偏高。这种攻击的特点是可以完全绕过普通的防火墙防护，轻松找一些Proxy代理就可实施攻击，缺点是对付只有静态页面的网站效果会大打折扣，并且有些Proxy会暴露攻击者的IP地址。

2、发现DDoS攻击

根据以下异常现象在网络入侵监测系统建立相应规则，能够较准确地监测出DDoS攻击。

(1)根据分析，攻击者在进行DDoS攻击前总要解析目标的主机名，BIND域名服务器能够记录这些请求。由于每台攻击服务器在进行一个攻击前会发出PTR反向查询请求，也就是说在DDoS攻击前域名服务器会接收到大量的反向解析目标IP主机名的PTR查询请求。

(2)当DDoS攻击一个站点时，会出现明显超出该网络正常工作时的极限通讯流量的现象。现在的技术能够分别对不同的源地址计算出对应的极限值。当明显超出此极限值时，就表明存在DDoS攻击的通讯。因此，可以在主干路由器端建立ACL访问控制规则以监测和过滤这些通讯。

(3)特大型的ICP和UDP数据包。正常的UDP会话一般都使用小的UDP包，通常有效数据内容不超过10字节。正常的ICMP消息也不会超过64到128字节。那些尺寸明显大得多的数据包很有可能就是控制信息通讯用的，主要含有加密后的目标地址和一些命令选项。一旦捕获到(没有经过伪造的)控制信息通讯，DDoS服务器的位置就暴露出来了，因为控制信息通讯数据包的目标地址是没有伪造的。

(4)不属于正常连接通讯的TCP和UDP数据包。最隐蔽的DDoS工具随机使用多种通讯协议(包括基于连接的协议)通过基于无连接通道发送数据。优秀的防火墙和路由规则能够发现这些数据包。另外，那些连接到高于1024而且不属于常用网络服务的目标端口的数据包也是非常值得怀疑的。

(5)数据段内容只包含文字和数字字符(例如，没有空格、标点和控制字符)的数据包。这往往是数据经过BASE64编码后而只会含有BASE64字符集字符的特征。TFN2K发送的控制信息数据包就是这种类型的数据包。TFN2K(及其变种)的特征模式是在数据段中有一串A字符(AAA)，这是经过调整数据段大小和加密算法后的结果。如果没有使用BASE64编码，对于使用了加密算法数据包，这个连续的字符就是“”。

(6)数据段内容只包含二进制和high-bit字符的数据包。虽然此时可能在传输二进制文件，但如果这些数据包不属于正常有效的通讯时，可以怀疑正在传输的是没有被BASE64编码但经过加密的控制信息通讯数据包(如果实施这种规则，必须将20、21、80等端口上的传输排除在外)。

3、应对DDoS攻击

当遭受DDoS攻击的时候要如何设法存活并继续提供正常服务呢?

由前面的介绍可以知道，若黑客攻击规模远高于你的网络频宽、设备或主机所能处理的能力，其实是很难反抗攻击的，但仍然有一些方法可以减轻攻击所造成的影响。首先是调查攻击来源，由于黑客由僵尸主机进行攻击，因此，可能无法直接查出黑客是由哪里发动攻击，必须一步一步从被攻击目标往回推，先调查攻击是由管辖网络的哪些边界路由器进来，上一步是外界哪台路由器，联络这些路由器的管理者(可能是某个ISP或电信公司)并寻求他们协助阻挡或查出攻击来源。假如，被攻击的目标只是单一IP，那么试图改个IP并更改其DNSmapping或许可以避开攻击，这是最快速而有效的方式;但是，攻击的目的就是要使正常使用者无法使用服务，更改IP的方式虽然避开攻击，以另一角度来看黑客也达到了他的目的。此外，假如攻击的手法较为单纯，可以由产生的流量找出其规则，那么利用路由器的ACLs(AccessControlLists)或防火墙规则也许可以阻挡，若可以发现流量都是来自同一来源或核心路由器，可以考虑暂时将那边的流量挡起来，当然这还是有可能将正常和异常的流量都一并挡掉，但至少其他来源可以得到正常的服务，这也是不得已的牺牲。此外，还可以考虑增加机器或频宽作为被攻击的缓冲之用，但这只是治标不治本的做法。最重要的是，必须立即着手调查并与相关单位协调解决。

4、预防DDoS攻击

预防DDoS攻击必须透过网络上各个团体和使用者的共同合作，制定更严格的网络标准来解决。每台网络设备或主机都需要随时更新其系统漏洞、关闭不需要的服务、安装必要的防毒和防火墙软件、随时注重系统安全，避免被黑客和自动化的DDoS程序植入攻击程序，以免成为黑客攻击的帮凶。网络管理人员可采取以下方法做好预防工作。

4.1节点

扫描网络管理员要定期扫描网络节点，分析并发现可能存在的安全漏洞，对新出现的漏洞及时进行修补。特别是骨干点的计算机，由于需要占用较高的带宽，因此，对这些主机本身加强主机安全是非常重要的。而且，连接到网络主节点的都是服务器级别的计算机，所以，定期扫描漏洞就变得更加重要了。

4.2配置防火墙

防火墙本身具备了抵御部分DDoS攻击的能力。在发现攻击行为存在时，可以牺牲备用设备引导攻击数据流，这样可以减轻或避免正常业务的顺利进行。当然如果企业或用户对网络的要求很高，笔者建议设立专用的服务器来防止DDoS攻击。

4.3充分利用网络设备

保护网络资源合理配置使用路由器、防火墙等网络设备，它们可将网络有效地保护起来。相对服务器的重启，网络路由器等网络设备的重启要容易得多，而且服务器数据不会有太多的损失。负载均衡技术的使用，可以在攻击发生时自动均衡设备的使用情况，最大限度地降低DDoS的攻击。

4.4过滤服务及端口

在默认情况下，服务器的很多端口是开放的，用户可以使用防火墙或一些管理软件来过滤不必要的服务和端口。只开放服务端口成为保障网络安全的流行做法，例如，用户可能会经常看到一个服务器只开放80端口等。

4.5检查访问者的来源

通过反向路由器查询的方法检查访问者的IP地址是否是真，如果发现虚假IP，应立即将其屏蔽。黑客在攻击时常采用假IP隐藏自己，因此，网络管理员有必要了解自己网络的用户访问情况。

4.6过滤所有被保留的IP地址

我们知道类似10.0.0.0、192.168.0.0和172.16.0.0这样的IP，并不是某个网段的固定IP地址，而是Internet内部保留的区域性IP地址，网络管理员应把被保留的IP过滤掉。

4.7限制SYN/ICMP流量

用户应在防火墙上配置SYN/ICMP的最大流量来限制SYN/ICMP所能占用的最大带宽。当出现大量的超过限定的SYN/ICMP流量时，管理员需要立即排查区分是否存在非法攻击行为。限制SYN/ICMP也是过去对付DDoS攻击最常使用的。据有关报道，电信级运营商已经开始积极运做，准备推出一系列的安全增值服务，IDC服务器托管商也已经积极行动起来，避免用户免受DDoS攻击的侵害，安全厂商更是在积极研究DDoS攻击的原理及防御措施，力求最大限度地扼杀DDoS攻击。笔者认为，任何个体是很难防御住巨大的数据流攻击，只有运营商、企业以及安全厂商共同联合起来，才能更好地克服DDoS给用户带来的伤害。