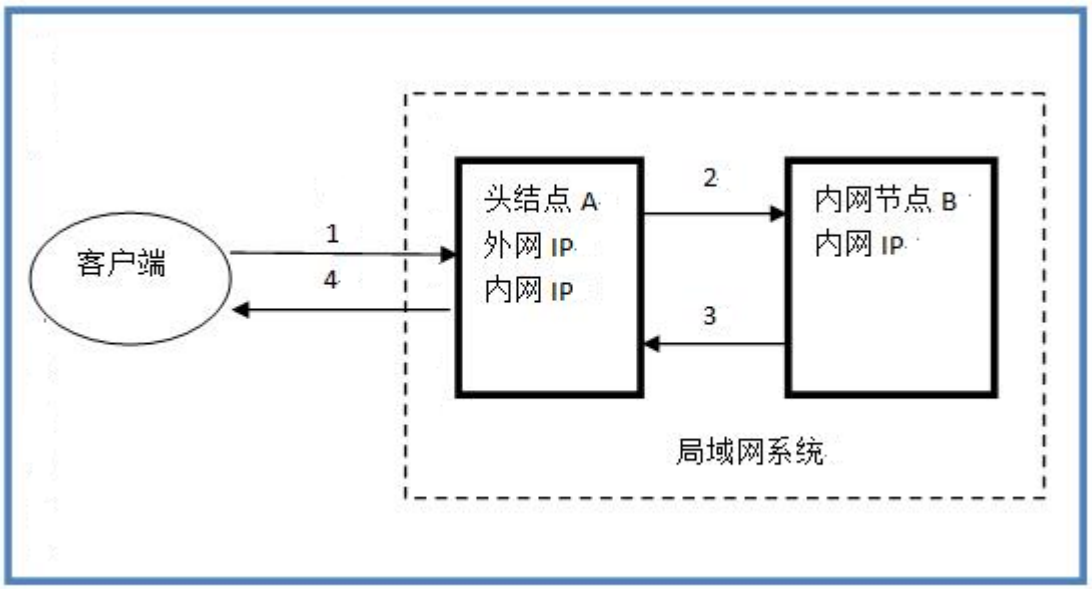
端口转发工具



目录

1) iptables 端口转发（测试有问题）.....1

2) rinetd 工具（测试可行）.....2

3） ngix 代理.....4

1) iptables 端口转发（测试有问题）

外网机器:

内网 ip: 10.1.1.42 eth0

外网 ip: 202.120.2.110 eth1

端口: 1689(TCP 端口)

内网机器:

内网 ip: 10.1.1.27 eth0

端口: 1688(TCP 端口)

配置 iptables 实现端口转发

iptables 的配置文件是放在 /etc/sysconfig/iptables 里的，使用类似 WEB 端口转发方案即可

1.开启内核转发，重启后还原

echo 1 > /proc/sys/net/ipv4/ip_forward

2.执行如下四条 iptables 规则

用 DNAT 作端口映射

iptables -t nat -A PREROUTING -d 202.120.2.110 -p tcp --dport 1689 -j DNAT --to-destination 10.1.1.27:1688

用 SNAT 作源地址转换（关键），以使回应包能正确返回

iptables -t nat -A POSTROUTING -d 10.1.1.27 -p tcp --dport 1688 -j SNAT --to 10.1.1.42

一些人经常忘了打开 FORWARD 链的相关端口，特此增加

iptables -A FORWARD -o eth0 -d 10.1.1.27 -p tcp --dport 1688 -j ACCEPT

iptables -A FORWARD -i eth0 -s 10.1.1.27 -p tcp --sport 1688 -j ACCEPT

3.保存 iptables 规则

/etc/init.d/iptables save

4.检查配置结果

iptables -t nat -L

5 重启 iptables

service iptables restart

6.查看状态

service iptables status

7.测试访问

vlmcs -v -l W7 202.120.2.110:1689

2) rinetd 工具（测试可行）

官网地址 http://www.boutell.com/rinetd

软件下载 wget http://www.boutell.com/rinetd/http/rinetd.tar.gz

安装：

tar zxvf rinetd.tar.gz

make

make install

编辑配置文件：

vim /etc/rinetd.conf

0.0.0.0 38000 172.19.94.3 38000

1.2.3.4 80 192.168.0.10 80

注：配置文件格式

bindaddress bindport connectaddress connectport

绑定的地址 绑定的端口 连接的地址 连接的端口

[Source Address] [Source Port] [Destination Address] [Destination Port]

源地址 源端口 目的地址 目的端口

0.0.0.0 表示本机绑定所有可用地址

将所有发往本机 38000 端口的请求转发到 172.19.94.3 的 38000 端口

将所有发往 1.2.3.4 的 80 端口请求转发到 192.168.0.10 的 80 端口

启动转发： rinetd -c /etc/rinetd.conf

关闭转发： pkill rinetd

注：把这条命令加到/etc/rc.local 里面就可以开机自动运行

查看端口状态命令：netstat -antup

需要注意

1.rinetd.conf 中绑定的本机端口必须没有被其它程序占用

2.运行 rinetd 的系统防火墙应该打开绑定的本机端口

例如：

-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 1111 -j ACCEPT

-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 222 -j ACCEPT

3） ngix 代理

官网：http://nginx.org/

安装：

（一）二进制安装

直接到官网下载相应的二进制文件即可（http://nginx.org/en/download.html）

（二）源码安装

安装准备：

1) gcc, g++
2) 安装 PCRE 库

   ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/ 下载最新的 PCRE 源码包，使用下面命令下载编译和安装 PCRE 包：

   2.1) wget ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-8.37.tar.gz

   2.2) tar -zxvf pcre-8.37.tar.gz

   2.3) cd pcre-8.34

   2.4) ./configure

   2.5) make

   2.6) make install
3) 安装 zlib
4) 安装 ssl
5) 安装 nginx

   5.1) wget http://nginx.org/download/nginx-1.4.2.tar.gz

   5.2) tar -zxvf nginx-1.4.2.tar.gz

   5.3) cd nginx-1.4.2

   5.4) ./configure --sbin-path=/usr/local/nginx/nginx --conf-path=/usr/local/nginx/nginx.conf --pid-path=/usr/local/nginx/nginx.pid --with-http_ssl_module --with-pcre=/opt/app/openet/oetal1/chenhe/pcre-8.37 --with-openssl=/opt/app/openet/oetal1/chenhe/openssl-1.0.1t

--with-zlib=/opt/app/openet/oetal1/chenhe/zlib-1.2.8

5.5) make

5.6) make install

--with-pcre=/usr/src/pcre-8.34 指的是 pcre-8.34 的源码路径。

--with-zlib=/usr/src/zlib-1.2.7 指的是 zlib-1.2.7 的源码路径。

nginx 编译选项说明：

make 是用来编译的，它从 Makefile 中读取指令，然后编译。

make install 是用来安装的，它也从 Makefile 中读取指令，安装到指定的位置。

configure 命令是用来检测你的安装平台的目标特征的。它定义了系统的各个方面，包括 nginx 的被允许使用的连接处理的方法，比如它会检测你是不是有 CC 或 GCC，并不是需要 CC 或 GCC，它是个 shell 脚本，执行结束时，它会创建一个 Makefile 文件。nginx 的 configure 命令支持以下参数：

--prefix=path 定义一个目录，存放服务器上的文件，也就是 nginx 的安装目录。默认使用 /usr/local/nginx。

--sbin-path=path 设置 nginx 的可执行文件的路径，默认为 prefix/sbin/nginx.

--conf-path=path 设置在 nginx.conf 配置文件的路径。nginx 允许使用不同的配置文件启动，通过命令行中的-c 选项。默认为 prefix/conf/nginx.conf.

--pid-path=path 设置 nginx.pid 文件，将存储的主进程的进程号。安装完成后，可以随时改变的文件名，在 nginx.conf 配置文件中使用 PID 指令。默认情况下，文件名为 prefix/logs/nginx.pid.

--error-log-path=path 设置主错误，警告，和诊断文件的名称。安装完成后，可以随时改变的文件名，在 nginx.conf 配置文件中使用的 error_log 指令。默认情况下，文件名为 prefix/logs/error.log.

--http-log-path=path 设置主请求的 HTTP 服务器的日志文件的名称。安装完成后，可以随时改变的文件名，在 nginx.conf 配置文件中使

用 的 access_log 指令。默认情况下，文件名 为 prefix/logs/access.log.

--user=name 设置 nginx 工作进程的用户。安装完成后，可以随时更改的名称在 nginx.conf 配置文件中 使用的 user 指令。默认的用户名是 nobody。

--group=name 设置 nginx 工作进程的用户组。安装完成后，可以随时更改的名称在 nginx.conf 配置文件中 使用的 user 指令。默认的为非特权用户。

--with-select_module --without-select_module 启用或禁用构建一个模块来允许服务器使用 select()方法。该模块将自动建立，如果平台不支持的 kqueue，epoll，rtsig 或/dev/poll。

--with-poll_module --without-poll_module 启用或禁用构建一个模块来允许服务器使用 poll()方法。该模块将自动建立，如果平台不支持的 kqueue，epoll，rtsig 或/dev/poll。

--without-http_gzip_module — 不编译压缩的 HTTP 服务器的响应模块。编译并运行此模块需要 zlib 库。

--without-http_rewrite_module 不编译重写模块。编译并运行此模块需要 PCRE 库支持。

--without-http_proxy_module — 不编译 http_proxy 模块。

--with-http_ssl_module — 使用 https 协议模块。默认情况下，该模块没有被构建。建立并运行此模块的 OpenSSL 库是必需的。

--with-pcre=path — 设置 PCRE 库的源码路径。PCRE 库的源码（版本 4.4 - 8.30）需要从 PCRE 网站下载并解压。其余的工作是 Nginx 的./configure 和 make 来完成。正则表达式使用在 location 指令和 ngx_http_rewrite_module 模块中。

--with-pcre-jit —编译 PCRE 包含“just-in-time compilation”（1.1.12 中，pcre_jit 指令）。

--with-zlib=path —设置的 zlib 库的源码路径。要下载从 zlib（版本 1.1.3 - 1.2.5）的并解压。其余的工作是 Nginx 的./ configure 和 make 完成。ngx_http_gzip_module 模块需要使用 zlib 。

--with-cc-opt=parameters — 设置额外的参数将被添加到CFLAGS变量。例如,当你在 FreeBSD 上使用 PCRE 库时需要使用:--with-cc-opt="-I /usr/local/include 。 .如需要需要增加 select() 支持的文件数量:--with-cc-opt="-D FD_SETSIZE=2048".

--with-ld-opt=parameters —设置附加的参数,将用于在链接期间。例如,当在 FreeBSD 下使用该系统的 PCRE 库,应指定:--with-ld-opt="-L /usr/local/lib".

典型实例(下面为了展示需要写在多行，执行时内容需要在同一行)

配置：

编辑 conf/nginx.conf 文件：

```
worker_processes  1;
events {
    worker_connections  1024;
}
http {
    include       mime.types;
    default_type  application/octet-stream;
    sendfile        on;
    keepalive_timeout  65;
    server {
        listen       80;
        server_name  127.0.0.1:8080;

        location / {
            proxy_pass   http://127.0.0.1:8080;
        }
    }
}
```

server 下的结点：

listen：监听 80 端口

server_name：转发到哪个地址

proxy_pass：代理到哪个地址

作用:

把网站端口改成非 80 的端口，然后通过 nginx 转发到域名上，这样就不用在域名后面加端口才能访问

nginx 常用命令（要进入到 nginx 的目录）：

开启：start nginx

重启：nginx -s reload