第 4 章 入门

4.1. 给新手的建议

祝贺你选择了 Nagios！Nagios 是一个非常强大且柔性化的软件，但可能需要不少心血来学习如何配置使之符合你所需，一旦掌握了它如何工作并怎样来工作时，你会觉得再也离不开它！ :-) 对于初次使用 Nagios 的新手这有几个建议需要遵从：

- **放松点 － 这会花些时间。**不要指望它事情可以在转瞬间就搞掟，没有那么容易。设置好 Nagios 是一个费点事的工作，部分是由于对 Nagios 设置并不清楚，而还可能是由于并不清楚如何来监控现有网络（或者说如何监控会更好）。
- **使用快速上手指南。**本帮助给出了快速安装指南是给那些新手尽快地将 Nagios 安装到位并运行起来而写就的。在不到二十分钟之内可以安装并监控本地的系统，一旦完成了，就可以继续学习配置 Nagios 了。
- **阅读文档。**如果掌握 Nagios 运行机制，可以高效地配置它并且使之无所不能。确信已经阅读了这些文档(是"配置 Nagios"和"基本操作"两章)。在更好地理解基础性配置之前可以对那些高级内容暂时不管。
- **获得他人协助。**如果已经阅读文档并检测了样本配置文件但仍然有问题，写一个 EMail 给 **nagios-users** 邮件列表并写清楚问题。由于在这个项目上我有不少事情要做，直接给我的邮件我可能无法回复，所以最好是求助于邮件列表，如果有较好的背景并且可以将问题描述清楚，或许有人可以指出如何正确来做。更多地信息请在这个链接 http://www.nagios.org/support/下寻找。

4.2. 旧 Nagios 升级到当前版本

目录

4.2.1. 从旧的 3.x 版本升级到当前版本

如果是使用 3.x 的旧版，肯定是要尽快升级到当前版本。新版本修正了许多错误，下面假定已经根据快速安装指南的操作步骤从源代码包开始安装好 Nagios，下面可以安装更新的版本。虽然下面的操作都是用 root 操作的，但可以不用 root 权限也可以升级成功。下面是升级过程...

先确认已经备份好现有版本的 Nagios 软件和配置文件。如果升级过程中有不对的，至少可以回退到旧版本。

切换为Nagios用户。使用Debian/Ubuntu系统的可以用sudo -s nagios来切换。

```
su -l nagios
```

下载最新的Nagios安装包(http://www.nagios.org/download/)。

```
wget
http://osdn.dl.sourceforge.net/sourceforge/nagios/nagios-3.x.tar.gz
```

展开源码包。

```
tar xzf nagios-3.x.tar.gz
```

```
cd nagios-3.x
```

运行Nagios源程序的配置脚本，把加入外部命令的组名加上，象这样：

```
./configure --with-command-group=nagcmd
```

编译源程序

```
make all
```

安装升级后的二进制程序、文档和Web接口程序。在这步时旧配置文件还不会被覆盖。

```
make install
```

验证配置并重启动Nagios

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

```
/sbin/service nagios restart
```

好了，升级完成!

4.2.2. 从2.x升级到3.x

Nagios从2.x升级到3.x并不难。升级过程如同上面的旧版3.x的升级过程。但是Nagios3.x中有几处配置文件的改动需要注意：

1. The old **service_reaper_frequency** variable in the main config file has been renamed to check_result_reaper_frequency.

2. The old **$NOTIFICATIONNUMBER$** macro has been deprecated in favor of new $HOSTNOTIFICATIONNUMBER$ and $SERVICENOTIFICATIONNUMBER$ macros.
3. The old **parallelize** directive in service definitions is now deprecated and no longer used, as all service checks are run in parallel.
4. The old **aggregate_status_updates** option has been removed. All status file updates are now aggregated at a minimum interval of 1 second.
5. Extended host and extended service definitions have been deprecated. They are still read and processed by Nagios, but it is recommended that you move the directives found in these definitions to your host and service definitions, respectively.
6. The old **downtime_file** file variable in the main config file is no longer supported, as scheduled downtime entries are now saved in the retention file. To preserve existing downtime entries, stop Nagios 2.x and append the contents of your old downtime file to the retention file.
7. The old **comment_file** file variable in the main config file is no longer supported, as comments are now saved in the retention file. To preserve existing comments, stop Nagios 2.x and append the contents of your old comment file to the retention file.

Also make sure to read the "What's New" section of the documentation. It describes all the changes that were made to the Nagios 3 code since the latest stable release of Nagios 2.x. Quite a bit has changed, so make sure you read it over.

4.2.3. 从 RPM 包安装状态升级

如果当前是用 RPM 包安装的，或是用 Debian/Ubuntu 的 APT 软件包来安装 Nagios 的，需要用源程序包来安装升级，下面是操作步骤：

i. Main config file (usually **nagios.cfg**)
ii. Resource config file (usually **resource.cfg**)
iii. CGI config file (usually **cgi.cfg**)
iv. All your object definition files

a. Configuration files
b. Retention file (usually **retention.dat**)
c. Current Nagios log file (usually **nagios.log**)
d. Archived Nagios log files

1. Backup your existing Nagios installation
2. Uninstall the original RPM or APT package
3. Install Nagios from source by following the quickstart guide
4. Restore your original Nagios configuration files, retention file, and log files
5. Verify your configuration and start Nagios

注意 RPM 和 APT 包把 Nagios 的文件放置的位置有所不同。在升级前要确保那些配置文件备份好以在碰到解决不了的升级问题时可以回退到旧版本。

4.3. 快速安装指南

4.3.1. 介绍

这些指南试图让你在二十分钟内用简单地指令操作下从源程序安装 Nagios 并监控你的本地机器。这里并不讨论那些高级指令对于 95%以上的想起步的用户而言这是基础。

4.3.2. 指南

现在可以提供如下 Linux 发行版本上的快速安装指南：

- 基于 Fedora 平台的快速指南
- 基于 openSUSE 平台的快速指南
- 基于 Ubuntu 平台的快速指南

你可以在 NagiosCommunity.org 的维基百科上找到更多的安装上手指南。什么？找不到你所用的操作系统版本的指南？在维基百科上给其他人写一条吧！

如果你在一个上面没列出的操作系统或 Linux 发行包上安装 Nagios，请参照 Fedora 快速指南来概要地了解一下你需要做的事情。命令名、路径等可能因不同的发行包或操作系统而不同，因而这时你可能需要些努力来搞一下安装文档里的东西。

4.3.3. 安装后该做的

一旦你正确地安装并使 Nagios 运行起来后，毫无疑问你不仅要监控你的主机，你需要审视一下更多的文档来做更多的事情...

- 监控 Windows 主机
- 监控 Linux/Unix 主机
- 监控 Netware 服务器
- 监控路由器和交换机
- 监控网络打印机
- 监控公众服务平台

4.4. 基于 Fedora 平台的快速指南

4.4.1. 介绍

本指南试图让你通过简单的指令以在 20 分钟内在 Fedora 平台上通过对 Nagios 的源程序的安装来监控本地主机。这里没有讨论更高级的设置项 — 只是一些基本操作，但这足以使 95%的用户启动 Nagios。

这些指令在基于 Fedora Core 6 的系统下写成的。

最终结果是什么

如果按照本指南安装，最后将是这样结果：

1. Nagios 和插件将安装到/usr/local/nagios
2. Nagios 将被配置为监控本地系统的几个主要服务(CPU 负荷、磁盘利用率等)
3. Nagios 的 Web 接口是 URL 是 http://localhost/nagios/

4.4.2. 准备软件包

在做安装之前确认要对该机器拥有 root 权限。

确认你安装好的 Fedora 系统上已经安装如下软件包再继续。

1. Apache
2. GCC 编译器
3. GD 库与开发库

可以用 yum 命令来安装这些软件包，键入命令：

```
yum install httpd

yum install gcc

yum install glibc glibc-common

yum install gd gd-devel
```

4.4.3. 操作过程

1)建立一个帐号

切换为 root 用户

```
su -l
```

创建一个名为 nagios 的帐号并给定登录口令

/usr/sbin/useradd nagios

passwd nagios

创建一个用户组名为 nagcmd 用于从 Web 接口执行外部命令。将 nagios 用户和 apache 用户都加到这个组中。

/usr/sbin/groupadd nagcmd

/usr/sbin/usermod -G nagcmd nagios

/usr/sbin/usermod -G nagcmd apache

2)下载 Nagios 和插件程序包

建立一个目录用以存储下载文件

mkdir ~/downloads

cd ~/downloads

下载 Nagios 和 Nagios 插件的软件包(访问 http://www.nagios.org/download/ 站点以获得最新版本)，在写本文档时，最新的 Nagios 的软件版本是 3.0rc1，Nagios 插件的版本是 1.4.11。

wget
http://osdn.dl.sourceforge.net/sourceforge/nagios/nagios-3.0rc1.tar.gz

wget
http://osdn.dl.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.11.tar.gz

3)编译与安装 Nagios

展开 Nagios 源程序包

cd ~/downloads

tar xzf nagios-3.0rc1.tar.gz

cd nagios-3.0rc1

运行 Nagios 配置脚本并使用先前开设的用户及用户组：

./configure --with-command-group=nagcmd

编译 Nagios 程序包源码

make all

安装二进制运行程序、初始化脚本、配置文件样本并设置运行目录权限

make install

make install-init

make install-config

make install-commandmode

现在还不能启动 Nagios－还有一些要做的...

4)客户化配置

样例配置文件默认安装在这个目录下/usr/local/nagios/etc，这些样例文件可以配置 Nagios 使之正常运行，只需要做一个简单的修改...

用你擅长的编辑器软件来编辑这个 /usr/local/nagios/etc/objects/contacts.cfg 配置文件，更改 email 地址 nagiosadmin 的联系人定义信息中的 EMail 信息为你的 EMail 信息以接收报警内容。

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

5)配置 WEB 接口

安装 Nagios 的 WEB 配置文件到 Apache 的 conf.d 目录下

make install-webconf

创建一个 nagiosadmin 的用户用于 Nagios 的 WEB 接口登录。记下你所设置的登录口令，一会儿你会用到它。

htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

重启 Apache 服务以使设置生效。

service httpd restart

6)编译并安装 Nagios 插件

展开 Nagios 插件的源程序包

```
cd ~/downloads

tar xzf nagios-plugins-1.4.11.tar.gz

cd nagios-plugins-1.4.11
```

编译并安装插件

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

make

make install
```

7)启动 Nagios

把 Nagios 加入到服务列表中以使之在系统启动时自动启动

```
chkconfig --add nagios

chkconfig nagios on
```

验证 Nagios 的样例配置文件

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

如果没有报错，可以启动 Nagios 服务

```
service nagios start
```

8)更改 SELinux 设置

Fedora 与 SELinux(安全增强型 Linux)同步发行与安装后将默认使用强制模式。这会在你尝试联入 Nagios 的 CGI 时导致一个"内部服务错误"消息。

如果是 SELinux 处于强制安全模式时需要做

```
getenforce
```

令 SELinux 处于容许模式

```
setenforce 0
```

如果要永久性更变它，需要更改/etc/selinux/config 里的设置并重启系统。

不关闭 SELinux 或是永久性变更它的方法是让 CGI 模块在 SELinux 下指定强制目标模式：

```
chcon -R -t httpd_sys_content_t /usr/local/nagios/sbin/

chcon -R -t httpd_sys_content_t /usr/local/nagios/share/
```

更多有关 Nagios 的 CGI 模块增加目标策略的强制权限方式见 NagiosCommunity.org 的维基百科 http://www.nagioscommunity.org/wiki。

9)登录 WEB 接口

你现在可以从 WEB 方式来接入 Nagios 的 WEB 接口了，你需要在提示下输入你的用户名(**nagiosadmin**)和口令，你刚刚设置的，这里用系统默认安装的浏览器，用下面这个超链接

```
http://localhost/nagios/
```

点击“服务详情”的引导超链来查看你本机的监视详情。你可能需要给点时间让 Nagios 来检测你机器上所依赖的服务因为检测需要些时间。

10)其他的变更

确信你机器的防火墙规则配置允许你可以从远程登录到 Nagios 的 WEB 服务。

配置 EMail 的报警项超出了本文档的内容，指向你的系统档案用网页查找或是到这个站点 NagiosCommunity.org wiki 来查找更进一步的信息，以使你的系统上可以向外部地址发送 EMail 信息。更多有关通知的信息可以查阅这篇文档。

11)完成了

祝贺你已经成功安装好 Nagios，但网络监控工作只是刚开始。毫无疑问你不是只监控本地系统，所以要看以下这些文档...

1. 对 Windows 主机的监控
2. 对 Linux/Unix 主机的监控
3. 对 Netware 服务器的监控
4. 监控路由器和交换机
5. 监控公众化服务(HTTP、FTP、SSH 等)

4.5. 基于 openSUSE 平台的快速指南

4.5.1. 介绍

本指南试图让你通过简单的指令以在 20 分钟内在你的 openSUSE 平台上通过对 Nagios 的源程序的安装来监控本地主机。这里没有讨论更高级的设置项 — 只是一些基本操作，但这足以使 95%的用户启动 Nagios。

这些指令在基于 openSUSE10.2 的系统下写成的。

4.5.2. 所需的软件包

确认你安装好的 openSUSE 系统之上已经安装了如下软件包再继续。你可以在 openSUSE 系统下用 **yast** 来安装软件包。

- apache2
- C/C++开发库

4.5.3. 操作过程

1）建立一个帐号

切换为 root 用户

```
su -l
```

创建新帐户名为 **nagios** 并给它一个登录口令

```
/usr/sbin/useradd nagios

passwd nagios
```

创建一个用户组名为 **nagios**，并把 nagios 帐户加入该组

```
/usr/sbin/groupadd nagios

/usr/sbin/usermod -G nagios nagios
```

创建一个用户组名为 **nagcmd** 来执行外部命令并可以通过 WEB 接口来执行。将 nagios 用户和 apache 用户都加到这个组中。

```
/usr/sbin/groupadd nagcmd

/usr/sbin/usermod -G nagcmd nagios

/usr/sbin/usermod -G nagcmd wwwrun
```

2）下载 Nagios 和插件程序包

建立一个目录用以存储下载文件

```
mkdir ~/downloads

cd ~/downloads
```

下载 Nagios 和 Nagios 插件的软件包（访问 http://www.nagios.org/download/ 站点以获得最新版本），在写本文档时，最新的 Nagios 的软件版本是 3.0rc1，Nagios 插件的版本是 1.4.11。

```
wget
http://osdn.dl.sourceforge.net/sourceforge/nagios/nagios-3.0rc1.tar.gz

wget
http://osdn.dl.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.11.tar.gz
```

3）编译与安装 Nagios

展开 Nagios 源程序包

```
cd ~/downloads

tar xzf nagios-3.0rc1.tar.gz

cd nagios-3.0rc1
```

运行 Nagios 配置脚本并使用先前开设的用户及用户组：

```
./configure --with-command-group=nagcmd
```

编译 Nagios 程序包源码

```
make all
```

安装二进制运行程序、初始化脚本、配置文件样本并设置运行目录权限

```
make install

make install-init

make install-config

make install-commandmode
```

现在还不能启动 Nagios — 还有一些要做的...

4）客户化配置

样例配置文件默认安装在这个目录下/usr/local/nagios/etc，这些样例文件可以配置 Nagios 使之正常运行，只需要做一个简单的修改...

用你擅长的编辑器软件来编辑这个 /usr/local/nagios/etc/objects/contacts.cfg 配置文件，更改 email 地址 nagiosadmin 的联系人定义信息中的 EMail 信息为你的 EMail 信息以接收报警内容。

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

5）配置 WEB 接口

安装 Nagios 的 WEB 配置文件到 Apache 的 conf.d 目录下

```
make install-webconf
```

创建一个 nagiosadmin 的用户用于 Nagios 的 WEB 接口登录。记下你所设置的登录口令，一会儿你会用到它。

```
htpasswd2 -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

重启 Apache 服务以使设置生效。

```
service apache2 restart
```

6）编译并安装 Nagios 插件

展开 Nagios 插件的源程序包

```
cd ~/downloads

tar xzf nagios-plugins-1.4.11.tar.gz

cd nagios-plugins-1.4.11
```

编译并安装插件

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

make

make install
```

7）启动 Nagios

把 Nagios 加入到服务列表中以使之在系统启动时自动启动

```
chkconfig --add nagios
```

```
chkconfig nagios on
```

验证 Nagios 的样例配置文件

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

如果没有报错，可以启动 Nagios 服务

```
service nagios start
```

8）登录 WEB 接口

你现在可以从 WEB 方式来接入 Nagios 的 WEB 接口了，你需要在提示下输入你的用户名（**nagiosadmin**）和口令，你刚刚设置的，这里用系统默认安装的浏览器，用下面这个超链接

```
konqueror http://localhost/nagios/
```

点击“服务详情”的引导超链来查看你本机的监视详情。你可能需要给点时间让 Nagios 来检测你机器上所依赖的服务因为检测需要些时间。

9）其他的变更

确信你机器的防火墙规则配置允许你可以从远程登录到 Nagios 的 WEB 服务。

你可以这样做：

1. 打开控制中心
2. 选择'打开超户设置'以打开 YaST 超户控制中心
3. 选择在'安全与用户'设置里的'防火墙'
4. 在防火墙的配置窗口中点击'允许的服务'选项
5. 在许可的服务中增加'HTTP 服务'，是'外部区'的部分
6. 点击'下一步'并选择'接受'以使得防火墙设置生效

配置 EMail 的报警项超出了本文档的内容，指向你的系统档案用网页查找或是到这个站点 NagiosCommunity.org wiki 来查找更进一步的信息，以使你的 openSUSE 系统上可以向外部地址发送 EMail 信息。

4.6. 基于 Ubuntu 平台的快速指南

4.6.1. 介绍

本指南试图让你通过简单的指令以在 20 分钟内在 Ubuntu 平台上通过对 Nagios 的源程序的安装来监控本地主机。没有讨论更高级的设置项－只是一些基本操作，但这足以使 95%的用户启动 Nagios。

这些指令在基于 Ubuntu6.10(桌面版)的系统下写成的。

What You'll End Up With

如果按照本指南安装，最后将是这样结果：

1. Nagios 和插件将安装到/usr/local/nagios
2. Nagios 将被配置为监控本地系统的几个主要服务(CPU 负荷、磁盘利用率等)
3. Nagios 的 Web 接口是 URL 是 http://localhost/nagios/

4.6.2. 所需软件包

确认你安装好的系统上已经安装如下软件包再继续。

1. Apache2
2. GCC 编译器与开发库
3. GD 库与开发库

可以用 apt-get 命令来安装这些软件包，键入命令：

```
sudo apt-get install apache2

sudo apt-get install build-essential

sudo apt-get install libgd2-dev
```

4.6.3. 操作过程

1)建立一个帐号

切换为 root 用户

```
sudo -s
```

创建一个名为 nagios 的帐号并给定登录口令

```
/usr/sbin/useradd nagios
```

passwd nagios

在 Ubuntu 服务器版(6.01 或更高版本)，创建一个用户组名为 **nagios**(默认是不创建的)。在 Ubuntu 桌面版上要跳过这一步。

/usr/sbin/groupadd nagios

/usr/sbin/usermod -G nagios nagios

创建一个用户组名为 **nagcmd** 用于从 Web 接口执行外部命令。将 nagios 用户和 apache 用户都加到这个组中。

/usr/sbin/groupadd nagcmd

/usr/sbin/usermod -G nagcmd nagios

/usr/sbin/usermod -G nagcmd www-data

2)下载 Nagios 和插件程序包

建立一个目录用以存储下载文件

mkdir ~/downloads

cd ~/downloads

下载 Nagios 和 Nagios 插件的软件包（访问 http://www.nagios.org/download/ 站点以获得最新版本），在写本文档时，最新的 Nagios 的软件版本是 3.0rc1，Nagios 插件的版本是 1.4.11。

wget
http://osdn.dl.sourceforge.net/sourceforge/nagios/nagios-3.0rc1.tar.gz

wget
http://osdn.dl.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.11.tar.gz

3)编译与安装 Nagios

展开 Nagios 源程序包

cd ~/downloads

tar xzf nagios-3.0rc1.tar.gz

cd nagios-3.0rc1

运行 Nagios 配置脚本并使用先前开设的用户及用户组：

./configure --with-command-group=nagcmd

编译 Nagios 程序包源码

make all

安装二进制运行程序、初始化脚本、配置文件样本并设置运行目录权限

make install

make install-init

make install-config

make install-commandmode

现在还不能启动 Nagios－还有一些要做的...

4)客户化配置

样例配置文件默认安装在这个目录下/usr/local/nagios/etc，这些样例文件可以配置 Nagios 使之正常运行，只需要做一个简单的修改...

用你擅长的编辑器软件来编辑这个 /usr/local/nagios/etc/objects/contacts.cfg 配置文件，更改 email 地址 nagiosadmin 的联系人定义信息中的 EMail 信息为你的 EMail 信息以接收报警内容。

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

5)配置 WEB 接口

安装 Nagios 的 WEB 配置文件到 Apache 的 conf.d 目录下

make install-webconf

创建一个 nagiosadmin 的用户用于 Nagios 的 WEB 接口登录。记下你所设置的登录口令，一会儿你会用到它。

htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

重启 Apache 服务以使设置生效。

/etc/init.d/apache2 reload

6)编译并安装 Nagios 插件

展开 Nagios 插件的源程序包

cd ~/downloads

tar xzf nagios-plugins-1.4.11.tar.gz

cd nagios-plugins-1.4.11

编译并安装插件

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

make

make install

7)启动 Nagios

把 Nagios 加入到服务列表中以使之在系统启动时自动启动

ln -s /etc/init.d/nagios /etc/rcS.d/S99nagios

验证 Nagios 的样例配置文件

/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

如果没有报错，可以启动 Nagios 服务

/etc/init.d/nagios start

8)登录 WEB 接口

你现在可以从 WEB 方式来接入 Nagios 的 WEB 接口了，你需要在提示下输入你的用户名(**nagiosadmin**)和口令，你刚刚设置的，这里用系统默认安装的浏览器，用下面这个超链接

http://localhost/nagios/

点击“服务详情”的引导超链来查看你本机的监视详情。你可能需要给点时间让 Nagios 来检测你机器上所依赖的服务因为检测需要些时间。

9)其他的变更

如果要接收 Nagios 的 EMail 警报，需要安装(Postfix)包

```
sudo apt-get install mailx
```

需要编辑 Nagios 里的 EMail 通知送出命令，它位于 **/usr/local/nagios/etc/commands.cfg** 文件中，将里面的'/bin/mail'全部替换为'/usr/bin/mail'。一旦设置好需要重启动 Nagios 以使配置生效。

```
sudo /etc/init.d/nagios restart
```

配置 EMail 的报警项超出了本文档的内容，指向你的系统档案用网页查找或是到这个站点 NagiosCommunity.org wiki 来查找更进一步的信息，以使 Ubuntu 系统上可以向外部地址发送 EMail 信息。

4.7. 监控 Windows 主机

4.7.1. 介绍

本文用来说明如何监控 Windows 主机的本地服务和特性，包括：

1. 内存占用率
2. CPU 负载
3. Disk 利用率
4. 服务状态
5. 运行进程
6. 等等

在 Windows 主机上的公众化服务(如 HTTP、FTP、POP3 等)可以查阅监控公众化服务这篇文档。

注意

如下的内容是假定你已经按照快速安装指南安装好了 Nagios 系统之后做的，下面所使用的样例配置文件(如 **commands.cfg**、**templates.cfg** 等)已经在安装过程中安装到位。

4.7.2. 概览

对 Windows 机器的监控私有服务需要在机器上安装代理程序。代理将会在检测插件与 Nagios 服务之间起网关代理作用。如果没有在机器上安装代理的话，Nagios 将无法对 Windows 私有服务或属性等进行监控。

在下面例子中，将在 Windows 机器上安装 NSClient++外部构件并使用 check_nt 插件检测和与 NSClient++构件进行通讯。如果你按照指南来安装的话，check_nt 插件已经安装到了 Nagios 服务器上。

如果愿意，可以用其他的 Windows 代理(象 NC_Net)替代 NSClient++构件所起的作用－只是要稍稍改一下对应的命令和服务定义等。下面将只是讨论安装了 NSClient++外部构件的情况。

4.7.3. 步骤

为完成对 Windows 机器的检测，有几个步骤要做，它们是：

1. 确认一下首要条件；
2. 在 Windows 机器上安装代理(在本例中是安装 NSClient++构件)；
3. 给 Windows 机器创建新的主机和服务对象定义；
4. 重启动 Nagios 守护进程。

4.7.4. 已经做了什么？

为使过程简单，已经完成了少量配置文件的工作：

1. 已经把 **check_nt** 命令加入到了 **commands.cfg** 文件中，就可以直接使用 **check_nt** 插件来监控 Windows 服务；
2. 一个 Windows 机器的主机对象模板(命名为 **windows-server**)已经在 **templates.cfg** 文件里创建好了，可以更容易地加入一个新的 Windows 主机对象定义。

常用的配置文件可以在/usr/local/nagios/etc/objects/目录里找到。如果愿意可以对里面的对象进行修改以适应你的要求。但是，如果你没有熟悉配置 Nagios 之前劝你不要这么做。开始时可以只是按照下面的指令操作来快速完成监控 Windows 机器。

4.7.5. 首要条件

首次监控一台 Winodws 机器时需要对 Nagios 做点额外的工作，记住，仅仅是监控*第一台* Windows 机器时需要做这些工作。

编辑 Nagios 的主配置文件

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

把下面这行最前面的#号去掉：

```
#cfg_file=/usr/local/nagios/etc/objects/windows.cfg
```

保存配置文件并退出。

刚才做的是什么呢？是让 Nagios 起用 /usr/local/nagios/etc/objects/windows.cfg 这个配置文件里的对象定义。在这个配置文件里可以加些 Windows 的主机与服务对象定义。该配置文件里已经包含有几个样例主机、主机组及服务对象定义。对于第一台 Windows 机器，可以只是简单地修改里面已经有的主机与服务对象定义而不要新创建一个。

4.7.6. 安装 Windows 代理程序

在用 Nagios 监控 Windows 机器的私有服务之前，需要先在机器上安装代理程序。推荐使用 NSClient++外部构件，它可以在 http://sourceforge.net/projects/nscplus 找到。如下指令可以安装一个基本的 NSClient++外部构件，同时也配置好 Nagios 来监控这台 Windows 机器。

1. 从 http://sourceforge.net/projects/nscplus 站点下载最新稳定版的 NSClient++软件包；

2. 展开软件包到一个目录下，如 C:\NSClient++；

3. 打开一个命令行窗口并切换到 C:\NSClient++目录下；

4. 用下面命令将 NSClient++系统服务注册到系统里：

```
nsclient++ /install
```

5. 用下面命令安装 NSClient++系统托盘程序('SysTray'是大小写敏感的)：

```
nsclient++ SysTray
```

6. 打开服务管理器并确认 NSClientpp 服务可以在桌面交互(看一下服务管理器里的'Log On'选项页)，如果没有允许桌面交互，点一下里面的选择项打开它。

7. 编辑 NSC.INI 文件(位于 C:\NSClient++目录)并做如下修改：

1. 去掉在[modules]段里的列出模块程序的注释，除了 CheckWMI.dll 和 RemoteConfiguration.dll；
2. 最好是修改一下在[Settings]段里的'password'选项；
3. 去掉在[Settings]段里的'allowed_hosts'选项注释，把 Nagios 服务所在主机的 IP 加到这一行里，或是置为空，让全部主机都可以联入；
4. 确认一下在[NSClient]段里的'port'选项里已经去掉注释并设置成'12489'(默认端口)；

8. 用下面命令启动 NSClient++服务：

```
nsclient++ /start
```

9. 如果安装正确，一个新的图标会出现在系统托盘里，是个黄圈里面有个黑色的'M'；

10. 完成了。这台 Windows 机器可以加到 Nagios 监控配置里了...

4.7.7. 配置 Nagios

为监控 Windows 机器下面要在 Nagios 配置文件里加几个对象定义。

编辑方式打开 **windows.cfg** 文件。

```
vi /usr/local/nagios/etc/objects/windows.cfg
```

给 Windows 机器加一个新的主机对象定义以便监控。如果是被监控的第一台 Windows 机器，可以只是修改 **windows.cfg** 文件里的对象定义。修改 **host_name**、**alias** 和 **address** 域以符合那台 Windows 机器。

```
define host{

        use             windows-server ; Inherit default values from a
Windows server template (make sure you keep this line!)

        host_name               winserver

        alias           My Windows Server

        address         192.168.1.2

        }
```

好了。下面可以加几个服务定义(在同一个配置文件里)以使 Nagios 监控 Windows 机器上的不同属性内容。如果是第一台 Windows 机器，可以只是修改 **windows.cfg** 里的服务对象定义。

注意

用你刚刚加好的主机对象定义里的 **host_name** 来替换例子里的"**winserver**"。

加入下面的服务定义以监控运行于 Windows 机器上的 NSClient++外部构件的版本。当到时间要升级 Windows 机器上的外部构件时这信息会很用有，因为它可以告知这台 Windows 机器上的 NSClient++需要升级到最新版本。

```
define service{

        use                     generic-service

        host_name                       winserver

        service_description     NSClient++ Version

        check_command           check_nt!CLIENTVERSION

        }
```

加入下面的服务定义以监控 Windows 机器的启动后运行时间。

```
define service{
        use                     generic-service
        host_name                       winserver
        service_description     Uptime
        check_command           check_nt!UPTIME
        }
```

加入下面的服务定义可监控 Windows 机器的 CPU 利用率，并在 5 分钟 CPU 负荷高于 90%时给出一个紧急警报或是高于 80%时给出一个告警警报。

```
define service{
        use                     generic-service
        host_name                       winserver
        service_description     CPU Load
        check_command           check_nt!CPULOAD!-l 5,80,90
        }
```

加入下面的服务定义可监控 Windows 机器的内存占用率，并在 5 分钟内存占用率高于 90%时给出一个紧急警报或是高于 80%时给出一个告警警报。

```
define service{
        use                     generic-service
        host_name                       winserver
        service_description     Memory Usage
        check_command           check_nt!MEMUSE!-w 80 -c 90
        }
```

加入下面的服务定义可监控 Windows 机器的 C:盘的磁盘利用率，并在磁盘利用率高于 90%时给出一个紧急警报或是高于 80%时给出一个告警警报。

```
define service{
```

```
        use                     generic-service

        host_name                       winserver

        service_description     C:\ Drive Space

        check_command           check_nt!USEDDISKSPACE!-l c -w 80 -c 90

        }
```

加入下面的服务定义可监控 Windows 机器上的 W3SVC 服务状态，并在 W3SVC 服务停止时给出一个紧急警报。

```
define service{

        use                     generic-service

        host_name                       winserver

        service_description     W3SVC

        check_command           check_nt!SERVICESTATE!-d SHOWALL -l
W3SVC

        }
```

加入下面的服务定义可监控 Windows 机器上的 Explorer.exe 进程，并在进程没有运行时给出一个紧急警报。

```
define service{

        use                     generic-service

        host_name                       winserver

        service_description     Explorer

        check_command           check_nt!PROCSTATE!-d SHOWALL -l
Explorer.exe

        }
```

都好了，已经加好了基础服务定义，可以监控 Windows 机器了，保存一下配置文件。

4.7.8. 口令保护

如果想指定保存在 Windows 机器上 NSClient++配置文件里的口令，可以修改 **check_nt** 命令定义，让它带着口令。编辑方式打开 **commands.cfg** 文件。

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

修改 **check_nt** 命令的定义，带上"-s <PASSWORD>"命令参数(这里的 PASSWORD 要换成你 Windows 机器的真正口令)，象这样：

```
define command{

        command_name    check_nt

        command_line    $USER1$/check_nt -H $HOSTADDRESS$ -p 12489 -s
PASSWORD -v $ARG1$ $ARG2$

        }
```

保存文件退出。

4.7.9. 重启动 Nagios

如果修改好 Nagios 配置文件，需要验证你的配置文件并重启动 Nagios。

如果验证配置文件过程中有什么错误信息，在做下一步前一定要修正好配置文件。一定要保证验证过程中不再有出错信息后再启动或重启动 Nagios！

4.8. 监控 Linux/Unix 主机

4.8.1. 介绍

本文档描述了如果监控 Linux/UNIX 的"私有"服务和属性，如：

1. CPU 负荷
2. 内存占用率
3. 磁盘利用率
4. 登录用户
5. 运行进程
6. 等

由 Linux 系统上的公众服务(HTTP、FTP、SSH、SMTP 等)可以按照这篇监控公众服务文档。

注意

如下内容是假定已经按照快速安装指南安装并设置好 Nagios。如下例子参考了样例配置文件(**commands.cfg**、**templates.cfg** 等)里的对象定义，样例配置文件已经在安装过程中安装就位。

4.8.2. 概览

[注意：本文档没有结束。推荐阅读文档 NRPE 外部构件里如何监控远程 Linux/Unix 服务器中的指令]

有几种不同方式来监控远程 Linux/UNIX 服务器的服务与属性。一个是应用共享式 SSH 密钥运行 **check_by_ssh** 插件来执行对远程主机的检测。这种方法本文档不讨论，但它会导致安装有 Nagios 的监控服务器很高的系统负荷，尤其是你要监控成百个主机中的上千个服务时，这是因为要建立/毁构 SSH 联接的总开销很高。

另一种方法是使用 NRPE 外部构件监控远程主机。NRPE 外部构件可以在远程的 Linux/Unix 主机上执行插件程序。如果是要象监控本地主机一样对远程主机的磁盘利用率、CPU 负荷和内存占用率等情况下，NRPE 外部构件非常有用。

4.9. 监控路由器和交换机

4.9.1. 介绍

本文档将介绍如何来监控路由器和交换机的状态。一些便宜的"无网管"功能的交换机与集线器不能配置 IP 地址而且对于网络是不可见的组成构件，因而没办法来监控这种东西。稍贵些的交换机和路由器可以配置 IP 地址可以用 PING 检测或是通过 SNMP 来查询状态信息。

下面将描述如果来监控这些有网管功能的交换机、集线器和路由器：

1. 包丢弃率，平均回包周期 RTA
2. SNMP 状态信息
3. 带宽与流量

注意

如下指令是假定你已经按快速安装指南安装好 Nagios。参考的样例配置是在已经按指南安装就位的配置文件(**commands.cfg**、**templates.cfg** 等)。

4.9.2. 概览

监控交换机与路由器可简可繁－主要是看拥有什么样设备与想监控什么内容。做为极为重要的网络组成构件，毫无疑问至少要监控一些基本状态。

交换机与路由器可以简单地用 PING 来监控丢包率、RTA 等数据。如果交换机支持 SNMP，就可以监控端口状态等，用 check_snmp 插件，也可以监控带宽(如果用了 MRTG)，用 check_mrtgtraf 插件。

check_snmp 插件只有当系统里安装了 net-snmp 和 net-snmp-utils 包后才编译。先确定插件已经在/usr/local/nagios/libexec 目录里再继续做，如果没有这个文件，安装 net-snmp 和 net-snmp-utils 包并且重编译并重新安装 Nagios 插件包。

4.9.3. 步骤

要监控交换机与路由器要有几步工作：

1. 第一时间执行些必备工作；
2. 给设备创建要监控的主机与服务对象定义；
3. 重启动 Nagios 守护进程。

4.9.4. 已经做了什么？

为了让工作轻松点，几个配置任务已经做好了：

1. 两个命令定义(**check_snmp** 和 **check_local_mrtgtraf**)已经加到了 **commands.cfg** 文件中。可以用 **check_snmp** 和 **check_mrtgtraf** 插件来监控网络打印机。
2. 一个交换机模板(命名为 **generic-switch**)已经创建在 **templates.cfg** 文件里。可以在对象定义里更容易地加一个新的交换机与路由器设备。

以上的监控配置文件可以在/usr/local/nagios/etc/objects/目录里找到。如果愿意可以修改这些定义或是加入其他适合需要的更好的定义。但推荐你最好是等到你熟练地掌握了 Nagios 配置之后再这么做。开始的时候，只要按上述的配置来监控网络里的路由器和交换机就可以了。

4.9.5. 必备工作

要配置 Nagios 用于监控网络里的交换机之前，有必要做点额外工作。记住，这是*首先*要做的工作才能监控。

编辑 Nagios 的主配置文件

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

移除文件里下面这行的最前面的(#)符号

```
#cfg_file=/usr/local/nagios/etc/objects/switch.cfg
```

保存文件并退出。

为何要这么做？这是要让 Nagios 检查 **/usr/local/nagios/etc/objects/switch.cfg** 配置文件来找些额外的对象定义。在文件里可以增加有关路由器和交换机设备的主机与服务定义。配置文件已经包含了几个样本主机、主机组和服务定义。做为监控路由器与交换机的*第一步*工作是最好在样例的主机与服务对象定义之上修改而不是重建一个。

4.9.6. 配置 Nagios

需要做些对象定义以监控新的交换机与路由器设备。

打开 **switch.cfg** 文件进行编辑。

```
vi /usr/local/nagios/etc/objects/switch.cfg
```

给要监控的交换机加一个新的主机对象定义。如果这是*第一台*要监控的交换机设备，可以简单地修改 **switch.cfg** 里的样例配置。修改主机对象里的 **host_name**、**alias** 和 **address** 域值来适用于监控。

```
define host{

        use             generic-switch          ; Inherit default values
from a template

        host_name       linksys-srw224p         ; The name we're giving
to this switch

        alias           Linksys SRW224P Switch ; A longer name
associated with the switch

        address         192.168.1.253           ; IP address of the
switch
```

```
        hostgroups      allhosts,switches       ; Host groups this switch
is associated with

}
```

4.9.7. 监控服务

现在可以加些针对监控交换机的服务对象定义(在同一个配置文件)。如果是*第一台*要监控的交换机设备，可以简单地修改 **switch.cfg** 里的样例配置。

注意

替换样例定义里的"**linksys-srw224p**"主机名为你刚才定义的名字，是修改在 **host_name** 域。

4.9.8. 监控丢包率和 RTA

增加如下的服务定义以监控自 Nagios 监控主机到交换机的丢包率和平均回包周期 RTA，在一般情况下每 5 分钟检测一次。

```
define service{

        use                     generic-service ; Inherit values from a
template

        host_name               linksys-srw224p ; The name of the host the
service is associated with

        service_description     PING            ; The service
description

        check_command           check_ping!200.0,20%!600.0,60% ; The
command used to monitor the service

        normal_check_interval   5       ; Check the service every 5
minutes under normal conditions

        retry_check_interval    1       ; Re-check the service every
minute until its final/hard state is determined

}
```

这个服务的状态将会处于：

1. 紧急(CRITICAL)－条件是 RTA 大于 600ms 或丢包率大于等于 60%；
2. 告警(WARNING)－条件是 RTA 大于 200ms 或是丢包率大于等于 20%；
3. 正常(OK)－条件是 RTA 小于 200ms 或丢包率小于 20%

4.9.9. 监控 SNMP 状态信息

如果交换机与路由器支持 SNMP 接口，可以用 **check_snmp** 插件来监控更丰富的信息。如果不支持 SNMP，跳过此节。

加入如下服务定义到你刚才修改的交换机对象定义之中

```
define service{

        use                     generic-service; Inherit values from a
template

        host_name               linksys-srw224p

        service_description     Uptime

        check_command           check_snmp!-C public -o sysUpTime.0

}
```

在上述服务定义中的 **check_command** 域里，用"-C public"来指定 SNMP 共同体名称为"public"，用"-o sysUpTime.0"指明要检测的 OID(译者注－MIB 节点值)。

如果要确保交换机上某个指定端口或接口的状态处于运行状态，可以在对象定义里加入一段定义：

```
define service{

        use                     generic-service; Inherit values from a
template

        host_name               linksys-srw224p

        service_description     Port 1 Link Status

        check_command           check_snmp!-C public -o ifOperStatus.1
-r 1 -m RFC1213-MIB

}
```

在上例中，"-o ifOperStatus.1"指出取出交换机的端口编号为 1 的 OID 状态。"-r 1"选项是让 check_snmp 插件检查返回一个正常(OK)状态，如果是在 SNMP 查询结果中存在"1"(1 说明交换机端口处于运行状态)如果没找到 1 就是紧急(CRITICAL)状态。"-m RFC1213-MIB"是可选的，它告诉 check_snmp 插件只加载"RFC1213-MIB"库而不是加载每个在系统里的 MIB 库，这可以加快插件运行速度。

这就是给 SNMP 库的例子。有成百上千种信息可以通过 SNMP 来监控，这完全取决于你需要做什么和如果来做监控。祝你好运！

提示

通常可以用如下命令来寻找你想用于监控的 OID 节点(用你的交换机 IP 替换 **192.168.1.253**)：

snmpwalk -v1 -c public 192.168.1.253 -m ALL .1

4.9.10. 监控带宽和流量

可以监控交换机或路由器的带宽利用率，用 MRTG 绘图并让 Nagios 在流量超出指定门限时报警。check_mrtgtraf 插件(它已经包含在 Nagios 插件软件发行包中)可以实现。

需要让 check_mrtgtraf 插件知道如何来保存 MRTG 数据并存入文件，以及门限等。在例子中，监控了一个 Linksys 交换机。MRTG 日志保存于 /var/lib/mrtg/192.168.1.253_1.log 文件中。这就是我用于监控的服务定义，它可以用于监控带宽数据到日志文件之中...

```
define service{

        use                     generic-service; Inherit values from a
template

        host_name               linksys-srw224p

        service_description     Port 1 Bandwidth Usage

        check_command
        check_local_mrtgtraf!/var/lib/mrtg/192.168.1.253_1.log!AVG!10
00000,2000000!5000000,5000000!10

}
```

在上例中，"/var/lib/mrtg/192.168.1.253_1.log"参数传给 check_local_mrtgtraf 命 令意思是插件的 MRTG 日志文件在这个文件里读写，

"AVG"参数的意思是取带宽的统计平均值，"1000000,200000"参数是指流入的告警门限（以字节为单位），"5000000,5000000"是输出流量紧急状态门限(以字节为单位)，"10"是指如果 MRTG 日志如果超过 10 分钟没有数据 返回一个紧急状态(应该每 5 分钟更新一次)。

保存该配置文件

4.9.11. 重启动 Nagios

一旦给 switch.cfg 文件里加好新的主机与服务对象定义，就可以开始对路由器与交换机进行监控。为了开始监控，需要先验证配置文件再重新启动 Nagios。

如果验证过程有有任何错误信息，修改配置文件再继续。一定要保证配置验证过程中没有错误信息再启动 Nagios!

4.10. 监控网络打印机

4.10.1. 介绍

本文件描述了如何监控网络打印机。特别是有内置或外置 JetDirect 卡的 HP 惠普打印机设备，或是其他(象 Troy PocketPro 100S 或 Netgear PS101)支持 JetDirect 协议的打印机。

check_hpjd 插件(该命令是 Nagios 插件软件发行包的标准组成部分)可以用 SNMP 使能的方式来监控 JetDirect 兼容型打印机。该插件可以检查如下打印机状态：

1. 卡纸
2. 无纸
3. 打印机离线
4. 需要人工干预
5. 墨盒墨粉低
6. 内存不足
7. 开外壳
8. 输出托盘已满
9. 和其他...

注意

如下指令假定你已经按照快速安装指南安装好 Nagios。可以参考安装好的样本配置文件

(**commands.cfg**、**templates.cfg** 等)。

4.10.2. 概览

监控网络打印机的状态很简单。有 JetDirect 功能的打印机一般提供 SNMP 功能，可以用 check_hpjd 插件来检测状态。

check_hpjd 插件只是当当前系统中安装有 net-snmp 和 net-snmp-utils 软件包时才会被编译和安装。要保证在/usr/local/nagios/libexec 目录下有 check_hpjd 文件再继承，否则，要安装好 net-snmp 和 net-snmp-utils 软件包再重新编译安装 Nagios 插件包。

4.10.3. 步骤

监控打印机需要做如下几步：

1. 做些事先准备工作；
2. 创建一个用于监控打印机的主机与服务对象定义；
3. 重启动 Nagios 守护进程。

4.10.4. 已经做了什么？

为使这项工作更轻松，几个配置工作已经做好：

1. **check_hpjd** 的命令定义已经加到了 **commands.cfg** 配置文件中，可以用 **check_hpjd** 插件来监控网络打印机；
2. 一个网络打印机模板(命名为 **generic-printer**)已经在 **templates.cfg** 配置文件里创建好，可以更方便地加入一个新打印机设备的主机对象。

上面的监控配置文件可以在/usr/local/nagios/etc/objects/目录里找到。如果想做，可以修改里面的定义以更好地适用于你的情况。但是在此之前，推荐你要熟悉 Nagios 的配置之后再做。起初，最好只是按下面的大概修改一下以实现对网络打印机的监控。

4.10.5. 事先准备工作

在配置 Nagios 用于监控网络打印机之前，有些额外工作，记住这是要对*第一台*打印机设备进行监控。

编辑 Nagios 的主配置文件。

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

移除下面这行最前面的(#)号：

#cfg_file=/usr/local/nagios/etc/objects/printer.cfg

保存文件并退出编辑。

为何要这样？告诉 Nagios 查找/usr/local/nagios/etc/objects/printer.cfg 文件以取得额外对象定义。该文件中将加入网络打印机设备的主机与服务对象定义。这个配置文件里已经包含有一个样本主机、主机组和服务定义。给第一台打印机设备做监控，可以简单地修改这个文件而不需重生成一个。

4.10.6. 配置 Nagios

需要创建几个对象定义以进行网络打印机的监控。

打开 printer.cfg 文件并编辑它。

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

增加一个你要监控的网络打印机设备的主机对象定义。如果这是第一台打印机设备，可以简单地修改 printer.cfg 文件里的样本主机定义。将合理的值赋在 host_name、alias 和 address 域里。

```
define host{

        use             generic-printer ; Inherit default values from a
template

        host_name       hplj2605dn      ; The name we're giving to this
printer

        alias           HP LaserJet 2605dn      ; A longer name
associated with the printer

        address         192.168.1.30    ; IP address of the printer

        hostgroups      allhosts        ; Host groups this printer is
associated with

}
```

现在可以给监控的打印机加些服务定义(在同一个配置文件里)。如果是第一台被监控的网络打印机，可以简单地修改 printer.cfg 里的服务配置。

注意

要用你要刚刚加上的被监控打印机主机名替换样例对象"**hplj2605dn**"里的 **host_name** 域值。

按如下方式加好对打印机状态检测的服务定义。服务用 check_hpjd 插件来检测打印机状态，默认情况下每 10 分钟检测一次。SNMP 共同体串是"public"。

```
define service{

        use                     generic-service         ; Inherit values
from a template

        host_name               hplj2605dn              ; The name of the
host the service is associated with

        service_description     Printer Status          ; The service
description

        check_command           check_hpjd!-C public    ; The command
used to monitor the service

        normal_check_interval   10      ; Check the service every 10
minutes under normal conditions

        retry_check_interval    1       ; Re-check the service every
minute until its final/hard state is determined

}
```

加入一个默认每 10 分钟进行一次的 PING 检测服务。用于检测 RTA、丢包率和网络联接状态。

```
define service{

        use                     generic-service

        host_name               hplj2605dn

        service_description     PING

        check_command           check_ping!3000.0,80%!5000.0,100%

        normal_check_interval   10
```

```
        retry_check_interval    1
}
```

保存配置文件。

4.10.7. 重启动 Nagios

一旦在 printer.cfg 文件里加好新的主机和服务对象定义就可以监控网络打印机。为了开始，应该先验证配置文件并重启动 Nagios。

如果在验证配置过程中有任何错误信息，修改好配置文件再继续。保证验证过程完成且没有任何错误的情况下再重启动 Nagios!

4.11. 监控 Netware 服务器

4.11.1. 介绍

本文档描述了如何对 Netware 服务器的"私有"服务和属性进行监控，象这些：

1. 内存占用率
2. 处理器利用率
3. 缓冲区使用情况
4. 活动的联接
5. 磁盘卷使用率
6. 等

由 Netware 服务器提供的公众服务(HTTP、FTP 等)的监控可以按文档监控公众服务来做。

4.11.2. 概览

TODO...

注意

我在找一个志愿者来写就 HOWTO 文档。我只能接触到一台旧的 Netware 4.11 服务器，所以无法跟上形势需要。如果可以更新这个文档，请把它张贴到 NagiosCommunity wiki 里。

4.11.3. 其他资源

Novell 有一些有关 Nagios 如何来做 Netware 监控的文档，在网站的 Cool Solutions 栏目里，包括：

1. MRTGEXT: NLM module for MRTG and Nagios
2. Nagios: Host and Service Monitoring Tool
3. Nagios and NetWare: SNMP-based Monitoring
4. Monitor DirXML/IDM Driver States with Nagios
5. Check NDS Login ability with Nagios
6. NDPS/iPrint Print Queue Monitoring by Nagios
7. check_gwiaRL Plugin for Nagios 2.0

4.12. 监控公众服务平台

4.12.1. 介绍

本 文档介绍如何来监控一些公众化的服务、应用及协议。所谓的"公众"服务是指在网络里常见的服务－不管是本地网络还是因特网上都是这样。公众服务包括象 －HTTP 服务(WEB)、POP3、IMAP/SMTP、FTP 和 SSH。其实在日常使用中还有更多的基础服务。这些服务与应用，包括所依托的协议，可 以被 Nagios 直接监控而不需要额外的支持。

与之相对，私有服务如果没有某些中间件做代理 Nagios 是无法监控的。主机上的私有服务比 如说 CPU 负荷、内存占用率、磁盘利用率、当前登录用户、进程信息等。这些私有服务或属性不能暴露给外部客户。这样就要在这些被监控的主机上安装一些中间 件做代理来取得此类信息。更多有关在不同类型主机上的私有服务的信息可以查阅如下文档：

1. 监控 Windows 主机
2. 监控 Netware 服务器
3. 监控 Linux/Unix 机器

提示

偶尔发现有些私有服务和应用的信息可以通过 SNMP 来做监控。SNMP 代理可以让远程监控系统提取一些有用信息。更多有关 SNMP 来监控服务信息的内容可以查阅一下监控交换机和路由器。

注意

如下内容假定你已经按照快速安装指南安装好 Nagios 系统。如下的配置样例对象可以参考安装包里的 **commands.cfg** 和 **localhost.cfg** 两个配置文件。

4.12.2. 监控服务的插件

当需要对一些应用、服务或协议进行监控时，可能已经有一个用于监控的插件了。Nagios 的官方插件包带有插件可以用于监控很多种服务与协议，在插件包的 contrib/目录下有很多可用的插件。而且在 NagiosExchange.org 网站上有许多额外的其他人写的插件可以试试看！

如果不巧没有找到要监控所需的插件，可以自己写一个。插件写起来很容易，不要怕它，读一读插件的开发这篇文档吧。

下面将会领略一下你迟早会用到的一些基础服务，每个服务都可以在 Nagios 插件包里找到对应的检测插件。下面就开始了...

4.12.3. 创建一个主机对象定义

在监控一个服务之前需要先定义一个服务要绑定的主机对象。可以把主机对象定义放在 cfg_file 域所指向的任何一个对象配置文件里或是放在由 cfg_dir 域所指向的任何一个目录下的文件里。如果已经定义过一个主机对象了，可以跳过这一步。

在本例中，假定你想在一台主机上监控许多种服务，把这台主机命名为 remotehost。可以把主机对象定义放在它自己的配置文件里或加到一个已有的对象定义文件里。这里有个 remotehost 对象的定义象是这样：

```
define host{

        use             generic-host            ; Inherit default values
from a template

        host_name               remotehost              ; The name we're
giving to this host

        alias           Some Remote Host        ; A longer name
associated with the host

        address         192.168.1.50            ; IP address of the host
```

```
        hostgroups              allhosts                ; Host groups
this host is associated with

        }
```

现在可以有了一个可被监控的主机对象了，之后就可以定义所要监控的服务。有了主机对象定义，服务对象定义可以加在任何一个对象配置文件的任何一个位置里。

4.12.4. 创建服务对象定义

在 Nagios 里每个要监控的服务都必须给出一个绑定在刚才定义出的主机上的一个服务对象。可以把服务对象放在任何一个由 cfg_file 域指向的对象配置文件里或是放在 cfg_dir 域所指向的目录下。

下面会给出一些要监控的样例服务(HTTP、FTP 等)的对象定义。

4.12.5. 监控 HTTP

有时不管是你或是其他人需要监控一个 Web 服务器。**check_http** 插件就是做这件事的。插件可以透过 HTTP 协议并监控响应时间、错误代码、返回页面里的字符串、服务器证书和更多的东西。

在 **commands.cfg** 文件里包含有一个使用 **check_http** 插件的命令定义。象是这样的：

```
define command{

        name            check_http

        command_name    check_http

        command_line    $USER1$/check_http -I $HOSTADDRESS$ $ARG1$

        }
```

对于监控在 **remotehost** 机器上的 HTTP 服务的简单的服务对象定义会象是这样的：

```
define service{

        use             generic-service         ; Inherit default values
from a template

        host_name               remotehost
```

```
        service_description     HTTP

        check_command   check_http

        }
```

这个服务对象定义将会监控位于 **remotehost** 机器上的 HTTP 服务。如果 WEB 服务在 10 秒内没有响应将产生警报或是返回了一个 HTTP 的错误码(403、404 等)也会产生警报。这是最基本的监控要求，很简单，不是么？

提示

对于更高级的监控，可以在命令行下手工运行 **check_http** 插件，带着**--help** 参数，看看这个插件有什么选项。**--help** 语法对于本文档所涉及的插件都适用。

下 面看看对 HTTP 服务更高级的监控。这个服务定义将会检测如下内容：在 URI(/download/index.php)里是否包含有"latest- version.tar.gz"字符串。如果没有指定字符串，或是 URI 非法也或许是在 5 秒内没有响应的话，它将产生一个错误。

```
define service{

        use             generic-service         ; Inherit default values
from a template

        host_name               remotehost

        service_description     Product Download Link

        check_command   check_http!-u /download/index.php -t 5 -s
"latest-version.tar.gz"

        }
```

4.12.6. 监控 FTP

监控 FTP 服务可以用 **check_ftp** 插件。在 **commands.cfg** 文件里包含有一个使用 **check_ftp** 插件的命令样例，象是这样：

```
define command{

        command_name    check_ftp
```

```
        command_line    $USER1$/check_ftp -H $HOSTADDRESS$ $ARG1$

        }
```

对 remotehost 主机上的 FTP 服务的简单监控的例子象是这样：

```
define service{

        use             generic-service         ; Inherit default values
from a template

        host_name               remotehost

        service_description     FTP

        check_command   check_ftp

        }
```

这个服务对象定义将会在 FTP 服务器 10 秒内不响应时给出一个警报。

下面对 FTP 做些高级监控。下面这个服务将检测 FTP 服务器绑定在 remotehost 主机上的 1023 端口的 FTP 服务。如果在 5 秒内没有响应或是服务里没有回应字符串"Pure-FTPd [TLS]"时将会产生警报。

```
define service{

        use             generic-service         ; Inherit default values
from a template

        host_name               remotehost

        service_description     Special FTP

        check_command   check_ftp!-p 1023 -t 5 -e "Pure-FTPd [TLS]"

        }
```

4.12.7. 监控 SSH

监控 SSH 服务可以用 check_ssh 插件。在 commands.cfg 文件里包含在使用 check_ssh 插件有样例命令，象是这样：

```
define command{
```

```
        command_name    check_ssh

        command_line    $USER1$/check_ssh $ARG1$ $HOSTADDRESS$

        }
```

监控 **remotehost** 上的 SSH 服务的简单例子象是这样：

```
define service{

        use             generic-service         ; Inherit default values
from a template

        host_name               remotehost

        service_description     SSH

        check_command   check_ssh

        }
```

这个服务对象定义将会在 SSH 服务 10 秒内不响应时给出一个警报。

下面对 SSH 做些高级监控。下面这个服务将检测 SSH 服务，如果 5 秒内不响应或是服务器版本串里没有能匹配字符串"OpenSSH_4.2"时产生一个警报。

```
define service{

        use             generic-service         ; Inherit default values
from a template

        host_name               remotehost

        service_description     SSH Version Check

        check_command   check_ssh!-t 5 -r "OpenSSH_4.2"

        }
```

4.12.8. 监控 SMTP

用 **check_smtp** 插件来监控 EMail 服务。**commands.cfg** 文件里包含有使用 **check_smtp** 插件的命令，象是这样：

```
define command{
```

```
        command_name    check_smtp

        command_line    $USER1$/check_smtp -H $HOSTADDRESS$ $ARG1$

        }
```

监控 remotehost 上的 SMTP 服务的简单例子象是这样：

```
define service{

        use             generic-service         ; Inherit default values
from a template

        host_name               remotehost

        service_description     SMTP

        check_command   check_smtp

        }
```

这个服务对象定义将会在 SMTP 服务器 10 秒内不响应时给出一个警报。

下面是更高级的监控。下面这个服务将检测 SMTP 服务，如果服务在 5 少内不响应或是返回串里没有包含字符串"mygreatmailserver.com"时将产生一个警报。

```
define service{

        use             generic-service         ; Inherit default values
from a template

        host_name               remotehost

        service_description     SMTP Response Check

        check_command   check_smtp!-t 5 -e "mygreatmailserver.com"

        }
```

4.12.9. 监控 POP3

用 check_pop 插件来监控 EMail 的 POP3 服务。commands.cfg 文件里包含有使用 check_pop 插件的命令，象是这样：

```
define command{
```

```
        command_name    check_pop

        command_line    $USER1$/check_pop -H $HOSTADDRESS$ $ARG1$

        }
```

监控 **remotehost** 上的 POP3 服务的简单例子象是这样：

```
define service{

        use             generic-service         ; Inherit default values
from a template

        host_name               remotehost

        service_description     POP3

        check_command   check_pop

        }
```

这个服务对象定义将会在 POP3 服务 10 秒内不响应时给出一个警报。

下面是更高级监控。下面这个服务将检测 POP3 服务，当服务在 5 秒内不响应或是返回串里没有包含字符串"mygreatmailserver.com"时将产生一个警报。

```
define service{

        use             generic-service         ; Inherit default values
from a template

        host_name               remotehost

        service_description     POP3 Response Check

        check_command   check_pop!-t 5 -e "mygreatmailserver.com"

        }
```

4.12.10. 监控 IMAP

用 **check_imap** 插件可以监控 EMail 的 IMAP4 服务。**commands.cfg** 文件里包含有使用 **check_imap** 插件的命令，象是这样：

```
define command{
```

```
        command_name    check_imap

        command_line    $USER1$/check_imap -H $HOSTADDRESS$ $ARG1$

        }
```

监控 **remotehost** 上的 IMAP 服务的简单例子象是这样：

```
define service{

        use             generic-service         ; Inherit default values
from a template

        host_name               remotehost

        service_description     IMAP

        check_command   check_imap

        }
```

这个服务对象定义将在 IMAP 服务 10 秒内不响应时给出一个警报。

下面是更高级监控。下面这个服务将检测 IAMP4 服务，当服务在 5 秒内不响应或是返回串里没有包含有"mygreatmailserver.com"时将产生一个警报。

```
define service{

        use             generic-service         ; Inherit default values
from a template

        host_name               remotehost

        service_description     IMAP4 Response Check

        check_command   check_imap!-t 5 -e "mygreatmailserver.com"

        }
```

4.12.11. 重启动 Nagios

一旦在配置文件里加入了一个新的主机或是服务对象，要开始对其进行监控，必须要验证配置文件 n 并重启动 Nagios。

如果验证过程里报出错误信息，要修正配置后再继续。要保证在验证过程里没有发现任何错误之后再启动或重启 Nagios!