Nagios的安装与配置

[安装环境 2](#_Toc363141327)

[nagios的安装 2](#_Toc363141328)

[创建nagios用户 2](#_Toc363141329)

[安装apache和php 2](#_Toc363141330)

[安装Nagios 3](#_Toc363141331)

[Nagios目录介绍 3](#_Toc363141332)

[安装plugins插件 3](#_Toc363141333)

[安装nrpe 3](#_Toc363141334)

[安装pnp4nagios-0.6.6插件 4](#_Toc363141335)

[Nagios的配置 6](#_Toc363141336)

[配置apache 6](#_Toc363141337)

[创建apache目录验证文件 7](#_Toc363141338)

[配置nagios 7](#_Toc363141339)

[修改cgi.cfg 7](#_Toc363141340)

[修改nagios.cfg 7](#_Toc363141341)

[修改commands.cfg 7](#_Toc363141342)

[修改contacts.cfg 8](#_Toc363141343)

[修改contactgroups.cfg 8](#_Toc363141344)

[添加被监控机 8](#_Toc363141345)

[监控机配置 8](#_Toc363141346)

[修改hosts.cfg 8](#_Toc363141347)

[修改services.cfg 9](#_Toc363141348)

[被监控机配置 9](#_Toc363141349)

[被监控机安装plugin 9](#_Toc363141350)

[被监控机安装nrpe 10](#_Toc363141351)

[配置 10](#_Toc363141352)

[nagios事件自动处理 11](#_Toc363141353)

[报警通知 13](#_Toc363141354)

[邮件报警通知 13](#_Toc363141355)

[安装配置sendmail 13](#_Toc363141356)

[nagios邮件报警配置 14](#_Toc363141357)

[短信报警通知 16](#_Toc363141358)

[安装飞信机器人 16](#_Toc363141359)

[Nagios短信报警配置 17](#_Toc363141360)

[故障排除 18](#_Toc363141361)

[邮件和短信分时报警 18](#_Toc363141362)

# 安装环境

**环境说明：**

监控机centos6.2 x86\_64

ip为172.26.188.201

被监控机centos6.2 x86\_64

ip为 172.26.188.202

**所用软件包：**

nagios-3.4.1.tar.gz

nagios-plugins-1.4.16.tar.gz

nrpe-2.13.tar.gz

pnp4nagios-0.6.19.tar.gz

下载地址

<http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz>

<http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-3.5.1.tar.gz>

http://downloads.sourceforge.net/project/pnp4nagios/PNP-0.6/pnp4nagios-0.6.24.tar.gz <http://downloads.sourceforge.net/project/nagios/nrpe-2.x/nrpe-2.15/nrpe-2.15.tar.gz>

http://sourceforge.net/projects/nagios/files/

# nagios的安装

## **创建nagios用户**

#useradd nagios

#mkdir /usr/local/nagios

#chown –R nagios.nagios /usr/local/nagios

## 安装apache和php

#yum install httpd

#yum install php php-mysql php-mbstring  php-gd\*

## 安装Nagios

tar -zxvf nagios-3.4.1.tar.gz

cd nagios

./configure --prefix=/usr/local/nagios

make all

make install

make install-init

make install-commandmode

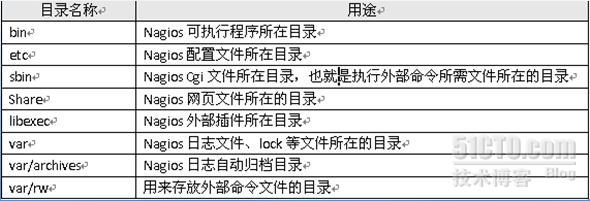
make install-config

#make install-webconf

#make install-exfoliation

## Nagios目录介绍

Nagios安装完成后，各个目录结构以及功能说明如下表所示：



## 安装plugins插件

这里下载的版本是nagios-plugins-1.4.16。

注意：插件版本与nagios版本的关联并不大。

#tar –zxvf nagios-plugins-1.4.16.tar.gz

#cd nagios-plugins-1.4.16

#./configure --prefix=/usr/local/nagios

# make

# make install

安装完成，在/usr/local/nagios下的libexec目录下，生成很多可执行文件，这些正是nagios所需要的插件。

## 安装nrpe

#yum install openssl

#tar -zxvf nrpe-2.13.tar.gz

#cd nrpe-2.13

#./configure --prefix=/usr/local/nagios

#make all

#make install-plugin

## 安装pnp4nagios-0.6.6插件

#yum install perl-Time-HiRes rrdtool

#tar -zxvf pnp4nagios-0.6.19.tar.gz  
#cd pnp4nagios-0.6.19  
# ./configure --with-nagios-user=nagios --with-nagios-group=nagios

make all  
make install

make install-webconf

make install-config

make install-init

创建配置文件

#cd /usr/local/pnp4nagios/etc

#cp misccommands.cfg-sample misccommands.cfg

#cp nagios.cfg-sample nagios.cfg

#cp rra.cfg-sample rra.cfg

#cd pages

#cp web\_traffic.cfg-sample web\_traffic.cfg

# cd ../check\_commands

#cp check\_all\_local\_disks.cfg-sample check\_all\_local\_disks.cfg

#cp check\_nrpe.cfg-sample check\_nrpe.cfg

#cp check\_nwstat.cfg-sample check\_nwstat.cfg

重启服务

#/etc/init.d/npcd restart

修改 nagios 的配置文件.打开performance\_data

#vi /usr/local/nagios/etc/nagios.cfg

打开注释项：

process\_performance\_data=1

host\_perfdata\_command=process-host-perfdata

service\_perfdata\_command=process-service-perfdata

修改 commands.cfg

#vi /usr/local/nagios/etc/objects/commands.cfg

##添加

# 'process-host-perfdata' command definition

define command{

        command\_name    process-host-perfdata

        command\_line    /usr/local/pnp4nagios/libexec/process\_perfdata.pl

        }

# 'process-service-perfdata' command definition

define command{

        command\_name    process-service-perfdata

        command\_line    /usr/local/pnp4nagios/libexec/process\_perfdata.pl

        }

添加小太阳模版，镶嵌在nagios页面上。

#vi /usr/local/nagios/etc/objects/templates.cfg

## 在最后添加

define host {

name       host-pnp

action\_url /pnp4nagios/graph?host=$HOSTNAME$&srv=\_HOST\_

register   0

}

define service {

name       srv-pnp

action\_url /pnp4nagios/graph?host=$HOSTNAME$&srv=$SERVICEDESC$

register   0

}

最后在hosts.cfg和services.cfg中调用host-pnp和srv-pnp模板

#vi hosts.cfg

define host {

use host-pnp

host\_name nagios-test1

alias nagios test1

address 172.26.188.202

contact\_groups admins

check\_command check-host-alive

max\_check\_attempts 5

notification\_interval 10

notification\_period 24x7

notification\_options d,u,r

}

#vi services.cfg

define service {

use srv-pnp

host\_name nagios-test1

service\_description check\_tcp 80

check\_period 24x7

max\_check\_attempts 4

normal\_check\_interval 3

retry\_check\_interval 2

contact\_groups admins

notification\_interval 10

notification\_period 24x7

notification\_options w,u,c,r

event\_handler\_enabled 1

event\_handler restart-httpd

check\_command check\_tcp!80

}

# Nagios的配置

## 配置apache

#vi/etc/httpd/conf/httpd.conf

找到：ServerName [www.example.com:80](http://www.example.com/)

修改为：

ServerName 172.26.188.201

找到：

User nobody

Group #-1

修改为：

User nagios

Group nagios

找到：

DirectoryIndex index.html index.html.var

修改为

DirectoryIndex index.html index.php

接着增加如下内容：

AddType application/x-httpd-php .php

为了安全其间，一般情况下要让nagios的web监控界面必须经过授权才能访问，这需要增加验证配置，即在httpd.conf文件最后添加如下信息：

#setting for nagios

ScriptAlias /nagios/cgi-bin "/usr/local/nagios/sbin"

<Directory "/usr/local/nagios/sbin">

AuthType Basic

Options ExecCGI

AllowOverride None

Order allow,deny

Allow from all

AuthName "Nagios Access"

AuthUserFile /usr/local/nagios/etc/htpasswd

Require valid-user

</Directory>

Alias /nagios "/usr/local/nagios/share"

<Directory "/usr/local/nagios/share">

AuthType Basic

Options None

AllowOverride None

Order allow,deny

Allow from all

AuthName "nagios Access"

AuthUserFile /usr/local/nagios/etc/htpasswd

Require valid-user

</Directory>

## 创建apache目录验证文件

在上面的配置中，指定了目录验证文件htpasswd，下面要创建这个文件：

# htpasswd -c /usr/local/nagios/etc/htpasswd ixdba

New password: (输入密码)

Re-type new password: (再输入一次密码)

Adding password for user ixdba

这样就在/usr/local/nagios/etc目录下创建了一个htpasswd验证文件，当通过<http://ip/nagios/>访问时就需要输入用户名和密码了。

## 配置nagios

### 修改cgi.cfg

refresh\_rate=30 #nagios主页的刷新时间，我设置成30秒自动刷新

use\_authentication=1 #开启认证功能

default\_user\_name=ixdba

### 修改nagios.cfg

去掉以下注释，没有的行添加上去：

cfg\_file=/usr/local/nagios/etc/objects/commands.cfg #nagios可调用的监控命令

cfg\_file=/usr/local/nagios/etc/objects/contacts.cfg #联系人配置

cfg\_file=/usr/local/nagios/etc/objects/contactgroups.cfg #联系人组配置

cfg\_file=/usr/local/nagios/etc/objects/timeperiods.cfg #监控时间配置

cfg\_file=/usr/local/nagios/etc/objects/templates.cfg #模板配置

cfg\_file=/usr/local/nagios/etc/objects/hosts.cfg #监控主机配置

cfg\_file=/usr/local/nagios/etc/objects/services.cfg #监控服务配置

check\_external\_commands=1 #允许web下重启nagios

command\_check\_interval=10s #命令检查时间间隔

### 修改commands.cfg

添加nrpe的定义：

# 'check\_nrpe ' command definition

define command{

command\_namecheck\_nrpe

command\_line $USER1$/check\_nrpe -H $HOSTADDRESS$ -c $ARG1$

}

### 修改contacts.cfg

添加联系人信息，内容如下：

define contact {

contact\_name test1

alias system administrator

service\_notification\_period 24x7

host\_notification\_period 24x7

service\_notification\_options w,u,c,r

host\_notification\_options d,u,r

service\_notification\_commands notify-service-by-email

host\_notification\_commands notify-host-by-email

email test1@test.com

}

### 修改contactgroups.cfg

添加联系人组信息，内容如下:

define contactgroup {

contactgroup\_name admins

alias system administrator group

members test1

}

# 添加被监控机

## 监控机配置

首先/usr/local/nagios/libexec/check\_nrpe -H $被监控主机ip

应该都能输出 NRPE 的版本： NRPE v2.13

nagios的全部配置文件在：/usr/local/nagios/etc目录下面

### 修改hosts.cfg

hosts.cfg定义被监控机的主机信息，配置如下：

define host {

use host-pnp

host\_name nagios-test1

alias nagios test1

address 172.26.188.202

contact\_groups admins

check\_command check-host-alive

max\_check\_attempts 5

notification\_interval 10

notification\_period 24x7

notification\_options d,u,r

}

### 修改services.cfg

services.cfg定义被监控机的监控项，配置如下：

define service {

use srv-pnp

host\_name nagios-test1

service\_description check\_tcp 80

check\_period 24x7

max\_check\_attempts 4

normal\_check\_interval 3

retry\_check\_interval 2

contact\_groups admins

notification\_interval 10

notification\_period 24x7

notification\_options w,u,c,r

event\_handler\_enabled 1

event\_handler restart-httpd

check\_command check\_tcp!80

}

## 被监控机配置

### 被监控机安装plugin

#groupadd nagios

#useradd -g nagios -d /usr/local/nagios nagios

#tar –zxvf nagios-plugins-1.4.16.tar.gz

#cd nagios-plugins-1.4.16

#./configure --with-nagios-user=nagios --with-nagios-group=nagios

# make

# make install

### 被监控机安装nrpe

tar -zxvf nrpe-2.13.tar.gz

cd nrpe-2.13

./configure --prefix=/usr/local/nagios

make all

make install-plugin

make install-daemon

make install-daemon-config

chown -R nagios:nagios /usr/local/nagios

将nrpe加入xinetd启动

#vi /etc/xinetd.d/nrpe

service nrpe

{

flags = REUSE

socket\_type = stream

port = 5666

wait = no

user = nagios

group = nagios

server = /usr/local/nagios/bin/nrpe

server\_args = -c /usr/local/nagios/etc/nrpe.cfg --inetd

log\_on\_failure += USERID

disable = no

only\_from = 127.0.0.1 172.26.188.201

}

在/etc/services对应位置加入如下行：

nrpe 5666/tcp # NRPE

启动nrpe

service xinetd restart

### 配置

被监控机主要修改了以下这几个文件

/usr/local/nagios/etc/nrpe.cfg

/etc/hosts.allow

/etc/rc.local

1. 配置 NRPE

首先查看/usr/local/nagios/libexec文件夹下面是否有监控脚本命令，这些脚本既为nrpe.cfg中定义命令需要调用的脚本

vi /usr/local/nagios/etc/nrpe.cfg

allowed\_hosts=172.26.188.201 #监控机IP

server\_address=172.26.188.202 #本机IP

在nrpe.cfg中添加自定义的命令

以下条命令为例

Command[check\_sdb2]=/usr/local/nagios/libexec/check\_disk–w 20% -c 10% -p /dev/sdb2

监测硬盘使用量，check\_sdb2为自定义名，check\_disk为调用的脚本，-w是警告，剩余20%容量时，-c，10%为严重值，-p指定要监控的硬盘

1. 修改/etc/hosts.allow增加监控机ip

echo 'nrpe:监控机ip ' >> /etc/hosts.allow

1. 启动 NRPE 守护进程：

/usr/local/nagios/bin/nrpe -c /usr/local/nagios/etc/nrpe.cfg–d

1. 设置开机自动启动NRPE。

echo "/usr/local/nagios/bin/nrpe -c /usr/local/nagios/etc/nrpe.cfg -d" >> /etc/rc.local

查看相应的端口：netstat -an |grep 5666

你会发现5666端口已开启。

**要点：**在添加被监控机时，**首先**要确定被监控机nrpe.cfg文件里有添加此命令行，前提是在被监控机的/usr/local/nagios/libexec文件夹下面有相应的监控命令的脚本；**其次**在监控机的command.cfg文件配置里也要有相应的配置，既依照文件里面nrpe的定义文件来添加相应的监控命令；**最后**是监控机的services.cfg文件中定义具体的监控项。

# nagios事件自动处理

## nagios通过SSH执行event\_hander

nagios使用event handlers来在任何人收到通知之前由Nagios做一些前期故障修复。

事件处理命令可以用shell或是perl脚本，脚本中应该处理以下宏：  
对服务的：$SERVICESTATE$、$SERVICESTATETYPE$和$SERVICEATTEMPT$；  
对主机的：$HOSTSTATE$、$HOSTSTATETYPE$和$HOSTATTEMPT$。  
脚本须检测这些作为命令行参数传入的值，并采取必要动作来处理这些值。

下面以nagios监控apache服务，当apache停止时自动启动apache。详细配置步骤如下:

1. 配置在Nagios监控机上无密码登录远程被监控机

* 生成ssh密钥文件

#su - nagios

$ ssh-keygen -t rsa

# 下面一直回车，不要设置密码

Generating public/private rsa key pair.

Enter file in which to save the key (/home/nagios/.ssh/id\_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/nagios/.ssh/id\_rsa.

Your public key has been saved in /home/nagios/.ssh/id\_rsa.pub.

The key fingerprint is:

d2:82:61:12:53:f9:53:75:77:8d:32:c0:ca:c8:20:60 nagios@nagios.itech.com

* 将生成的密钥拷贝到要远程被监控主机上

$ scp .ssh/id\_rsa.pub nagios-test1:/usr/local/nagios/

* 在要远程登录的被监控机器上配置公钥

$ ssh nagios@nagios-test1

$ nagios@nagios-test1's password:

$ cat id\_rsa.pub >> .ssh/authorized\_keys

$ chmod 600 .ssh/authorized\_keys

$ exit

* 测试无密码登录

$ ssh nagios@nagios-test1

1. 在远程被监控机器上配置sudo

使nagios用户可以以root身份运行/usr/local/nagios/libexec/eventhandlers/restart-httpd脚本

# visudo

添加如下行：

nagios ALL=(root) NOPASSWD:/usr/local/nagios/libexec/eventhandlers/restart-httpd

注释如下行：

#Defaults requiretty

1. 在远程机器上编写apache重启脚本  
   vi /usr/local/nagios/libexec/eventhandlers/restart-httpd

内容如下:

#!/bin/sh

#

# Event handler script for restarting the Apache server on the remote machine

#

# Note: This script will only restart the Apache server if the service is

# retried 2 times (in a "soft" state) or if the web service somehow

# manages to fall into a "hard" error state.

#

#

# What state is the Apache service in?

case "$1" in

OK)

;;

WARNING)

;;

UNKNOWN)

;;

CRITICAL)

# Is this a "soft" or a "hard" state?

case "$2" in

SOFT)

# What check attempt are we on? We don't want to restart the Apache server on the first

# check, because it may just be a fluke!

case "$3" in

2)

echo -n "Restarting Apache service..."

/sbin/service httpd restart

;;

esac

;;

HARD)

echo -n "Restarting Apache service..."

/sbin/service httpd restart

;;

esac

;;

esac

exit 0

1. 配置Nagios监控机上的配置文件

* 修改nagios.cfg

enable\_event\_handlers=1 #开启全局事件处理

* 在命令配置文件中定义重启apache的命令

# cd /usr/local/nagios/etc/

#vi objects/commands.cfg

#restart the apache service on a remote server

define command{

command\_name restart-httpd

command\_line /usr/bin/ssh nagios@$HOSTADDRESS$ "sudo /usr/local/nagios/libexec/eventhandlers/restart-httpd $SERVICESTATE$ $SERVICESTATETYPE$ $SERVICEATTEMPT$"

}

* 配置主机监控文件

修改services.cfg配置，加入事件处理配置  
#vi /usr/local/nagios/etc/objects/services.cfg

define service {

use srv-pnp

host\_name nagios-test1

service\_description check\_tcp 80

check\_period 24x7

max\_check\_attempts 4

normal\_check\_interval 3

retry\_check\_interval 2

contact\_groups admins

notification\_interval 10

notification\_period 24x7

notification\_options w,u,c,r

event\_handler\_enabled 1 #启用事件处理

event\_handler restart-httpd #事件处理执行的命令

check\_command check\_tcp!80

}

1. 测试

在被监控机停止apache服务，等待几分钟后检查被监控机的apache服务是否自动启动。

## Nagios通过NRPE执行event\_hander

1. 在远程机器上编写service重启脚本  
   vi /usr/local/nagios/libexec/service\_hander

内容如下:

#!/bin/sh

#

# Event handler script for restarting the service on the remote machine

#

# Note: This script will only restart service if the service is

# retried 2 times (in a "soft" state) or if the services somehow

# manages to fall into a "hard" error state.

#

#

# What state is service in?

case "$1" in

OK)

;;

WARNING)

;;

UNKNOWN)

;;

CRITICAL)

# Is this a "soft" or a "hard" state?

case "$2" in

SOFT)

# What check attempt are we on? We don't want to restart the service on the first

# check, because it may just be a fluke!

case "$3" in

2)

echo -n "Restarting $4 service..."

/sbin/service $4 restart

;;

esac

;;

HARD)

echo -n "Restarting $4 service..."

/sbin/service $4 restart

;;

esac

;;

esac

exit 0

重新编译nrpe，需要让nrpe传递参数

./configure --prefix=/usr/local/nagios --enable-command-args

修改nrpe.cfg

vi /usr/local/nagios/etc/nrpe.cfg

dont\_blame\_nrpe 0改为1

command[restart\_service]=/usr/local/nagios/libexec/service\_hander $ARG1$ $ARG2$ $ARG3$ $ARG4$

配置完nrpe后需要重启启动nrpe服务

服务端配置（修改ssh方式下的command.cfg）

vi /usr/local/nagios/etc/objects/commands.cfg

找到以下内容：

define command{

command\_name restart-httpd

command\_line /usr/bin/ssh nagios@$HOSTADDRESS$ "sudo /usr/local/nagios/libexec/eventhandlers/restart-httpd $SERVICESTATE$ $SERVICESTATETYPE$ $SERVICEATTEMPT$"

}

修改为：

define command{

command\_name restart-httpd

command\_line /usr/local/nagios/libexec/check\_nrpe -H 172.26.188.202 -c restart\_service -a $SERVICESTATE$ $SERVICESTATETYPE$ $SERVICEATTEMPT$ httpd

}

# 报警通知

## 邮件报警通知

### 安装配置sendmail

sendmail安装

yum -y install sendmail sendmail-cf sendmail-devel dovecot cyrus-sasl cyrus-sasl-lib cyrus-sasl-plain

sendmail配置

* + 配置MTA

vim /etc/mail/sendmail.mc

A、找到DAEMON\_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl

修改为DAEMON\_OPTIONS(`Port=smtp,Addr=0.0.0.0, Name=MTA')dnl

B、找到TRUST\_AUTH\_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl

define(`confAUTH\_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl

去掉注释及前面的dnl，把两行移到最前面。

* + 生成sendmail.cf主配置文件。

mv /etc/mail/sendmail.cf /etc/mail/sendmail.cf-bak  
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf

* + 设置local-host-names文件

echo "nagios-server.com" >>/etc/mail/local-host-names  
此处abc.com是域名，即DNS解析MX解析的域名

* + 设置访问控制access和access.db。

Acces的配置为：地址 控制

Vi /etc/mail/access

localhost.localdomain RELAY

localhost RELAY

127.0.0.1 RELAY

192.168.0.71 RELAY

192.168.0.129 RELAY

Makemap hash access < access

* + 添加账户：

groupadd usermail  
adduser -g usermail -s /sbin/nologin user1  
passwd user1

* + 设置别名

vim /etc/aliases

### nagios邮件报警配置

* 配置邮件报警command

#vi /usr/local/nagios/etc/objects/commands.cfg

# 'notify-host-by-email' command definition

define command{

command\_name notify-host-by-email

command\_line /usr/bin/printf "%b" "\*\*\*\*\* Nagios \*\*\*\*\*\n\nNotification Type: $NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress: $HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n" | /bin/mail -s "\*\* $NOTIFICATIONTYPE$ Host Alert: $HOSTNAME$ is $HOSTSTATE$ \*\*" $CONTACTEMAIL$

}

# 'notify-service-by-email' command definition

define command{

command\_name notify-service-by-email

command\_line /usr/bin/printf "%b" "\*\*\*\*\* Nagios \*\*\*\*\*\n\nNotification Type: $NOTIFICATIONTYPE$\n\nService: $SERVICEDESC$\nHost: $HOSTALIAS$\nAddress: $HOSTADDRESS$\nState: $SERVICESTATE$\n\nDate/Time: $LONGDATETIME$\n\nAdditional Info:\n\n$SERVICEOUTPUT$\n" | /bin/mail -s "\*\* $NOTIFICATIONTYPE$ Service Alert: $HOSTALIAS$/$SERVICEDESC$ is $SERVICESTATE$ \*\*" $CONTACTEMAIL$

}

* 配置联系人

# vi /usr/local/nagios/etc/objects/contacts.cfg

define contact {

contact\_name test1

alias system administrator

service\_notification\_period 24x7

host\_notification\_period 24x7

service\_notification\_options w,u,c,r

host\_notification\_options d,u,r

service\_notification\_commands notify-service-by-email

host\_notification\_commands notify-host-by-email

**email test1@test.com**

}

define contact {

contact\_name test2

alias test2

service\_notification\_period 24x7

host\_notification\_period 24x7

service\_notification\_options w,u,c,r

host\_notification\_options d,u,r

service\_notification\_commands notify-service-by-email

host\_notification\_commands notify-host-by-email

**email test2@test.com**

}

* 配置联系人组

# vi /usr/local/nagios/etc/objects/contactgroups.cfg

define contactgroup {

contactgroup\_name network

alias network administrator group

members test1

}

define contactgroup {

contactgroup\_name system

alias system administrator group

members test2

}

* 监控主机配置

# vi /usr/local/nagios/etc/objects/hosts.cfg

define host {

use host-pnp

host\_name nagios-test1

alias nagios test1

address 172.26.188.202

contact\_groups network,system

check\_command check-host-alive

max\_check\_attempts 5

notification\_interval 10

notification\_period 24x7

notification\_options d,u,r

}

* 监控服务配置

# vi /usr/local/nagios/etc/objects/services.cfg

define service {

use srv-pnp

host\_name nagios-test1

service\_description check\_tcp 80

check\_period 24x7

max\_check\_attempts 4

normal\_check\_interval 3

retry\_check\_interval 2

contact\_groups system,network

notification\_interval 10

notification\_period 24x7

notification\_options w,u,c,r

event\_handler\_enabled 1

event\_handler restart-httpd

check\_command check\_tcp!80

}

## 短信报警通知

短信报警可以通过139邮箱或qq邮箱绑定手机实现，设置相对比较简单，这里不做介绍。

本文采用飞信机器人实现nagios短信报警。

### 安装飞信机器人

下载飞信机器人及机器人支持库，下载地址：<http://www.it-adv.net/>

该软件为绿色版，解压后即可使用。

发送短信命令： fetion --mobile=13711123456 --pwd=mypwd --to=137xxxxxxxx --msg-gb=测试

**提示：**  
1. 使用复杂的密码（数字+字母+符号），将不会弹出图形验证码(我测试时是这种表现)。  
2. 第一次使用机器人时，即使是复杂密码，也可能会弹出图形验证码，以后将不再提示。  
3. 如果您是使用其它程序调用本程序，请在运行时，指定**--exit-on-verifycode=1，**否则，程序将无限期等待。当您以该参数运行机器人时，当弹出图形验证码时，程序将以退出码29结束( 在 linux 中，通过$?获取，在 windows中，通过 %ERRORLEVEL% 获取）  
4. 请确保目录权限可写。

### Nagios短信报警配置

* 编写发送脚本

#vi sendsms.sh

#!/bin/bash

fetionDir=/usr/local/fetion

cd $fetionDir

DIR=`pwd`

user=138\*\*\*\*\*\*\*\*

pwd=yourpassword

echo "================" >> msg.txt

echo "sms $1 $2" >> $DIR/msg.txt

echo "quit" >> $DIR/msg.txt

echo "================" >> msg.txt

$fetionDir/fetion --exit-on-verifycode=1 --mobile=$user --pwd=$pwd --to=$1 --msg-utf8="$2"

* 修改飞信目录权限（否则运行nagios的用户无法调用fetion发送短信）

#chown -R nagios.nagios /usr/local/fetion

#chmod –R 755 /usr/local/fetion/

* Nagios配置
* 定义command，增加如下内容。

# vi /usr/local/nagios/etc/objects/commands.cfg

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

#

# notify-host-by-sendmsg/notify-service-by-sendmsg //定义发送飞信报警的命令

#

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

define command{

command\_name notify-host-by-fetion

command\_line /usr/local/fetion/sendsms.sh $CONTACTPAGER$ "Host $HOSTSTATE$ alert for $HOSTNAME$($HOSTADDRESS$) on $TIME$."

}

define command{

command\_name notify-service-by-fetion

command\_line /usr/local/fetion/sendsms.sh $CONTACTPAGER$ "$TIME$:$SERVICEDESC$($HOSTADDRESS$) is $SERVICESTATE$."

}

* 定义contact，增加短信通知，如要修改如下内容

# vi /usr/local/nagios/etc/objects/contacts.cfg

define contact {

contact\_name test1

alias system administrator

service\_notification\_period 24x7

host\_notification\_period 24x7

service\_notification\_options w,u,c,r

host\_notification\_options d,u,r

**service\_notification\_commands notify-service-by-email,notify-service-by-fetion**

**host\_notification\_commands notify-host-by-email,notify-host-by-fetion**

email [test1@test.com](mailto:test1@test.com)

pager 138\*\*\*\*\*\*\*\*

}

### 故障排除

1、nagios无法发送短信报警。

**现象：**

(1)以web方式查看nagios监控状态，显示故障，却收不到报警短信。

(2)系统日志出现“Warning: Attempting to execute the command "/usr/local/fetion/fetion --mobile=138\*\*\*\*\*\*\*\* --pwd=little228 --to=138\*\*\*\*\*\*\*\* --msg-utf8="[linux1-CRITICAL] CHECK\_NRPE: Socket timeout after 10 seconds. (20:51:12)"" resulted in a return code of 127. Make sure the script or binary you are trying to execute actually exists...”。

**测试fetion：**

用手工方式执行飞信发短信没问题，再切换成用户nagios，也没问题。

**处理过程：**

（1）登陆该服务器，查看系统日志和nagios日志，如想象(2)。

（2）查看飞信所在的目录，检查权限设置。

（3）先修改一下他的/etc/sudoers文件，新增一行“nagios ALL=NOPASSWD:/usr/local/feiton/feiton”保存。

（4）切换到nagios用户：su - nagios

（5）执行sudo /usr/local/feiton/feiton ,报无lib\*.so文件。

（6）这些文件在/usr/local/feiton里呆着呢，先看看/lib里有没有这些。

（7）切换回root,查一下/lib目录，果然没有这些lib\*.so。

（8）把这些个lib\*.so拷贝到/lib目录。

（9）再切换到nagios用户执行sudo /usr/local/feiton/feiton 哈哈，没报错了。

（10）停止一个监控机上的服务，片刻，就发短信了。

对于监控短信报警，还是建议使用短信接口，不行该死的飞信，不能发其他运营商的手机短信。

## 邮件和短信分时报警

如我们需要每天上午让nagios通过邮件来报警，每天下午通过短信报警。

实现方法如下：定义2个timeperiod，如am、pm，定义contact时分邮件联系人和短信联系人，分别应用am、pm，最后在host、service中通知到相应的联系人。配置示例如下。

* 定义timeperiod，增加如下内容

define timeperiod{

timeperiod\_name am

alias am

sunday 00:00-12:00

monday 00:00-12:00

tuesday 00:00-12:00

wednesday 00:00-12:00

thursday 00:00-12:00

friday 00:00-12:00

saturday 00:00-12:00

}

define timeperiod{

timeperiod\_name pm

alias pm

sunday 12:00-24:00

monday 12:00-24:00

tuesday 12:00-24:00

wednesday 12:00-24:00

thursday 12:00-24:00

friday 12:00-24:00

saturday 12:00-24:00

}

* 定义contact，内容如下

define contact {

contact\_name test1 //定义上午接收邮件报警用户

alias system administrator

service\_notification\_period am

host\_notification\_period am

service\_notification\_options w,u,c,r

host\_notification\_options d,u,r

service\_notification\_commands notify-service-by-email

host\_notification\_commands notify-host-by-email

email test1@test.com

}

define contact {

contact\_name test2 //定义下午接收短信报警用户

alias test2

service\_notification\_period pm

host\_notification\_period pm

service\_notification\_options w,u,c,r

host\_notification\_options d,u,r

service\_notification\_commands notify-service-by-fetion

host\_notification\_commands notify-host-by-fetion

email test2@test.com

}

* 定义contactgroups，内容如下

define contactgroup {

contactgroup\_name network

alias network administrator group

members test1

}

define contactgroup {

contactgroup\_name system

alias system administrator group

members test2

}

* 定义host，内容如下

define host {

use host-pnp

host\_name nagios-test1

alias nagios test1

address 172.26.188.202

contact\_groups network,system

check\_command check-host-alive

max\_check\_attempts 5

notification\_interval 60

notification\_period 24x7

notification\_options d,u,r

}

define host {

use host-pnp

host\_name GIS01

alias GIS01

address 172.26.188.212

contact\_groups network,system

check\_command check-host-alive

max\_check\_attempts 5

notification\_interval 60

notification\_period 24x7

notification\_options d,u,r

}

* 定义service，内容如下

define service {

use srv-pnp

host\_name nagios-test1

service\_description check\_tcp 80

check\_period 24x7

max\_check\_attempts 4

normal\_check\_interval 3

retry\_check\_interval 2

contact\_groups system,network

notification\_interval 60

notification\_period 24x7

notification\_options w,u,c,r

event\_handler\_enabled 1

event\_handler restart-httpd

check\_command check\_tcp!80

}