xinetd(e**X**tended **I**nter**NET** services **D**aemon)

xinetd提供类似于inetd+tcp wrapper的功能，但更加强大、安全。

**1. xinetd特色**

1．强大的存取控制功能

— 内置对恶意用户和善意用户的差别待遇设定。

— 使用libwrap支持，其效能更甚于tcpd。

— 可以限制连接的等级，基于主机的连接数和基于服务的连接数。

— 设置特定的连接时间。

— 将某个服务设置到特定的主机以提供服务。

2．有效防止DoS攻击

— 可以限制连接的等级。

— 可以限制一个主机的最大连接数，从而防止某个主机独占某个服务。

— 可以限制日志文件的大小，防止磁盘空间被填满。

3．强大的日志功能

— 可以为每一个服务就syslog设定日志等级。

— 如果不使用syslog，也可以为每个服务建立日志文件。

— 可以记录请求的起止时间以决定对方的访问时间。

— 可以记录试图非法访问的请求。

4．转发功能

可以将客户端的请求转发到另一台主机去处理。

5．支持IPv6

xinetd自xinetd 2.1.8.8pre\*起的版本就支持IPv6，可以通过在./configure脚本中使用with-inet6 capability选项来完成。注意，要使这个生效，核心和网络必须支持IPv6。当然IPv4仍然被支持。

6．与客户端的交互功能

无论客户端请求是否成功，xinetd都会有提示告知连接状态。

7．Xinetd的缺点

当前，它最大的缺点是对RPC支持的不稳定性，但是可以启动protmap，使它与xinetd共存来解决这个问题。

**2. 使用xinetd启动守护进程**

原则上任何系统服务都可以使用xinetd，然而最适合的应该是那些常用的网络服务，同时，这个服务的请求数目和频繁程度不会太高。像DNS和Apache就不适合采用这种方式，而像FTP、 Telnet、SSH等就适合使用xinetd模式，系统默认使用xinetd的服务可以分为如下几类。

① 标准Internet服务：telnet、ftp。

② 信息服务：finger、netstat、systat。

③ 邮件服务：imap、imaps、pop2、pop3、pops。

④ RPC服务：rquotad、rstatd、rusersd、sprayd、walld。

⑤ BSD服务：comsat、exec、login、ntalk、shell、talk。

⑥ 内部服务：chargen、daytime、echo、servers、services、time。

⑦ 安全服务：irc。

⑧ 其他服务：name、tftp、uucp。

具体可以使用xinetd的服务在/etc/services文件中指出。这个文件的节选内容如下所示：

# /etc/services:

# $Id: services,v 1.40 2004/09/23 05:45:18 notting Exp $

# service-name  port/protocol  [aliases ...]   [# comment]

tcpmux      1/tcp               # TCP port service multiplexer

tcpmux      1/udp               # TCP port service multiplexer

rje     5/tcp                   # Remote Job Entry

rje     5/udp                   # Remote Job Entry

echo        7/tcp

echo        7/udp

discard     9/tcp       sink null

discard     9/udp       sink null

………

**服务名** **端口/协议** **别名      #注释**

Internet 网络服务文件中，记录网络服务名和它们对应使用的端口号及协议。

一般情况下，不要修改该文件的内容，因为这些设置都是Internet标准的设置。一旦修改，可能会造成系统冲突，使用户无法正常访问资源。

Linux系统的端口号的范围为0~65 535，不同范围的端口号有不同的意义。

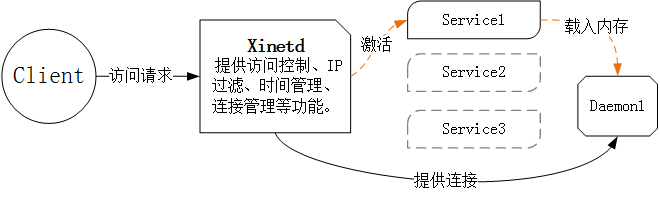
— 0：不使用。

— 1~1023：系统保留，只能由root用户使用。

— 1024~4999：由客户端程序自由分配。            已被使用的端口1024~49151

— 5000~65535：由服务器程序自由分配。          动态和私有端口49151~65535

**3. Xinet工作方式：**

[](http://photo.blog.sina.com.cn/showpic.html#blogid=88cdde9f01019fg5&url=http://s13.sinaimg.cn/orignal/88cdde9fhd0060373532c)

3.1 服务的启动和管理主要有stand alone和super daemon两种。

**stand alone**由启动脚本启动，脚本一般存储在/etc/init.d/，通过/etc/init.d/[service name] restart 启动。服务直接加载在内存中，响应迅速，适合访问量较大的服务。

**super daemon**通过超级守护进程—xinetd来控制服务。系统服务托管在xinetd进程下，当有请求访问端口时，xinetd会启动该服务处理访问请求。同时xinetd还提供监控、访问控制等安全特性，缺点是响应速度较慢。

3.2 xinetd提供的服务处理模式分为两种，一种是多线程，一种是单线程。

multi-threaded：为每一个允许的访问建立一个daemon提供服务。

single-threaded：为所有访问建立一个daemon提供服务。

3.3 Daemon工作状态：

signal-control：只要有客户端请求，就立即执行。如打印机服务(cupsd)

interval-control：每隔一段时间执行。如atd、crond。

**4. 数据包访问服务器资源：**

数据包**->**iptables**->**tcp wrapper**->**服务**->**Selinux**->**权限、ACL**->**本地资源

数据包**->**iptables**->**tcp wrapper**->**xinetd**->**

**└>**服务**->**SeLinux**->**权限、ACL**->**本地资源

**Iptables：**对数据包ip、端口、mac等信息进行过滤。

**Tcp wrapper：**针对tcp包头进行ip、端口等信息的过滤。

**Xinetd：**监控、访问控制、时间管理、链接管理等。

**服务：**服务本身的权限控制、访问控制、资源控制。

**SeLinux：**控制进程对系统资源的访问，进程上下文匹配文件资源上下文标签。

**权限、ACL：**目录以及文件的权限、访问控制列表，也可以实现许多资源控制。

**本地资源：**存储在本地的一系列需要被外部地址访问的资源。银行卡密码等 ^.^~

# grep -i 'disable' /etc/xinetd.d/\* //查看Xinetd服务状态

**5.** **/etc/xinetd.conf — xinetd主配置文件**

defaults

{

# The next two items are intended to be a quick access place to

# temporarily enable or disable services. //启动、停止服务

#       enabled        =

#       disabled        =

# Define general logging characteristics. //日志选项

        log\_type        = SYSLOG daemon info //日志的记录级别（共7级）

        log\_on\_failure  = HOST           //发生错误时记录主机信息

        log\_on\_success  = PID HOST DURATION EXIT       //成功登陆时记录的信息

# Define access restriction defaults //访问控制

#       no\_access       = //禁止访问

#       only\_from       = //仅允许访问

#       max\_load        = 0 //用一个浮点数作为负载系数，当1分钟内负载达到时，该服务将停止处理后续连接，只有Linux、Solaris、FreeBSD支持。

        cps             = 50 10      //每秒超过50个连接，等待10s。

        instances       = 50              //同一个服务的最大被连接数

        per\_source      = 10             //同一客户端的最大连接数

# Address and networking defaults //网络设置

#       bind            = //绑定主机ip地址

#       mdns            = yes //支持组播DNS（multicast DNS），目前只有Mac OS X和linux支持。

        v6only          = no             //仅允许IPv6

# setup environmental attributes //环境属性设置

#       passenv         = //xinetd环境中的环境变量表，该表在激活时传递给服务。

        groups          = yes //设定组名

        umask           = 002 //设定文件权限掩码

# Generally, banners are not used. This sets up their global defaults //设置登录显示的信息

#       banner          = //建立连接时就显示该文件信息。

#       banner\_fail     = //连接失败时，显示文件内的信息。

#       banner\_success  = //连接成功时，显示文件内的信息。

}

includedir /etc/xinetd.d

xinetd守护进程读取/etc/xinetd.d/之中的配置信息，其他未指定的参数均按/etc/xinetd.conf中的默认配置设置。

**6. /etc/xinetd.d/ — xinetd子配置文件**

/etc/xinetd.d/ — xinetd子配置文件，大概都是形如以下格式。

service

{

<<A name=OLE\_LINK7>attribute> <<A name=OLE\_LINK6>assign\_op> ...

}

解释：

1.      Service\_name与/etc/services有关，xinetd会启动与名字对应的端口。

2.      Attribute是一些xinetd管理参数

3.      assign\_op参数设定方法

                   =：设定参数

                   +=：在默认配置中加入这些参数

                   -=：在默认配置中去掉这些参数

**Attribute参数：（待补充）**

|  |  |
| --- | --- |
| Attribute属性 | explain说明 |
| 一般选项：服务启动、启动参数 | |
| Disable | 设定值：[yes|no]   预设：disable = yes  要启动服务，将此参数设置为disable = no |
| Id | 设定值：[服务名称]  对于多个同名的服务，可用id来取代服务名。如/etc/xinetd.d/time-stream |
| Server | 设定值：[program]  启动服务的进程。如server = /usr/bin/rsync |
| server\_args | 设定值：[服务启动参数]  设置服务启动时需要的参数。如rsync：server\_args = --daemon  最终服务启动方式：/usr/bin/rsync --daemon |
| user | 设定值：[使用者账号]  如果xinetd以 root身份管理，这个选项可以指定其他用户。daemon会以指定的用户启动服务。 |
| mdns | 设定值：[yes/no]   预设：yes  支持组播DNS（multicast DNS），目前只有Mac OS X和linux支持。查询主机发送一个组播DNS查询数据包，局域网内被查询的主机响应。（与ARP类似） |
| group | 和user意思相同！ |
| banner | 设定值：[文件路径]  建立连接时，显示该文件中预设信息。 |
| banner\_success | 设定值：[文件路径]  建立连接成功时，显示该文件中预设信息。 |
| banner\_fail | 设定值：[文件路径]  建立连接失败时，显示该文件中预设信息。 |
| 一般选项：封包处理方式 | |
| socket\_type | 设定值：[stream|dgram|raw] 与封包有关  Stream — TCP数据包  dgram — UDP数据包  raw — 服务需要直接访问IP |
| protocol | 设定值：[tcp|udp] 一般使用socket\_type取代这个选项  参考/etc/protocols中的通讯协议，一般使用tcp、udp。 |
| wait | 设定值：[yes(single)|no(multi)]  预设值：wait = no  设定服务的Multi-threaded或single-threaded运行方式。  一般设定为wait = no，允许为每个连接创建一个daemon。  Udp — yes       Tcp — no |
| instances | 设定值：[数字或UNLIMITED]  服务可接受的最大连接数。 |
| per\_source | 设定值：[数字或NULIMITED]  每个IP的最大连接数。 |
| cps | 设定值：[数字1  数字2]  避免短时间内大量的访问请求导致系统过于繁忙，可以设置这个选项。  数字1：一秒内能够接受的最多新连接  数字2：若超过数字1，暂时关闭该服务的秒数。 |
| max\_load | 设定值：浮点数[0 2.5 4.5等]  用一个浮点数作为负载系数，当1分钟内负载达到时，该服务将停止处理后续连接，只有Linux、Solaris、FreeBSD支持。Xinetd编译需要-with-loadavg参数 |
| Flags | 设定值：  [INTERCEPT,NORETRY,IDONLY,NAMEINARGS,NODELAY,KEEPALIVE,NOLIBWRAP,SENSOR,IPv4,IPv6,LABELED,REUSE] 可以使用多个参数    SENSOR:作用是使用一个SENSOR代替当前服务。  需要注意的几个问题：  1．  该服务是管理员不想提供的服务；  2．  服务不能检查的扫描动作；  3．  记录向该服务访问的客户地址，并记录到作用于全局的no\_access列表中，使得请求过该服务的IP在deny\_time过期之前一直都拒绝访问；  4．  xinetd认为该服务server属性是INTERNAL；  5．  使用该标记的socket\_type为stream，需要设置wait为no。 |
| 一般选项：日志记录选项 | |
| log\_type | 设定值：[日志进程 日志等级]  运行服务时，记录的日志信息。默认日志记录等级是info。 |
| Log\_on\_success  log\_on\_failure | 设定值：[PID,HOST,USERID,EXIT,DURATION]  『成功登录』或『失败登录』后记录的内容：  PID：服务进程号；  HOST：远程主机IP  USERID：客户登录的账号  EXIT：断开连接时记录的项目  DURATION：用户使用服务的时间 |
| 高级选项：网络端口、联机机制等 | |
| Env | 设定值：[变量名称=变量内容]  设定一个环境变量，在服务被激活时追加到服务环境变量中。 |
| Passenv | 设定值：[变量1=值1 变量2=值2 变量3=值3]  用空格分开的xinetd环境中的环境变量表，该表在激活时传递给服务程序。 |
| Port | 设定值：[端口号]  设置服务监听的端口，注意port与/etc/services内的记录要相同。 |
| Redirect | 设定值：[远程主机IP port]  将客户端对服务的请求转发到另外一台服务器上去。 |
| Includedir | 设定值：[目录名称]  将外部配置导入当前的配置文件。 |
| Nice | 设定值：[-20至19]  设定服务运行的优先级。 |
| 安全控制 | |
| bind | 设定值：[IP]  绑定主机的一个ip地址，只为它提供服务。多ip地址时有用。 |
| interface | 设定值：[IP]  同上 |
| only\_from | 设定值：[0.0.0.0, 192.168.1.0/24, hostname, domainname]  只允许某些ip访问服务。 |
| no\_access | 设定值：[0.0.0.0, 192.168.1.0/24, hostname, domainname]  拒绝某些主机访问服务。 |
| access\_times | 设定值：[00:00-12:00, HH:MM-HH:MM]  设置服务可访问的时间段。 |
| umask | 设定值：[000, 777, 022]  设定文件权限掩码 |

**6.1 基本属性**

上面的列表是xinetd可用的所有属性，针对一个服务只需要指定几个属性。

xinetd设定服务必需的属性：

|  |  |
| --- | --- |
| 属性 | 适用范围 |
| socket\_type | 所有服务 |
| user | Non\_internal service only 非内部服务 |
| server | Non\_internal service only 非内部服务 |
| wait | 所有服务 |
| protocol | 不在/etc/services中的所有RPC服务和所有其他服务 |
| rpc\_vision | 所有RPC服务 |
| rpc\_number | 不在/etc/rpc中的任何RPC服务 |
| port | 不在/etc/services中的非RPC服务 |

**6.2 支持多操作符的属性**

对于大多数的服务而言，在针对一个服务的设定中操作符只能出现一次，并只支持=操作符，下面六个属性可以支持多个操作符。

支持多操作符的属性：

|  |  |
| --- | --- |
| 属性 | 适用范围 |
| only\_from | 支持所有操作符 |
| no\_access |
| log\_on\_success |
| log\_on\_failure |
| passenv |
| env | 不支持-=操作符 |

**6.3 默认属性**

defaults项是实现为所有服务指定某些属性的默认值。默认值可被每个服务项取消或修改。

可用的defaults属性：

|  |  |
| --- | --- |
| 属性 | 适用范围 |
| log\_on\_success | 可以用=操作符改写，或用+=或-=操作符修改 |
| log\_on\_failure |
| only\_from |
| no\_access |
| passenv |
| instances | 可以用=操作符改写 |
| log\_type |
| disabled | 注销的服务 |
| enabled | 指定启用的服务 |

**6.4 disabled与enabled**

前者的参数是禁用的服务列表，后者的参数是启用的服务列表。共同点是格式相同（属性名、服务名列表与服务中间用空格分开，例如disabled = in.tftpd in.rexecd），此外，它们都是作用于全局的。如果在disabled列表中被指定，那么无论包含在列表中的服务是否有配置文件和如何设置，都将被 禁用；如果enabled列表被指定，那么只有列表中的服务才可启动，如果enabled没有被指定，那么disabled指定的服务之外的所有服务都可 以启动。

**6.5 注意问题**

① 在重新配置的时候，下列的属性不能被改变：socket\_type、wait、protocol、type；

② 如果only\_from和no\_access属性没有被指定（无论在服务项中直接指定还是通过默认项指定），那么对该服务的访问IP将没有限制；

③ 地址校验是针对IP地址而不是针对域名地址。

**7. 对内外网分别应用不同设置—telnet为例**

7.1 要求：

**内网：**

o 绑定内网；

o 对127.0.0.0/8 开放登陆权限；

o 没有连接限制；

o 127.0.0.100和127.0.0.200不允许使用telnet登陆。

**外网：**

o 绑定外网

o 仅开放192.168.0.0/24网络和.edu.cn域可以登陆；

o 开放的时间为早上1-9 点，晚上20-24点；

o 服务最大允许10个连接。

7.2 配置：

service telnet

{

#——————————增加———

disable = yes    #启动

bind = 127.0.0.1       #绑定服务在127.0.0.1这个ip上

only\_from = 127.0.0.0/8           #允许登陆

no\_access = 127.0.0.{100,200}        #限制登陆

instances = UNLIMITED                     #不限制服务最大连接数

#——————————增加———

        flags           = REUSE

        socket\_type     = stream

        wait            = no

        user            = root

        server          = /usr/sbin/in.telnetd

        log\_on\_failure  += USERID

        disable         = yes

}

service telnet

{

#——————————增加———

disable = yes    #启动

bind = 192.168.0.250       #绑定服务在192.168.1.100这个ip上

only\_from = 192.168.0.0/24    #允许访问

only\_from += .edu.cn                #允许访问

access\_times = 01:00-9:00 20:00-23:59 #提供telnet服务的时间

instances = 10          #服务仅提供10个链接

#——————————增加———

        flags           = REUSE

        socket\_type     = stream

        wait            = no

        user            = root

        server          = /usr/sbin/in.telnetd

        log\_on\_failure  += USERID

        disable         = yes

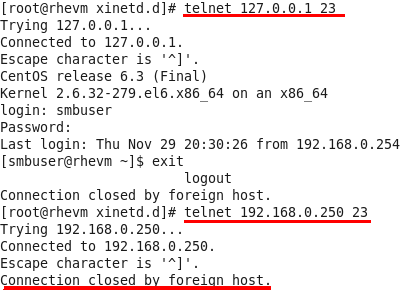
}

7.3 测试：

xinetd监听了2个ip地址。

[xinetd不太详的详解](http://photo.blog.sina.com.cn/showpic.html#blogid=88cdde9f01019fg5&url=http://s5.sinaimg.cn/orignal/88cdde9fhd0060a0667a4)

Xinetd为不同的网段提供了不同的访问控制。

[](http://photo.blog.sina.com.cn/showpic.html#blogid=88cdde9f01019fg5&url=http://s10.sinaimg.cn/orignal/88cdde9fhd0060eb15e89)

7.4 总结：

不同的ip地址相同的服务可以使用相同的端口。一个主机有2块网卡对应不同的网络，xinetd可提供不同的配置。

**8. 利用xinetd设置sensor陷阱，隔离恶意访问**

许多暴露在公网的服务器都承担了巨大的安全风险，时时可可都可能被扫描、被攻击。当有人连接管理员不希望被访问的端口时，xinetd可以记录这个端口，并拒绝这个家伙对服务器中xinetd所托管服务端口的访问请求，这样他就跟你的服务器无缘了。

Xinetd.conf参数中有一项deny\_time是用来设置拒绝时间的，flags选项用来为服务添加标志，不同的标志有不同的功能，具体如下：

**Deny\_time =**

                   直到xinetd重启 | 仅记录日志 | 拒绝X分钟

**Flags=SENSOR** //记录访问服务的IP地址，添加到作用于全局的no\_access列表中，使得请求过该服务的IP在deny\_time过期之前一直都拒绝访问。

**8.1** **实验：**

#yum install rsh-server //服务端

#yum install rsh                 //客户端，提供rsh、rlogin、rcp命令

#yum install telnet-server //telnet用于测试

#yum install telnet

#chkconfig rlogin on

#chkconfig rsh on

#chkconfig telnet on

#vim /etc/xinetd.d/rlogin

service login

{

        disable = no

        socket\_type            = stream

        wait               = no

        user               = root

        log\_on\_success          += USERID

        log\_on\_failure          += USERID

        server              = /usr/sbin/in.rlogind

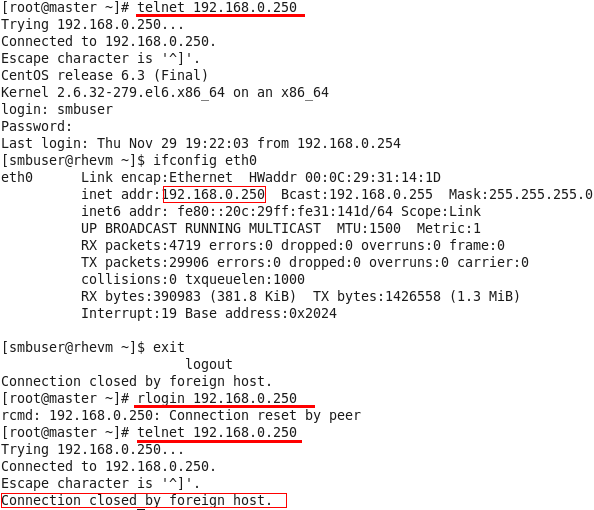
        flags               = SENSOR

        deny\_time             = forever

}

8.2 客户机测试

#yum install telnet rsh

[](http://photo.blog.sina.com.cn/showpic.html#blogid=88cdde9f01019fg5&url=http://s5.sinaimg.cn/orignal/88cdde9fhd0061032dce4)

Xinetd服务实际上监听了rlogin-server的端口，由于带有flags=SENSOR标记，sensor会记录客户ip并将其添加到真对全局的no\_access列表里面去。所以，恶意的访问都不能访问xinetd托管的服务了。

注：重启xinetd服务，禁用ip会失效。

另外一个高级用法是，利用sensor为任意端口做陷阱。

1.修改/etc/services下对应服务的端口

2.在/etc/xinetd.d/telnet配置文件中添加port=80

这样外部访问80端口的请求都会被禁止，可以将apache修改为其他非常规端口。

**9. 建立自定义xinetd托管服务**

1.编写一个test.c程序并编译

#cat /tmp/test.c

#include

main()

{

printf ("hello world\n");

return 0;

}

#gcc /tmp/test.c -o /tmp/test

2.在/etc/xinetd.d/中增加配置文件

#cp telnet test

#vim test

service test\_server

{

        disable = no

        flags           = REUSE

        socket\_type     = stream

        wait            = no

        user            = root

        server          = /tmp/test

        log\_on\_failure  += USERID

        port            =9015

}

3.在/etc/services里添加一行

test\_server        9015/tcp

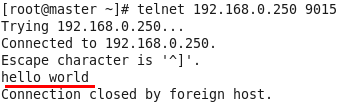
4.重启xinetd服务

#service xinetd restart

5.测试

另一台计算机上运行

#telnet 192.168.0.250 9015

[](http://photo.blog.sina.com.cn/showpic.html#blogid=88cdde9f01019fg5&url=http://s9.sinaimg.cn/orignal/88cdde9fhd006119db5f8)

自定义xinetd服务，需要一定的linux C开发技能，要求熟悉linux C语言的网络开发。这对我来说还是很有难度的。

**10. xinetd防止拒绝服务攻击（Denial of Services）的原因**

xinetd能有效地防止拒绝服务攻击（Denial of Services）的原因如下。

1．限制同时运行的进程数

通过设置instances选项设定同时运行的并发进程数：

**Instances=20**

当服务器被请求连接的进程数达到20个时，xinetd将停止接受多出部分的连接请求。直到请求连接数低于设定值为止。

2．限制一个IP地址的最大连接数

通过限制一个主机的最大连接数，从而防止某个主机独占某个服务。

**per\_source=5**

这里每个 IP地址的连接数是5个。

3．限制日志文件大小，防止磁盘空间被填满

许多攻击者知道大多数服务需要写入日志。入侵者可以构造大量的错误信息并发送出来，服务器记录这些错误，可能就造成日志文件非常庞大，甚至会塞满硬盘。同时会让管理员面对大量的日志，而不能发现入侵者真正的入侵途径。因此，限制日志文件大小是防范拒绝服务攻击的一个方法。

**log\_type FILE.1 /var/log/myservice.log 8388608 15728640**

这里设置的日志文件FILE.1临界值为8MB，到达此值时，syslog文件会出现告警，到达15MB，系统会停止所有使用这个日志系统的服务。

4．限制负载

xinetd 还可以使用限制负载的方法防范拒绝服务攻击。用一个浮点数作为负载系数，当负载达到这个数目的时候，该服务将暂停处理后续的连接。

**max\_load = 2.8**

上面的设定表示当一项系统负载达到2.8时，所有服务将暂时中止，直到系统负载下降到设定值以下。

注意：

要使用这个选项，编译时应加入“--with-loadavg”，xinetd将处理max-load配置选项，从而在系统负载过重时关闭某些服务进程，来实现防范某些拒绝服务攻击。

5．限制所有服务器数目（连接速率）

xinetd 可以使用cps选项设定连接速率，下面的例子：

**cps = 25 60**

上面的设定表示服务器每秒最多启动25个连接，如果达到这个数目将停止启动新服务60秒。在此期间不接受任何请求。

6．限制对硬件资源的利用

通过rlimit\_as和rlimit\_cpu两个选项可以有效地限制一种服务对内存、中央处理器的资源占用：

**rlimit\_as = 8M**

**rlimit\_cpu=20**

上面的设定表示对服务器硬件资源占用的限制，最多可用内存为8MB，CPU每秒处理20个进程。

   xinetd的一个重要功能是它能够控制从属服务可以利用的资源量，通过设置可以达到这个目的，有助于防止某个xinetd服务占用大量资源，从而导致“拒绝服务”情况的出现。