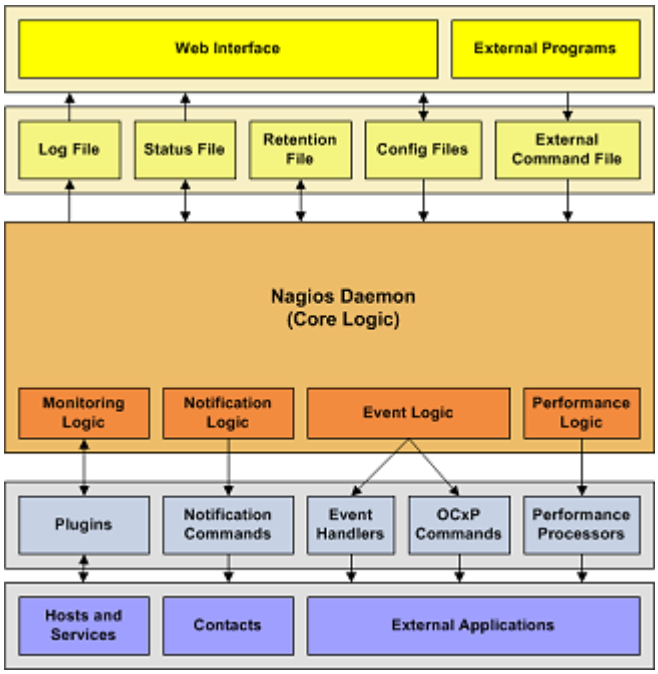
第 11 章 软件集成相关的内容

11.1. 软件集成概览

11.1.1. 介绍



Nagios 可以非常容易地与现有框架集成，这也就是为何 Nagios 被广泛地应用的一原因。有不少方式来与现有管理软件进行集成，你使用管理软件来监控你所拥有的各种各样的新型或用户定制的硬件、服务或是应用程序。

11.1.2. 集成的要点

为了监控新硬件、服务或是应用程序，审视如下的文档：

1. Nagios 插件
2. 插件 API
3. 强制检测
4. 事件处理句柄

为使 Nagios 取得外部应用程序的数据，审视如下的文档：

1. 强制检测
2. 外部命令

将状态、性能或是告警信息报送给外部应用，审视如下文档：

1. 事件处理句柄
2. OCSP 和 OCHP 命令
3. 性能数据
4. 告警

11.1.3. 集成事例

我记录下了一些事例来看一下 Nagios 是如何与外部程序集成的，它们是：

1. TCP Wrappers(安全事件报警)
2. SNMP Traps (卷备份作业的状态)

11.2. SNMP 陷井集成

11.2.1. 介绍

注意

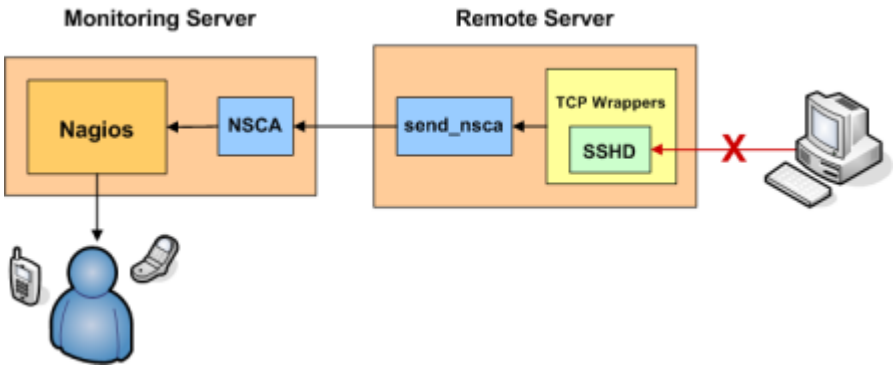
Nagios 并没有设计成一个可替代完全 SNMP 管理功能的象 HP OpenView 或 OpenNMS 那样的应用程序。然而，你可以在 Nagios 中设置好 SNMP 陷井来接收来自于网络中的主机发出的 SNMP 警报。

SNMP 的无所不管除了恶长以外一无是处。接收 SNMP 消息并将它放到 Nagio 里(象强制检测结果一样)是件很繁闷的事。为使之更简单，建议你取出 Alex Burger 的 SNMP Trap Translator 项目，它位于 http://www.snmptt.org，这里面在 Nagios 里集成了 Net-SNMP、SNMPTT 及增强型的消息陷井处理系统。

好了，就这么多。

11.3. TCP Wrapper 集成

11.3.1. 介绍



本 文档解释如何容易地在 Nagios 里用 TCP Wrapper 对联接尝试被拒绝而产生警报。例如，一个非授权主机试图联接 SSH 服务器，可以在 Nagios 里收到一个含有被拒绝主机名的警报。如果在 Linux/Unix 机器上实现了这个功能，就会惊讶地发现在网络里竟会有如此多的端口扫描。

集成的前提是：

1. 已经熟悉强制检测及其工作方式；
2. 已经熟悉可变服务及其工作方式；
3. 想要报警的主机(就是使用了 TCP wrappers 的机器)是一台远程主机(在例子中命名为 **firestorm**)。如果与运行 Nagios 的机器是同一台，需要对下面给出的例子做些修改才行。
4. 已经在 Nagios 监控服务机上安装有 NSCA 守护服务并且在那台安装有 TCP Wrapper 的远程主机上安装有 NSCA(**send_nsca**)客户端软件。

11.3.2. 定义一个服务

如果条件具备，给远程主机(**firestorm**)创建一个主机对象定义。

下一步，给这台运行有 TCP Wrapper 的主机(**firestorm**)在对象配置文件里加一个服务对象定义，服务对象定义的可能会是这样的：

```
define service{

        host_name                       firestorm

        service_description             TCP Wrappers

        is_volatile                     1

        active_checks_enabled           0

        passive_checks_enabled          1

        max_check_attempts              1
```

```
        check_command                   check_none

        ...

        }
```

在服务对象定义里有几个很重要：

1. 打开**可变服务(is_volatile=1)**功能开关，因为每一个检测到的报警都要送出一个通知；
2. 服务的自主检测被关闭，而强制检测功能打开。这说明 Nagios 将不会对服务自主地做检测－全部报警信息将是由外部源通过强制方式提供给 Nagios；
3. 服务对象里的 **max_check_attempts** 值设定为 1。这保证了当首条报警产生时就有送出一个通知。

11.3.3. 配置 TCP Wrappers

现在需要修改在 **firestorm** 机器上的**/etc/hosts.deny** 文件了。为使 TCP wrappers 对每个被拒绝的联接尝试都送出一条报警，需要加上这一行：

```
ALL: ALL: RFC931: twist
(/usr/local/nagios/libexec/eventhandlers/handle_tcp_wrapper %h %d) &
```

这行里假定在 **firestorm** 机器上有个脚本名字是 **handle_tcp_wrapper** 且放在**/usr/local/nagios/libexec/eventhandlers/**目录下，下面会给出脚本内容。

11.3.4. 写那个脚本

最后一件事是在 **firestorm** 上写那个 **handle_tcp_wrapper** 脚本，它将会把报警送给 Nagios 监控服务器，它可能会是这样的：

```
#!/bin/sh

/usr/local/nagios/libexec/eventhandlers/submit_check_result firestorm
"TCP Wrappers" 2 "Denied $2-$1" > /dev/null 2> /dev/null
```

注意 **handle_tcp_wrapper** 脚本调用了 **submit_check_result** 脚本来真正地送出报警。假定 Nagios 服务器被命名为 **monitor**，这个 **submit check_result** 脚本内容可能会是这样的：

```
#!/bin/sh

# Arguments

#       $1 = name of host in service definition
```

```
#       $2 = name/description of service in service definition

#       $3 = return code

#       $4 = output


/bin/echo -e "$1\t$2\t$3\t$4\n" | /usr/local/nagios/bin/send_nsca
monitor -c /usr/local/nagios/etc/send_nsca.cfg
```

11.3.5. 搞好了

已经全配置完成了，可以重启动一下 **firestorm** 机器上的 **inetd** 进程，并且重启动一下监控服务器上的 Nagios 进程。搞掟了。当 **firestorm** 上的 TCP wrappers 拒绝了一次联接尝试时，将会在 Nagios 里看到一条报警。报警的插件输出将会是这样的：

```
Denied sshd2-sdn-ar-002mnminnP321.dialsprint.net
```

11.4. Nagios 外部构件

11.4.1. 介绍

Nagios 有许多"外部构件"软件包可供使用。外部构件可以扩展 Nagios 的应用并使之与其他软件集成。

外部构件可用于：

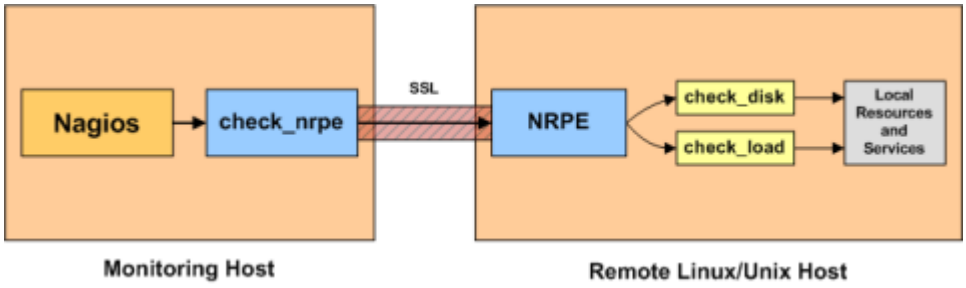
1. 通过 WEB 接口来管理配置文件
2. 监控远程主机(*NIX, Windows,等)
3. 实现对远程主机的强制检测
4. 减化并扩展告警逻辑
5. ...和其他更多事情

你可以通过访问如下站点找寻外部构件：

1. Nagios.org
2. SourceForge.net
3. NagiosExchange.org

这里对一些我开发的外部构件给一个简洁的介绍...

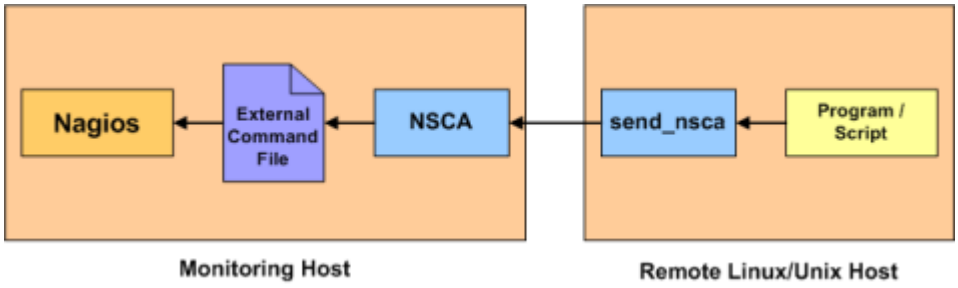
11.4.2. NRPE



NRPE 是一个可在远程 Linux/Unix 主机上执行的插件的外部构件包。如果你需要监控远程的主机上的本地资源或属性，如磁盘利用率、CPU 负荷、内存利用率等时是很有用的。象是用 check_by_ssh 插件来实现的功能一样，但是它不需要占用更多的监控主机的 CPU 负荷－当你需要监控成百上千个主机是这个很重要。

NRPE 外部构件包和文档可以在 http://www.nagios.org/上找到。

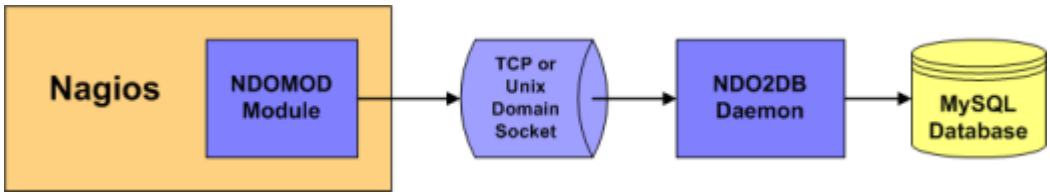
11.4.3. NSCA



NSCA 是一个可在远程 Linux/Unix 主机上执行强制检测并将结果传给 Nagios 守护进程的外部构件包。这在分布式和冗余/失效监控的设置时非常有用。

NSCA 外部构件包和文档可以在 http://www.nagios.org/上找到。

11.4.4. NDOUtils



NDOUtils 是一个可以把全部状态信息保存到 MySQL 数据库里的外部构件。外个 Nagios 的库实例都可以把它们监控的信息保存到统一的中心数据库并集中报告。它将为一个 Nagios 新的基于 PHH 的 WEB 接口程序提供数据源服务。

NDOUtils 外部构件包和文档可以在 http://www.nagios.org/上找到。