HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA ĐÀO TẠO SAU ĐẠI HỌC

MÔN: AN NINH MẠNG

BÁO CÁO

Đề tài:

NGHIÊN CỨU HỆ THỐNG GIÁM SÁT QUẢN TRỊ MẠNG NAGIOS

Lớp: M13CQIS02-B
Nhóm thực hiện:

1. **Nguyễn Thị Hạt**
2. **Lê Việt Hải**
3. **Trần Đăng Doanh**

Giảng viên: Hoàng Đăng Hải

THÁNG 10 - 2014

MỤC LỤC

LỜI MỞ ĐẦU

Network mornitoring hay tiếng việt hiểu là giám sát, theo dõi mạng là một trong những vấn đề hiện nay trở lên rất quan trọng trong việc quản trị các hệ thống mạng. Nó hạn chế tối đa việc mạng bị gián đoạn trong quá trình hoạt động. Nó đảm bảo việc khai thác tài nguyên có hiệu quả, đảm bảo an toàn, tin cậy cho những dịch vụ cung cấp… Hiện nay có rất nhiều công cụ giám sát mạng hỗ trợ cho công việc của người quản trị. Chức năng của chúng là giám sát thạng thái hoạt động của các thiết bị mạng, các dịch vụ mạng, và các máy đầu cuối tham gia vào mạng và thông báo cho người quản trị khi có sự cố hoặc khả năng sẽ sảy ra sự cố. Có cả những hệ thống thương mại như HPopen View… Hay nguồn mở như openNMS, Cacti, Nagios… Mỗi hệ thống lại có những ưu nhược điểm riêng. Tuy nhiên khả năng của chúng lại không hơn nhau nhiều lắm. Bài Báo cáo này tập trung vào việc nghiên cứu một hệ thống giám sát dựa trên Nagios, một sản phẩm nguồn mở được sử dụng rộng rãi.

Từ khi ra đời 2002 đến nay Nagios đã liên tục phát triển và rất được quan tâm. Cộng đồng quan tâm và sử dụng Nagios cho đến nay theo thống kê của http://nagios.org là vào khoảng 250.000 người. Từ phiên bản 1.0 đầu tiên, đến nay Nagios đã phát triển nên phiên bản 3.x và vẫn liên tục cho ra những phiên bản mới với tính năng mạnh mẽ hơn. Đặc biệt Nagios có khả năng phân tán. Vì vậy nó có thể giám sát các mạng khổng lồ, đạt cỡ 100.000 node.

Nội dung Báo cáo:

Chương 1: Giới thiệu tổng quan về Nagios, đưa ra cái nhìn khái quát về hệ thống Nagios.

Chương 2: Giới thiệu cơ bản về đặc điểm và cách thức cấu hình trong Nagios.

Chương 3: Chi tiết các chức năng của hệ thống Nagios.

Chương 4: Các khái niệm, vấn đề liên quan đến hệ thống Nagios.

CHƯƠNG 1. TỔNG QUAN VỀ NAGIOS

1.1. Chức năng của Nagios

- Giám sát trạng thái hoạt động của các dịch vụ mạng (SMTP, POP3, IMAP, HTTP, ICMP, FTP, SSH, DHCP, LDAP, DNS, name server, web proxy, TCP port, UDP port, cở sở dữ liệu: mysql, portgreSQL, oracle)

- Giám sát các tài nguyên các máy phục vụ và các thiết bị đầu cuối (chạy hệ điều hành Unix/Linux, Windows, Novell netware): tình trạng sử dụng CPU, người dùng đang log on, tình trạng sử dụng ổ đĩa cứng, tình trạng sử dụng bộ nhớ trong và swap, số tiến trình đang chạy, các tệp log hệ thống.

- Giám sát các thông số an toàn thiết bị phần cứng trên host như: nhiệt độ CPU, tốc độ quạt, pin, giờ hệ thống…

- Giám sát các thiết bị mạng có IP như router, switch và máy in. Với Router, Switch, Nagios có thể theo dõi được tình trạng hoạt động, trạng thái bật tắt của từng cổng, lưu lượng băng thông qua mỗi cổng, thời gian hoạt động liên tục (Uptime) của thiết bị. Với máy in, Nagios có thể nhận biết được nhiều trạng thái, tình huống sảy ra như kẹt giấy, hết mực…

- Cảnh báo cho người quản trị bằng nhiều hình thức như email, tin nhắn tức thời (IM), âm thanh …nếu như có thiết bị, dịch vụ gặp trục trặc

- Tổng hợp, lưu giữ và báo cáo định kỳ về tình trạng hoạt động của mạng.

1.2 Đặc điểm của Nagios

- Các hoạt động kiểm tra được thực hiện bởi các plugin cho máy phục vụ Nagios và các mô đun client trên các thiết bị của người dùng cuối, Nagios chỉ định kỳ nhận các thông tin từ các plugin và xử lý những thông tin đó (thông báo cho người quản lý, ghi vào tệp log, hiển thi lên giao diện web…).

- Thiết kế plugin đơn giản cho phép người dùng có thể tự định nghĩa và phát triển các plugin kiểm tra các dịch vụ theo nhu cầu riêng bằng các công cụ lập trình như shell scripts, C/C++, Perl, Ruby, Python, PHP, C#.

• Có khả năng kiểm tra song song trạng thái hoạt động của các dịch vụ(đồng thời kiểm tra nhiều dịch vụ).

• Hỗ trợ khai báo kiến trúc mạng. Nagios không có khả năng nhật dạng được topo của mạng. toàn bộ các thiết bị, dịch vụ muốn được giám sát đều phải khai báo và định nghĩa trong cấu hình.

• Gửi thông báo đến người/nhóm người được chỉ định sẵn khi dịch vụ/host được giám sát gặp vấn đề và khi chúng khôi phục hoạt động bình thường.(qua e-mail, pager, SMS, IM…)

• Khả năng định nghĩa bộ xử lý sự kiện thực thi ngay khi có sự kiện sảy ra với host/ dịch vụ.

• Giao diện web cho phép xem trạng thái của mạng, thông báo, history, tệp log.

1.3. Kiến trúc và tổ chức hoạt động

1.3.1 Kiến trúc của Nagios

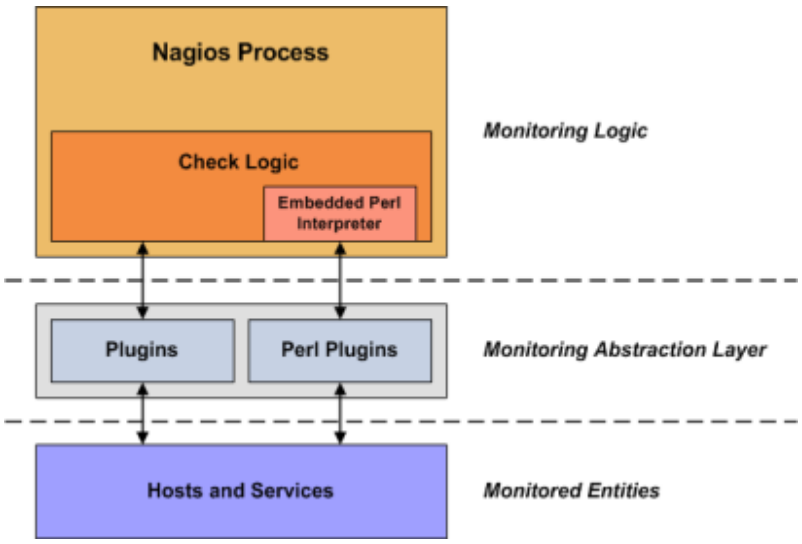
Hệ thống Nagios gồm hai phần chính:

1. Lõi Nagios
2. Plugin

Phần lõi nagios có chức năng quản lý các host/dịch vụ được giám sát, thu thập các kết quả kiểm tra (check) host/dịch vụ từ các plugin gửi về, biểu diễn trên giao diện chương trình, lưu trữ và thông báo cho người quản trị. Ngoài ra nó còn tổng hợp và đưa ra các báo cáo về tình hình hoạt động chung hoặc của từng host/dịch vụ trong một khoảng thời gian nào đó.

Plugin là bộ phận trực tiếp thực hiện kiểm tra host/dịch vụ. Mỗi một loại dịch vụ đều có một plugin riêng biệt được viết để phục vụ riêng cho công việc kiểm tra dịch vụ đó. Plugin là các script (Perl, C …) hay các tệp đã được biên dịch (executable). Khi cần thực hiện kiểm tra một host/dịch vụ nào đó Nagios chỉ việc gọi plugin tương ứng và nhật kết quả kiểm tra từ chúng. Với thiết kế như thế này, hệ thống Nagios rất dễ dàng được mở rộng và phát triển. Bất kì một thiết bị hay dịch vụ nào cũng có thể được

giám sát nếu như viết được plugin cho nó. Hình bên dưới cho ta thấy sự tương quan giữa các thành phần trong Nagios.



Hình 1.1 Sơ đồ tổ chức của Nagios

1.3.2. Cách thức tổ chức hoạt động

Nagios có 5 cách thực thi các hành động kiểm tra:

1.3.2.1. Kiểm tra dịch vụ trực tiếp.

Đối với các dịch vụ mạng có giao thức giao tiếp qua mạng như smtp, http, ftp… Nagios có thể tiến hành kiểm tra trực tiếp một dịch vụ xem nó đang hoạt động hay không bằng cách gửi truy vấn kết nối dịch vụ đến server dịch vụ và đợi kết quả trả về. Các plugin phục vụ kiểm tra này được đặt ngay trên server Nagios.

1.3.2.2. Chạy các plugin trên máy ở xa bằng secure shell

Nagios server không có cách nào có thể truy cập trực tiếp client để theo dõi những thông tin như tình trạng sử dụng ổ đĩa, swap, tiến trình … Để làm được việc này thì trên máy được giám sát phải cài plugin cục bộ. Nagios sẽ điều khiển các plugin cục bộ trên client qua secure shell ssh bằng plugin *check_by_ssh.* Phương pháp này yêu cầu một tài khoản truy cập host được giám sát nhưng nó có thể thực thi được tất cả các plugin được cài trên host đó.

1.3.2.3. Bộ thực thi plugin từ xa (NRPE - Nagios Remote Plugin Executor)

NRPE là một addon đi kèm với Nagios. Nó trợ giúp việc thực thi các plugin được cài đặt trên máy/thiết bị được giám sát. NRPE được cài trên các host được giám sát. Khi nhận được truy vấn từ Nagios server thì nó gọi các plugin cục bộ phù hợp trên host này, thực hiện kiểm tra và trả về kết quả cho Nagios server. Phương pháp này không đòi hỏi tài khoản truy cập host được giám sát như sử dụng ssh. Tuy nhiên cũng như ssh các plugin phục vụ giám sát phải được cài đặt trên host được giám sát. NRPE có thể thực thi được tất cả các loại plugin giám sát. Nagios có thể điều khiển máy cài NRPE kiểm tra các thông số phần cứng, các tài nguyên, tình trạng hoạt động của máy đó hoặc sử dụng NRPE để thực thi các plugin yêu cầu truy vấn dịch vụ mạng đến một máy thứ 3 để kiểm tra hoạt động của các dịch vụ mạng như http, ftp, mail…

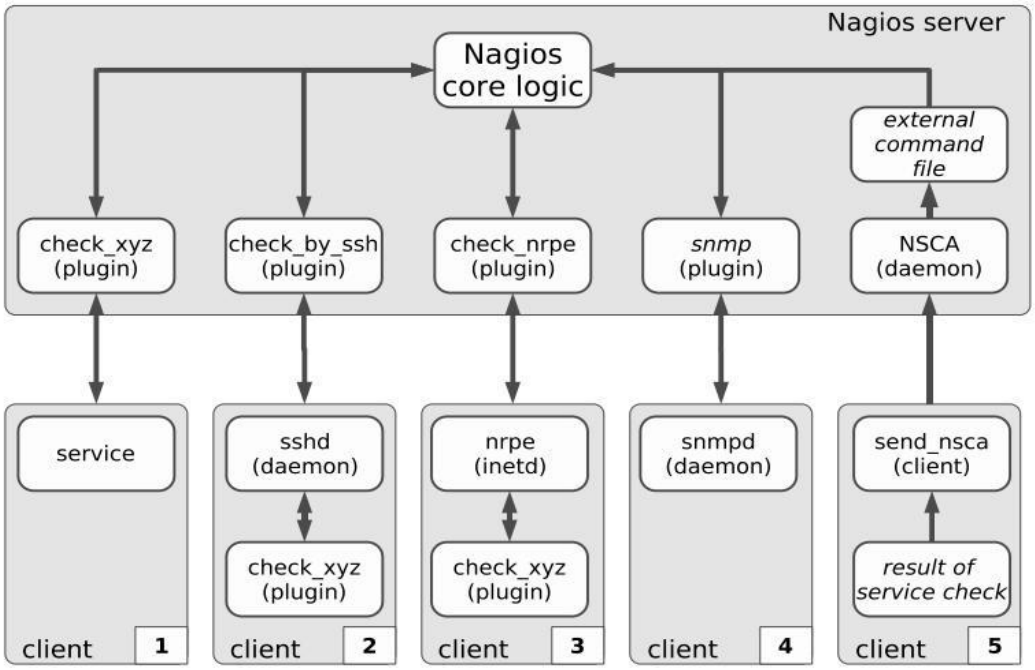
1.3.2.4 Giám sát qua SNMP

Cốt lõi của giao thức SNMP (SimpleNetwork Management Protocol)là tập hợp đơn giản các hoạt động giúp nhà quản trị mạng có thể quản lý, thay đổi trạng thái thiết bị. Hiện nay rất nhiều thiết bị mạng hỗ trợ giao thức SNMP như Switch, router, máy in, firewall ... Nagios cũng có khả năng sử dụng giao thức SNMP để theo dõi trạng thái của các client, các thiết bị mạng có hỗ trợ SNMP. Qua SNMP, Nagios có được thông tin về tình trạng hiện thời của thiết bị. Ví dụ như với SNMP, Nagios có thể biết được các cổng của Switch, router có mở hay không, thời gian Uptime (chạy liên tục) là bao nhiêu…

1.3.2.5. NSCA (Nagios Service Check Acceptor)

Nagios được coi là một phần mềm rất mạnh vì nó dễ dàng được mở rộng và kết hợp với các phần mềm khác. Nó có thể tổng hợp thông tin từ các phần mềm kiểm tra của hãng thứ ba hoặc các tiến trình Nagios khác về trạng thái của host/dịch vụ. Như thế Nagios không cần phải lập lịch và chạy các hành động kiểm tra host/dịch vụ mà các ứng dụng khác sẽ thực hiện điều này và báo cáo thông tin về cho nó. Và các ứng dụng kiểm tra có thể tận dụng được khả năng rất mạnh của Nagios là thông báo và tổng hợp báo cáo. Nagios sử dụng công cụ NSCA để gửi các kết quả kiểm tra từ ứng

dụng của bạn về server Nagios. Công cụ này giúp cho thông tin gửi trên mạng được an toàn hơn vì nó được mã hóa và xác thực.



Hình 1.2 Các cách thức thực hiện kiểm tra.

Hình trên cho ta cái nhìn tổng quan về các cách thức kiểm tra dịch với nagios. Có 5 client được giám sát bằng 5 cách thức khác nhau:

- client 1: Nagios sử dụng plugin 'check_xyz' được cài đặt ngay trên server Nagios để gửi truy vấn kiểm tra dịch vụ trên client(http, ftp, dns, smtp…)
- client 2, 3: Nagios sử dụng các plugin trung gian để chạy plugin 'check_xyz' giám sát được cài đặt trực tiếp trên client. (bởi vì có những dịch vụ không có hỗ trợ giao thức trao đổi qua mạng, ví dụ khi bạn muốn kiểm tra dung lượng ổ đĩa cứng còn trống trên client…)
- client 4: Kiểm tra dịch vụ qua giao thức snmp, nagios server sẽ sử dụng plugin check_snmp để kiểm tra các dịch vụ trên client có hỗ trợ giao thức SNMP. Rất nhiều thiết bị mạng như router, switch, máy in… có hỗ trợ giao thức SNMP.
- Client 5: Đây là phương pháp kiểm tra bị động. Nagios không chủ động kiểm tra dịch vụ mà là client chủ động gửi kết quả kiểm tra dịch vụ về cho Nagios thông qua plugin NSCA. Phương pháp này được áp dụng nhiều trong giám sát phân tán. Với các mạng có quy mô lớn, người ta có thể dùng nhiều server Nagios để giám sát từng phần của mạng. Trong đó có một server Nagios trung tâm thực hiện tổng hợp kết quả từ các server Nagios con thông qua plugin NSCA.

CHƯƠNG 2. TỔNG QUAN CẤU HÌNH

2.1. Tổng quan cấu hình

2.1.1. Các tệp cấu hình chương trình

Thư mục */usr/local/nagios/etc/*

- Tệp cấu hình chính *nagios.cfg*. Thiết đặt những tùy chọn chung nhất của Nagios, tác động đến cách thức hoạt động của Nagios. Trong nagios.cfg bạn có thể khai báo đường dẫn các tệp cấu hình còn lại, tệp log, tệp đệm… . hoặc bật tắt các tùy chọn cấu hình như cho phép thông báo, sử dụng lệnh ngoại trú, kiểm tra bị động, cách thức log, cập nhật…

- Tệp cấu hình tài nguyên *resource.cfg*. Các tệp tài nguyên dùng để lưu trữ các nhãn(macro) được định nghĩa bởi người dùng, và lưu trữ những thông tin nhạy cảm(như mật khẩu…), ẩn với CGIs. Bạn có thể chỉ định một hay nhiều tùy chọn tệp tài nguyên bằng cách sử dụng chỉ thị resource_file trong tệp cấu hình chính.

- Tệp cấu hình CGI *cgi.cfg*. Tệp cấu hình CGI chứa tập các chỉ thị ảnh hưởng đến hoạt động của CGIs và cách thức hiển thị thông tin trên giao diện web.

2.1.2. Các tệp cấu hình đối tượng

Thư mục */usr/local/nagios/etc/objects*

- Nơi lưu trữ các tệp cấu hình đối tượng được giám sát và quản lý trong nagios. Các tệp định nghĩa đối tượng được sử dụng để định nghĩa host, dịch vụ, liên hệ(contacts), nhóm liên hệ(contactgroups), lệnh… đây là nơi định nghĩa tất cả mọi thứ mà bạn muốn giám sát và cách mà bạn giám sát chúng. Bạn có thể chỉ định một hay nhiều tệp định nghĩa đối tượng bằng sử dụng các chỉ thị cfg_file và cfg_dir trong tệp cấu hình chính. Các tệp cấu hình sẵn có là:

- Localhost.cfg //định nghĩa các máy linux
- Contact.cfg //đn người dùng
- Printer.cfg //đn các máy in

- Switch.cfg //đn switch
- Window.cfg //đn máy window
- Command.cfg //đn các lệnh
- Template.cfg //mẫu đn có sẵn
- Timeperiods.cfg //đn các chu ki thời gian

2.2. Cách thức định nghĩa đối tượng trong các tệp cấu hình đối tượng

Các đối tượng (bao gồm host, dịch vụ, người liên hệ, lệnh, nhóm, chu kỳ thời gian) có thể đươc định nghĩa trong bất kì tệp nào có đuôi .cfg và khai báo đường dẫn trong tệp cấu hình chính qua tùy chọn cfg_file. Tệp template.cfg đã có sẵn những định nghĩa đối tượng chuẩn, các định nghĩa đối tượng mới có thể kế thừa khuôn mẫu của định nghĩa chuẩn và có thể thay đổi đi mội số tùy chọn cho phù hợp với từng yêu cầu sử dụng.

2.2.1. Định nghĩa host

Host là một trong những đối tượng cơ bản nhất được giám sát. Đặc điểm của host là:

- Host thường là các thiết bị vật lý trên mạng như server, workstation, router,switch, printer…
- Host có địa chỉ xác định(IP hoặc MAC).
- Host thường có ít nhất một dịch vụ liên quan đến nó.
- Một host có thể có mối quan hệ cha/con, phụ thuộc với host khác.

Khi định nghĩa đối tượng host bạn có thể kế thừa mẫu định nghĩa host có trong tệp template.cfg. Mẫu định nghĩa này có trong phần phụ lục cuối tài liệu. Tuy nhiên với mỗi host được định nghĩa mới thì có 3 tùy chọn bắt buộc phải khai báo cho phù hợp. Đó là tên host, bí danh và địa chỉ IP của host.

```
define host{
```

```
use                     linux-server  //kế thừa định nghĩa mẫu có sẵn
host_name               fedora10
alias                   f10
address                 192.168.1.254
…
}
```

2.2.2. Định nghĩa dịch vụ

Định nghĩa dịch vụ dùng để khai báo dịch vụ được giám sát chạy trên host. Dịch vụ ở đây có thể hiểu là các dịch vụ mạng thực sự như là POP, SMTP, HTTP… hay là chỉ là một số số liệu của host như số lượng người dùng, ổ đĩa còn trống… Các tùy chọn dưới đây là bắt buộc khi định nghĩa một dịch vụ mới. Các tùy chọn còn lại có thể tham khảo phần phụ lục.

```
define service{
        host_name               linux-server
  service_description     check-disk-sda1
  check_command                 check-disk!/dev/sda1
  max_check_attempts  5
  check_interval 5
  retry_interval   3
  check_period            24x7
  notification_interval     30
  notification_period     24x7
  notification_options     w,c,r
  contact_groups                linux-admins
  }
```

Tuy nhiên cũng giống như định nghĩa host, nếu sử dụng kế thừa từ định nghĩa mẫu thì khi định nghĩa một host mới chỉ cần khai báo 4 tùy chọn:

```
define service{
        use                     generic-service
        host_name               linux-server
        service_description     check-disk-sda1
```

```
check_command                check-disk!/dev/sda1
}
```

2.2.3. Định nghĩa Lệnh

Tất cả các hành động của Nagios như kiểm tra host/dịch vụ, thông báo, xử lý sự kiện đều được thực hiện bằng cách gọi lệnh. Tất cả các lệnh trong Nagios đều được định nghĩa trong tệp cấu hình commands.cfg.

Khuôn dạng của một lệnh được định nghĩa:

```
define command{
    command_name        Tên lệnh
    command_line Người dùng/script! Danh sách tham số
}
```

Ví dụ:

```
define command{
    command_name        check_local_disk
    command_line $USER1$/check_disk! –w $ARG1$ -c $ARG2$ -p $ARG3$
}
```

Một lệnh được định nghĩa gồm hai thành phần tên lệnh và nội dung lệnh. Trong đó $USER1$ là nhãn người dùng được định nghĩa trong tệp tài nguyên resource.cfg. $ARG1$, $ARG2$, $ARG3$ là các nhãn tham số vào của lệnh. Và check_disk trong ví dụ trên có thể thay bằng một script, file tự chạy bất kì… Như ví dụ trên, khi cần kiểm tra ổ đĩa cứng của một host A, Nagiso sẽ gọi lệnh check_local_disk với các tham số vào của host A. Lệnh này sẽ thực thi script check_disk với các tham số đó.

2.2.4. Các định nghĩa khác

Ngoài ra còn các định nghĩa khác như nhóm host, nhóm dịch vụ, nhóm liên lạc, chu kỳ thời gian được giới thiệu trong phần phụ lục của tài liệu …

CHƯƠNG 3. CÁC DỊCH VỤ GIÁM SÁT

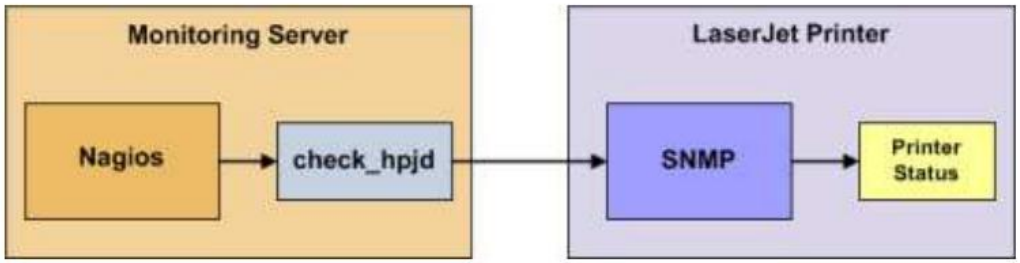
3.1. Giám sát các thiết bị mạng

Nagios giám sát các thiết bị qua giao thức SNMP. Vì vậy máy giám sát(Nagios) phải cài đặt net-snmp và net-snmp-utils với redhat/fedora hoặc libsnmp-base, snmp, snmpd,libsnmp15 với debian/ubuntu trước khi biên dịch và cài đặt nagios plugin. Các thiết bị được giám sát phải có IP, hỗ trợ snmp, và snmp ở trạng thái bật.

3.1.1. Máy in

3.1.1.1. Tổng quan

Nagios sử dụng plugin *check_hpjd* cho việc giám sát trạng thái của máy in. Plugin *check_hpjd* sử dụng giao thức SNMP để xác định trạng thái của máy in.



Hình 3.1 Giám sát máy in

Check_hpjd có khả năng phát hiện, cảnh báo, ghi lại các sự cố của máy in như:

- kết nối đến máy in(ping đến máy in)
- Kẹt giấy
- Hết giấy
- Máy in tắt
- Yêu cầu xen vào
- Mực ít
- Thiếu bộ nhớ
- Khay ra giấy bị đầy

3.1.1.2. Cấu hình giám sát máy in

Mặc định lệnh "check_hpjd" đã được định nghĩa trong tệp commands.cfg. Nó cho phép bạn gọi plugin *check_hpjd* plugin để giám sát máy in trong mạng. thứ nữa là đã có một mẫu định nghĩa máy in(được gọi là generic-printer) được tạo trong tệp templates.cfg. Nó cho phép bạn thêm một định nghĩa máy in mới khá đơn giản. Khi định nghĩa máy in được giám sát mới bạn chỉ cần khai báo sử dụng mẫu này và tùy chỉnh một số tùy chọn cho phù hợp.

Trong lần đầu tiên cấu hình Nagios giám sát máy in bạn cần phải sửa tệp cấu hình Nagios. Và sau đó không phải làm lại việc này nữa.

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

Sóa dấu (#) ở đầu dòng như bên dưới trong tệp cấu hình:

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

Lưu tệp cấu hình và thoát. Đây là khai báo sử dụng tệp cấu hình cho máy in.

Tệp cấu hình /usr/local/nagios/etc/objects/printer.cfg sẽ là nơi để bạn thêm những định nghĩa host và dịch vụ mới cho máy in. Tệp cấu hình này đã chứa một vài ví dụ về định nghĩa host, hostgroup, và dịch vụ. Bạn có thể sửa những mẫu này để giám sát một máy in trong lần đầu tiên cấu hình. Bạn cần phải định nghĩa mới đối tượng máy in khi giám sát một máy in mới. Mở tệp printer.cfg.

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

Thêm một định nghĩa host mới cho máy in trong mạng mà bạn sẽ giám sát. Thay đổi trường host_name, alias, và address theo các giá trị của máy in.

```
define host{
    use             generic-printer              ; Thừa kế giá trị mặc định của mẫu
    host_name               hplj2605dn      ; Tên của máy in
    alias           HP LaserJet 2605dn      ; Tên khác của máy in
    address         192.168.1.30             ; Địa chỉ IP của máy in
    hostgroups      allhosts              ; Host groups của máy in
    }
```

Bây giờ bạn có thể bổ xung định nghĩa các dịch vụ được giám sát. Nếu là lần đầu tiên định nghĩa thì bạn có thể sửa luôn định nghĩa dịch vụ mẫu trong tệp printer.cfg.

Thêm định nghĩa dịch vụ bên dưới để kiểm tra trạng thái của máy in. 10 phút một lần check_hpjd plugin sẽ kiểm tra trạng thái của máy in.

```
define service{
    use                 generic-service         ; Kế thừa từ mẫu
    host_name           hplj2605dn              ; Tên của máy in được giám sát
    service_description     Printer Status          ; Mô tả dịch vụ
    check_command               check_hpjd!-C public    ; Lệnh để sử dụng giám sát
dịch vụ
    normal_check_interval   10      ;  kiểm tra lại dịch vụ sau 10 phút
    }
```

Thêm định nghĩa dịch vụ bên dưới để ping đến máy in 10 phút một lần. Nó phục vụ cho việc giám sát RTA, sự mất gói tin, và kết nối của mạng.

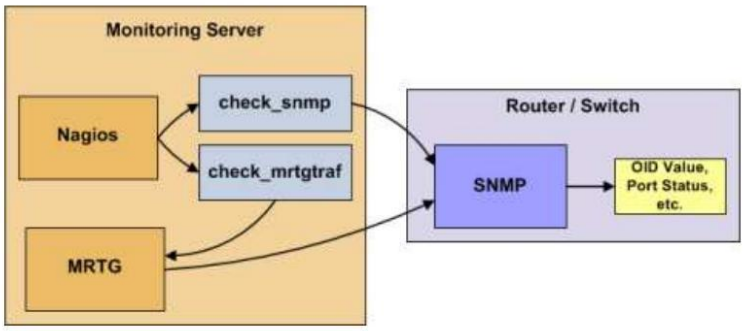
```
define service{
    use                 generic-service
    host_name               hplj2605dn
    service_description     PING
    check_command           check_ping!3000.0,80%!5000.0,100%
    normal_check_interval   10
    retry_check_interval    1
    }
```

Lưu tệp lại và kiểm chứng lại cấu hình và khởi động lại Nagios.

3.1.2. Switch, router

3.1.2.1. Tổng quan

Nagios sử dụng 2 plugin giám sát các thiết bị này đó là *check_snmp*, *check_mrtgtraf*. Nếu muốn sử dụng *check_mrtgtraf* để giám sát băng thông thì máy Nagios phải cài MRTG(chương trình giám sát lưu lượng mạng). Hình bên dưới mô tả cách thức thực hiện việc giám sát Router/switch.



Hình 3.2: Giám sát Router/Switch

Khả năng giám sát của Nagios:

- Kết nối đến thiết bị(ping thiết bị).
- Trạng thái up/down của các cổng.
- Sử dụng băng thông,lưu lượng trên các cổng.
- Tỉ lệ mất gói tin, trung bình trễ trọn vòng(RTA)

3.1.2.2. Cấu hình giám sát router/switch

Hai lệnh *check_snmp* và *check_local_mrtgtraf* đã được định nghĩa trong tệp commands.cfg. Chúng cho phép bạn gọi plugin *check_snmp* và *check_mrtgtraf* plugin để giám sát router/switch.

Mẫu định nghĩa Router/switch (được gọi là generic-switch) đã được tạo trong tệp templates.cfg. Nó cho phép bạn thêm các định nghĩa router/switch host rất nhanh chóng. Các tệp cấu hình trên được đặt trong thư mục */usr/local/nagios/etc/objects/* . Bạn có thể sử dụng các định nghĩa sẵn có này hoặc thêm các đinh nghĩa cho phù hợp với nhu cầu của mình.

Trong lần đầu tiên cấu hình Nagios giám sát switch bạn cần phải sửa tệp cấu hình Nagios. Và sau đó không phải làm lại việc này nữa

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

Sóa dấu (#) ở đầu dòng như bên dưới trong tệp cấu hình:

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

Lưu lại và thoát.

Tệp tin /usr/local/nagios/etc/objects/switch.cfg là nơi để định nghĩa cho host và dịch vụ router and switch. Trong này có sẵn một số định nghĩa về host, hostgroup, và dịch vụ mẫu. Trong lần đầu tiên định nghĩa router/switch bạn có thể sửa luôn các định nghĩa mẫu này tốt hơn là tạo một định nghĩa mới.

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

Tạo một định nghĩa host đơn giản như bên dưới.

```
define host{
    use             generic-switch         ; kế thừa giá trị mặc định từ mẫu
    host_name               linksys-srw224p        ; tên của switch
    alias           Linksys SRW224P Switch  ; bí danh của switch
    address         192.168.1.253          ;  địa chỉ IP của switch
    hostgroups      allhosts,switches                   ; Host group của switch
```

3.1.2.3. Giám sát Tỉ lệ mất gói tin, trung bình trễ trọn vòng

Ví dụ thêm định nghĩa dịch vụ dưới đây để giám sát việc mất gói tin và RTA(round trip average) giữa Nagios host và switch 5 phút một lần trong điều kiện bình thường.

```
define service{
    use                 generic-service ; Inherit values from a template
    host_name                   linksys-srw224p
    check_command           check_ping!200.0,20%!600.0,60%  ;
}
```

Dịch vụ này trả về:

- CRITICAL, nếu round trip average (RTA) lớn hơn 600 milliseconds hoặc số gói bị mất trên 60%,

- WARNING, cảnh báo nếu RTA lớn hơn 200ms hoặc gói tin bị mất lớn hơn 20%.

- OK, Ngược lại chạy bình thường nếu RTA nhỏ hơn 200ms và số gói bị mất nhỏ hơn 20%.

3.1.2.4. Giám sát thông tin trạng thái qua SNMP

Nếu switch hay router của bạn hỗ trợ SNMP, bạn có thể giám sát rất nhiều thông tin bằng check_snmp plugin. Bổ xung định nghĩa dịch vụ bên dưới để định nghĩa uptime(thời gian chạy liên tục) của switch.

```
define service{
    use                 generic-service ;     kế thừa giá trị từ mẫu
    host_name                 linksys-srw224p
    service_description    Uptime
    check_command         check_snmp!-C public -o sysUpTime.0
    }
```

Trong mục check_command ở trên, tham số "-C public" chỉ ra rằng tên SNMP là "public" và "-o sysUpTime.0" chỉ ra OID được kiểm tra.

Nếu bạn muốn giám sát một giao diện/cổng(port/interface) nào đó trên switch ở trạng thái up hay down, bạn thêm một định nghĩa dịch vụ như sau:

```
define service{
    use                 generic-service Thừa kế giá trị từ mẫu
    host_name                 linksys-srw224p
    service_description    Port 1 Link Status
    check_command         check_snmp!-C public -o ifOperStatus.1 -r 1 -m
RFC1213-MIB
    }
```

Trong ví dụ trên, tham số "-o ifOperStatus.1" chỉ vị trí cổng ở đây là 1. Tham số "-r 1" có ý nghĩa là check_snmp plugin trả về trạng thái OK nếu "1" được tìm thấy trong kết quả SNMP (1 chỉ trạng thái "up"của cổng) và CRITICAL nếu nó không tìm thấy. Tham số "-m RFC1213-MIB" chỉ ra rằng check_snmp plugin chỉ tải "RFC1213-MIB" thay vì tải các MIB được cài trên hệ thống của bạn. Điều này giúp làm tăng tốc độ mọi thứ lên.

Lưu ý:

Bạn có thể tìm ra các OID được giám sát trên switch bằng cách sử dụng lệnh(thay 192.168.1.253 thành địa chỉ IP của switch bạn quản lý):

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

3.1.2.5 Giám sát băng thông và tỉ lệ lưu lượng

Nếu bạn đang giám sát băng thông sử dụng trên switch hay router sử dụng MRTG, bạn có thể nhận được cảnh báo khi tỉ lệ lưu lượng đạt tới ngưỡng mà bạn định trước. check_mrtgtraf plugin (sẵn có trong các bản Nagios plugin được phân phối) cho phép bạn làm điều này. Bạn cũng cần phải cho check_mrtgtraf plugin biết tệp log nào lưu trữ những dữ liệu MRTG , ngưỡng giới hạn, v.v... Trong ví dụ này, chúng ta sẽ giám sát một cổng trên Linksys switch. Tệp MRTG log được lưu trong /var/lib/mrtg/192.168.1.253_1.log. Đây là định nghĩa dịch vụ mà chúng ta sử dụng để giám sát dữ liệu băng thông được lưu trong tệp log.

```
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!1000000,2000000!5000000,5000000!10
    }
```

Trong ví dụ trên, tệp "/var/lib/mrtg/192.168.1.253_1.log" được khai báo trong phần check_local_mrtgtraf chỉ ra rằng plugin sẽ đọc tệp MRTG log khi xử lý. Tùy chọn "AVG" có ý nghĩa là sử dụng số liệu thống kê băng thông trung bình. Tùy chọn "1000000,2000000" là cảnh báo khi tới ngưỡng trong việc tăng tỉ lệ lưu lượng(bằng byte). Tùy chọn "5000000,5000000" là tới hạn lưu lượng gởi đi (bằng byte). Tùy chọn "10" chỉ ra rằng plugin trả về trạng thái CRITICAL nếu tệp MRTG log không được cập nhật sau 10 minute (thường là được cập nhật 5 phút một lần).

Lưu lại tệp, khởi động lại Nagios.

3.2. Giám sát máy đầu cuối

3.2.1. Giám sát các tài nguyên trên máy đầu cuối

Trên mỗi máy tính đầu cuối được cài một Agent. Agent này sẽ thực hiện việc kiểm tra trạng thái các tài nguyên trên chính máy đó. Nagios giao tiếp với Agent này

để thu thập kết quả. NSClient++ là Agent được sử dụng trên máy được giám sát chạy hệ điều hành window và NRPE trên máy được giám sát chạy hệ điều hành linux. Nagios sử dụng 2 plugin để giao tiếp với các Agent này là check_nt cho window và check_nrpe cho linux. Khả năng giám sát:

- Tải CPU.
- Tình trạng sử dụng ổ đĩa cứng.
- Tình trạng sử dụng bộ nhớ trong, và swap.
- Số người dùng đang logon, số tiến trình đang chạy và tệp log hệ thống trên linux Giám sát từng dịch vụ, tiến trình trên window.

Chi tiết cách thức cài đặt, cấu hình tham khảo phần phụ lục.

3.2.2. Giám sát các thông số an toàn phần cứng trên máy đầu cuối

Plugin giám sát các thông số an toàn phần cứng là *check_sensors.* Máy được giám sát phải cài đặt LM sensors và nhân phải được cập nhật module driver phù hợp. Các thông số được giám sát là:

- Nhiệt độ CPU.
- Tốc độ quạt.
- Pin.
- Giờ hệ thống.

Chi tiết cách thức cài đặt, cấu hình tham khảo phần phụ lục.

3.3. Giám sát các dịch vụ mạng

Đối với các dịch vụ mạng như HTTP, POP3, IMAP, FTP, SSH… là các dịch vụ dùng chung, công khai. Nagiso thường giám sát được trạng thái của các dịch vụ này mà không cần bất cứ yêu cầu truy cập đặc biệt nào. Không như các dịch vụ riêng, Nagios không thể giám sát được nếu như không có các agent trung gian. Ví dụ các dịch vụ có liên quan đến host như là tải CPU, tình trạng sử dụng bộ nhớ trong, ổ đĩa, … Vì những thông tin này thường không được công khai với bên ngoài và yêu cầu

quyền truy cập. Khi giám sát các dịch vụ mạng, Nagios sẽ gọi các plugin được đặt ngay trên server Nagios gửi một yêu cầu dịch vụ đến host cung cấp dịch vụ, hoặc gọi một plugin trên một host và yêu cầu dịch vụ trên host thứ 2 rồi thu thập kết quả trả về.

3.3.1. Giám sát web server

3.3.1.1. Tổng quan

Nagios sử dụng plugin *check_http* trong việc giám sát dịch vụ HTTP trên web server. *Check_http* có thể nhận biết được các thông tin sau:

- Thời gian trả lời của web server.
- Mã lỗi trả về của dịch vụ http (403 : không tìm thấy tệp, 404: lỗi xác thực).
- Nội dung chuỗi trả về của http có chứa chuỗi s cho trước không.
- Một URL nào đó có còn nằm trên web server hay không.

3.3.2.2. Cấu hình giám sát

Tất cả các dịch vụ đều được cung cấp bởi một host nào đó. Mọi định nghĩa dịch vụ giám sát đều phải khai báo host cung cấp. Định nghĩa host cung cấp nếu nó chưa được định nghĩa (định nghĩa vào một tệp cấu hình bất kì được khai báo trong tệp cấu hình chính nagios.cfg). Ví dụ định nghĩa một host cung cấp:

```
define host{
    use             generic-host                ; kế thừa giá trị mặc định từ mẫu
    host_name           remotehost   ; Tên của host
    alias           Some Remote Host    ; Tên khác của host
    address         192.168.1.50          ;  địa chỉ IP của host
    hostgroups          allhosts        ; Nhóm của host
    }
```

Định nghĩa một dịch vụ đơn giản cho việc giám sát dịch vụ HTTP trên máy ở xa như sau:

```
define service{
```

```
use             generic-service         ; kế thừa giá trị mặc định từ mẫu
host_name               remotehost
service_description     HTTP
check_command   check_http
}
```

Định nghĩa dịch vụ này sẽ giám sát dịch vụ HTTP chạy trên máy ở xa. Nó sẽ tạo cảnh báo nếu web server không trả lời sau 10 giây hoặc web server trả về mã lỗi(403,404…)

Lưu ý:

Để giám sát ở mức sâu hơn bạn có thể xem hướng dẫn check_http plugin với tham số dòng lệnh là --help. Cú pháp --help có ở tất cả các plugin.

Dưới đây là một định nghĩa dịch vụ ở mức sâu hơn. Nó sẽ kiểm tra xem */download/index.php* URI có chứa chuỗi "latest-version.tar.gz" hay không. Thông báo lỗi nếu không tìm thấy, URI không hợp lệ, hay là web server trả lời sau 5 giây.

```
define service{
    use             generic-service         ; kế thừa giá trị mặc định từ mẫu
    host_name               remotehost
    service_description     Product Download Link
    check_command       check_http!-u   /download/index.php   -t   5   -s   "latest-
version.tar.gz"
    }
```

3.3.2. Giám sát proxy server

Từ cách thức phục vụ của một web proxy, chúng ta có thể sử dụng plugin *check_http* để thực hiện việc truy vấn đến proxy yêu cầu dịch vụ và nhận kết quả trả về.

Giám sát hoạt động bằng cách truy vấn đến proxy, yêu cầu một địa chỉ URL như sau http://www.google.com.vn:

nagios@linux:nagios/libexec$ ./check_http -H www.swobspace.de \

-I 192.168.1.13 -p 3128 -u http://www.swobspace.de

HTTP OK HTTP/1.0 200 OK -2553 bytes in 0.002 seconds

Trong đó –I : là tham số địa chỉ của proxy cần kiểm tra

-H tên của host cung cấp URL cần lấy

-u URL muốn lấy

Định nghĩa lệnh:

```
define command{
command_name check_proxy
command_line $USER1$/check_http -H www.google.de \
-u http://www.google.de -I $HOSTADDRESS$ -p $ARG1$
}
```

Định nghĩa host proxy được giám sát

```
define service{
service_description Webproxy
host_name linux01
check_command check_proxy!3128
...
}
```

3.3.3. Giám sát file server

Khi bạn cần giám sát FTP server, bạn sử dụng check_ftp plugin. Tệp commands.cfg có sẵn định nghĩa lệnh sử dụng plugin này:

```
define command{
    command_name    check_ftp
    command_line    $USER1$/check_ftp -H $HOSTADDRESS$ $ARG1$
    }
```

Dưới đây là định nghĩa dịch vụ đơn giản cho việc giám sát PTF server từ xa:

```
define service{
    use             generic-service         ; kế thừa giá trị mặc định từ mẫu
    host_name               remotehost
    service_description     FTP
```

```
    check_command   check_ftp
    }
```

Định nghĩa dịch vụ này sẽ giám sát dịch vụ PTP và tạo ra thông báo nếu server không trả lời sau 10 giây.

Còn dưới đây là một định nghĩa dịch vụ ở mức sâu hơn. Dịch vụ sẽ kiểm tra FTP server chạy trên cổng 1023 của host ở xa. Nó sẽ tạo ra thông báo nếu server không trả lời sau 5 giây hoặc thông điệp server trả lời không có chuỗi "Pure-FTPd [TLS]".

```
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]"
    }
```

3.3.4. Giám sát mail server

3.3.4.1. Giám sát dịch vụ smtp

check_smtp plugin được sử dụng để giám sát email server. Tệp commands.cfg chứa định nghĩa lệnh sử dụng check_smtp plugin:

```
define command{
    command_name    check_smtp
    command_line    $USER1$/check_smtp -H $HOSTADDRESS$ $ARG1$
    }
```

Dưới đây là định nghĩa dịch vụ đơn giản cho việc giám sát SMTP server

```
define service{
    use             generic-service         ; kế thừa giá trị mặc định từ mẫu
    host_name               remotehost
    service_description     SMTP
    check_command   check_smtp
    }
```

Định nghĩa dịch vụ này sẽ giám sát dịch vụ SMTP server và tạo ra thông báo nếu SMTP server không trả lời sau 10 giây.

Còn định nghĩa dưới đây sẽ kiểm tra SMTP server và tạo ra thông báo nếu server không trả lời sau 5 giây và thông điệp trả về từ server không chứa đoạn "mygreatmailserver.com".

```
define service{
 use             generic-service         ; kế thừa giá trị mặc định từ mẫu
    host_name               remotehost
    service_description     SMTP Response Check
    check_command   check_smtp!-t 5 -e "mygreatmailserver.com"
    }
```

3.3.4.2. Giám sát dịch vụ POP3

check_pop plugin được sử dụng để giám sát dịch vụ POP3 trên mail server . Tệp commands.cfg chứa định nghĩa lệnh sử dụng check_pop plugin:

```
define command{
    command_name    check_pop
    command_line    $USER1$/check_pop -H $HOSTADDRESS$ $ARG1$
    }
```

Dưới đây là định nghĩa dịch vụ đơn giản cho việc giám sát dịch vụ POP3 trên host ở xa:

```
define service{
    use             generic-service         ; kế thừa giá trị mặc định từ mẫu
    host_name               remotehost
    service_description     POP3
    check_command   check_pop
    }
```

Định nghĩa dịch vụ này sẽ giám sát dịch vụ POP3 và tạo ra thông báo nếu POP3 không trả lời sau 10 giây.

Còn định nghĩa dưới đây sẽ kiểm tra dịch vụ POP3 và tạo ra thông báo nếu server không trả lời sau 5 giây và thông điệp trả về từ server không chứa đoạn "mygreatmailserver.com".

```
define service{
```

```
use             generic-service         ; kế thừa giá trị mặc định từ mẫu
host_name               remotehost
service_description     POP3 Response Check
check_command   check_pop!-t 5 -e "mygreatmailserver.com"
}
```

3.3.4.3. Giám sát dịch vụ IMAP

check_imap plugin được sử dụng để giám sát dịch vụ IMAP4 trên mail server . Tệp commands.cfg chứa định nghĩa lệnh sử dụng check_imap plugin:

```
define command{
    command_name    check_imap
    command_line    $USER1$/check_imap -H $HOSTADDRESS$ $ARG1$
    }
```

Dưới đây là định nghĩa dịch vụ đơn giản cho việc giám sát dịch vụ IMAP4 trên host ở xa:

```
define service{
    use             generic-service         ; kế thừa giá trị mặc định từ mẫu
    host_name               remotehost
    service_description     IMAP
    check_command   check_imap
    }
```

Định nghĩa dịch vụ này sẽ giám sát dịch vụ IMAP4 và tạo ra thông báo nếu IMAP4 không trả lời sau 10 giây.

Còn định nghĩa dưới đây sẽ kiểm tra dịch vụ IMAP4 và tạo ra thông báo nếu server không trả lời sau 5 giây và thông điệp trả về từ server không chứa đoạn "mygreatmailserver.com".

```
define service{
    use             generic-service         ; kế thừa giá trị mặc định từ mẫu
    host_name               remotehost
    service_description     IMAP4 Response Check
    check_command   check_imap!-t 5 -e "mygreatmailserver.com"
    }
```

Khởi động lại Nagios. Chú ý là mỗi lần bạn thêm một định nghĩa dịch vụ mới vào tệpc cấu hình thì bạn phải kiểm chứng lại tệp đó, và khởi động lại Nagios. Nếu quá trình kiểm chứng có lỗi thì phải cấu hình lại cho đúng đến khi không còn lỗi thì mới khởi động lại Nagios.

3.3.5. Giám sát Các dịch vụ khác

Ngoài những dịch vụ trên Nagios còn sẵn có plugin cung cấp việc giám sát các dịch vụ: SSH, LDAP, DHCP, DNS, database, cổng TCP, cổng UDP… Phần cài đặt và định nghĩa các dịch vụ này có thể tham khảo ở phần phụ lục.

3.4. Cảnh báo cho người quản trị

Không phải lúc nào người quản trị cũng có thể dõi theo và nắm bắt mọi hoạt động của mạng qua giao diện của hệ thống giám sát. Bởi vậy bất cứ hệ thống giám sát mạng nào cũng cần cung cấp chức năng thông báo cho người quản trị qua các phương tiện truyền tin như email, sms, IM… Nagios cung cấp một hệ thống thông báo linh hoạt và qua nhiều phương tiện truyền tin khác nhau. Trong nagios, thông báo sảy ra khi host/dịch vụ thay đổi từ trạng thái này sang trạng thái khác. Tuy nhiên không phải bất cứ host/dịch vụ nào cũng có thể nhận thông báo. Thông báo trước khi đến được các liên lạc nó phải qua nhiều bộ lọc khác nhau. Khi một sự kiện sảy ra với một host/dịch vụ nào đó thì trước khi quyết định ra một thông báo cho người quản trị, Nagios sẽ thực hiện kiểm tra:

- Cấu hình của Nagios có cho phép gửi thông báo hay không.(tùy chọn enable_notifications)

- Host/dịch vụ được kiểm tra có trong thời gian được lập lịch ngừng hoạt động không (downtime). Nếu host/dịch vụ đang trong thời gian downtime thì cách hành động kiểm tra host/dịch vụ đó vẫn được thực thi. Kết quả được lưu lại trong Nagios còn thông báo thì sẽ không được gửi đi.

- Host/dịch vụ được kiểm tra có đang bị Flapping không (nếu cấu hình bật tùy chọn phát hiện flap). Chi tiết về tình trạng Flapping có trong chương 5.

- Từng host/dịch vụ có thể được cấu hình để chỉ thông báo cho người quản trị một số tình trạng nhất định.

- Mỗi host/dịch vụ có thể được định nghĩa một chu kì thời gian cho thông báo. Nếu khoảng thời gian thông báo được tạo không nằm trong giới hạn này thì thông báo cũng bị loại.

- Thời gian từ lần thông báo trước đến thời điểm hiện tại kiểm tra đã lớn hơn khoảng thời gian được khai báo trong tùy chọn <notification_interval> hay chưa. Đây là tùy chọn cấu hình khoảng thời gian giữa hai lần thông báo kề nhau.

- Cuối cùng Nagios kiểm tra cấu hình xem những người dùng Nagios nào được nhận thông báo về tình trạng của host/dịch vụ đang được kiểm tra .

Chi tiết cấu hình gửi thông báo có trong phần phụ lục của tài liệu.

3.5. Tổng hợp báo cáo

Ngoài chức năng giám sát và cảnh báo các trạng thái hiện thời của các thành phần mạng Nagios còn có thể lập báo cáo về tình trạng hoạt động của các thành phần mạng trong một khoảng thời gian nhất định. Báo cáo có thể được lập với từng host/dịch vụ, từng nhóm hoặc toàn bộ mạng với các bộ lọc trạng thái(SORT/HARD), tình trạng(OK, WARNING, CRITICAL, UNKNOWN). Từ các số liệu trong báo cáo người quản trị nắm được tình trạng hoạt động của các thành phần mạng trong một khoảng thời gian nhất định, đánh giá được độ ổn định của các thành phần mạng. Việc tổng hợp báo cáo được thực hiện khá đơn giản qua giao diện web.

Chương 4. Các vấn đề liên quan

4.1. Các khái niệm cơ bản trong Nagios

4.1.1. Kiểm tra host

Host được kiểm tra bởi Nagios daemon khi:

* Trong khoảng thời gian được định nghĩa trong tùy chọn check_interval (khoảng thời gian giữa hai lần kiểm tra kế tiếp) và retry_interval (Khoảng thời gian thực hiện kiểm tra lại để xác nhận khi phát hiện host thay đổi trạng thái) của định nghĩa cấu hình host.

* Khi dịch vụ mà host đó cung cấp thay đổi trạng thái. Thường thì khi dịch vụ thay đổi trạng thái thì host cũng thay đổi trạng thái. Ví dụ nếu Nagios phát hiện ra dịch vụ HTTP thay đổi trạng thái từ CRITICAL sang OK, thì rất có thể là host cung cấp dịch vụ này vừa khởi động lại và đang chạy.

* Khi có host con của host đó bị đặt vào trạng thái UNREARCHABLE. Đó là trạng thái mà Nagios không liên lạc được với host đó hay có thể hiểu là mất đường truyền đến host đó.

Nagios phân loại host ra ba trạng thái:

* UP : hoạt động bình thường.

* DOWN: tạm dừng hoạt động.

* UNREACHABLE: không tìm thấy.

4.1.2. Kiểm tra dịch vụ

Dịch vụ được kiểm tra bởi Nagios daemon khi:

* Trong khoảng thời gian được định nghĩa trong tùy chọn check_interval (khoảng thời gian giữa hai lần kiểm tra kế tiếp) và retry_interval (Khoảng thời gian thực hiện kiểm tra lại để xác nhận khi phát hiện dịch vụ thay đổi trạng thái) của định nghĩa cấu hình dịch vụ.

Nagios phân loại dịch vụ thành bốn trạng thái:

- OK: Hoạt động bình thường.
- WARNING: Có thể hoạt động nhưng chưa chính xác hoặc có thể không hoạt động.
- UNKNOWN: Không xác định được.
- CRITICAL: Không hoạt động.

4.1.3. Khái niệm trạng thái SORT/HARD

Ví dụ ta có một định nghĩa kiểm tra dịch vụ DNS như sau

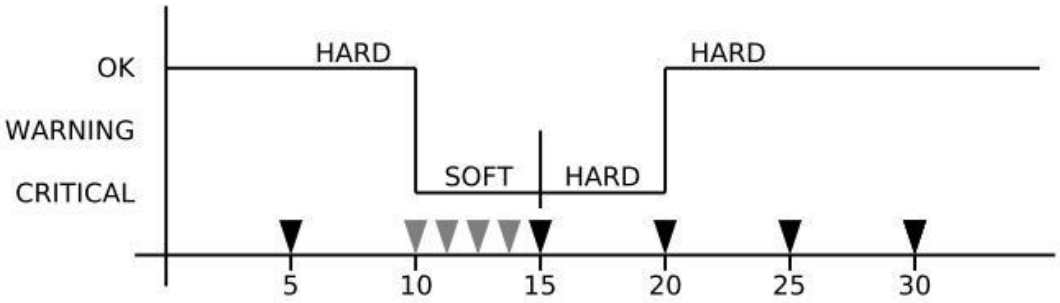
```
define service{
host_name proxy
service_description DNS
...
normal_check_interval 5
retry_check_interval 1
max_check_attempts 5
...
}
```

Trong đó

normal_check_interval: khoảng thời gian giữa các lần kiểm tra bình thường(là 5 phút).

retry_check_interval: nếu gặp lỗi, sau 1 phút kiểm tra lại để xác nhận (soft state).

max_check_attempts: thực hiện kiểm tra lại 5 lần, nếu lỗi vẫn sảy ra. Nagios kết luận chắc chắn dịch vụ thay đổi trạng thái (hard state).

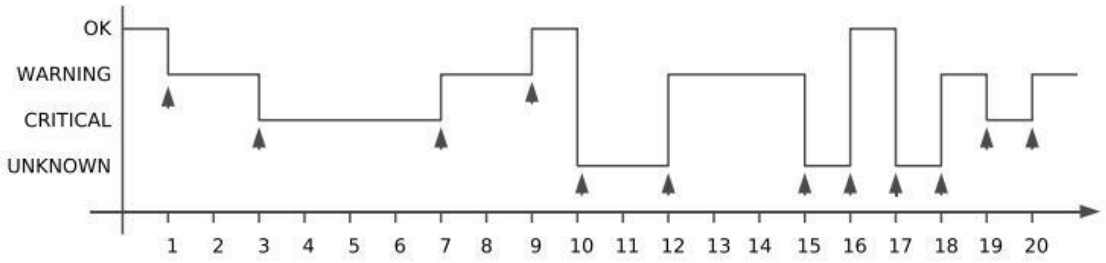


- Vậy khi Nagios chắc chắn về trạng thái của một host/dịch vụ thì nó đặt là HARD STATE. Thông thường khi bắt đầu phát hiện host/dịch vụ thay đổi trạng thái Nagios thực hiện lại vài lần kiểm tra để xác nhận, tùy vào cấu hình. Trong khoảng thời gian đó host/dịch vụ được đặt là SOFT STATE. Khi host/dịch vụ được đặt vào tình trạng SOFT STATE hoặc khi nó khôi phục lại trạng thái cũ từ tình trạng SOFT STATE thì không có bất cứ thông báo nào được gửi. Tuy nhiên những sự kiện này vẫn được ghi vào tệp log và có thể xem được qua giao diện chương trình.

4.1.4. Khái niệm FLAP

Nếu trạng thái của host/ dịch vụ không ổn định, thay đổi liên tục. Người quản trị sẽ nhận được rất nhiều thông báo trong một khoảng thời gian ngắn. Nó không chỉ gây khó chịu mà còn làm rối loạn việc xác định vấn đề lỗi. Nagios có thể phát hiện vấn đề này và đặt trạng thái đó là flapping.

Để phát hiện tình trạng flap của một dịch vụ, Nagios lưu lại 21 kết quả kiểm tra dịch vụ gần nhất, tức là tối đa lưu lại 20 lần thay đổi trạng thái của dịch vụ. Hình dưới đây mô tả sự thay đổi trạng thái của một dịch vụ:



Từ hình trên ta có thể thấy là trong 20 lần kiểm tra, dịch vụ thay đổi trạng thái 12 lần. Nagios dựa vào số liệu này để thông báo dịch vụ đang rơi vào tình trạng flapping hoặc thoát khỏi tình trạng flapping. Khi flapping sảy ra, Nagios sẽ ghi sự kiện này vào tệp log, đặt thông tin flap vào phần comment của dịch vụ và dừng hành động thông báo trạng thái dịch vụ.

Phát hiện flap được cấu hình ở 2 vị trí; tệp cấu hình chính nagios.cfg (cài đặt cấu hình nói chung) và trong định nghĩa của từng dịch vụ cụ thể.

Trong tệp cấu hình chính:

```
#/etc/nagios/nagios.cfg
...
enable_flap_detection=1  // cho phép phát hiện flap.
low_service_flap_threshold=5.0   //ngưỡng dưới flap
high_service_flap_threshold=20.0          //ngưỡng trên flap
...
```

Đoạn cấu hình trên có nghĩa là nếu có từ 5 lần trở lên dịch vụ được ghi nhận là thay đổi trong 20 lần kiểm tra thì dịch vụ đó được đặt vào tình trạng flapping.

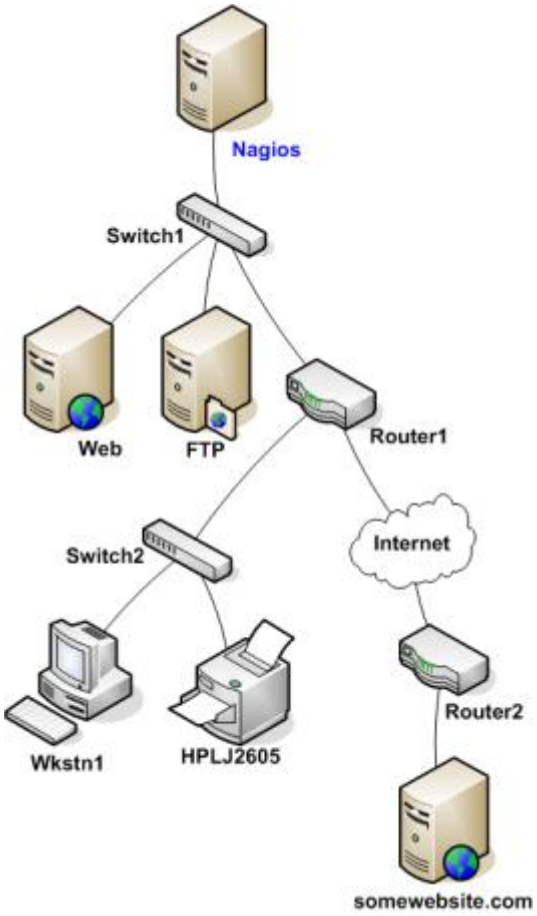
Chúng ta có thể thiết đặt tùy chọn phát hiện flapping và đặt ngưỡng flap cho từng dịch vụ trong phần định nghĩa đối tượng dịch vụ.

```
define service{
host_name linux01
…
flap_detection_enabled 1
low_flap_threshold 6.0
high_flap_threshold 20.0
...
}
```

Tương tự phát hiện flapping đối với host.

4.1.5. Mối quan hệ cha/con giữa các host và phân biệt trạng thái down/unrearchable

Nagios là phần mềm chưa có khả năng tự phát hiện ra các node và kiến trúc của mạng. Công việc này do người dùng tự định nghĩa và quyết định theo quy tắc nhất định. Nagios được coi là trung tâm giám sát. Các thiết bị(A) có đường kết nối vật lý trực tiếp đến server Nagios được có mối quan hệ là con của Nagios. Các thiết bị kết nối trực tiếp đến A được coi là con của A.. Cứ như vậy kiến trúc mạng được định nghĩa và mở rộng qua mối quan hệ cha/con này, với Nagios là trung tâm.



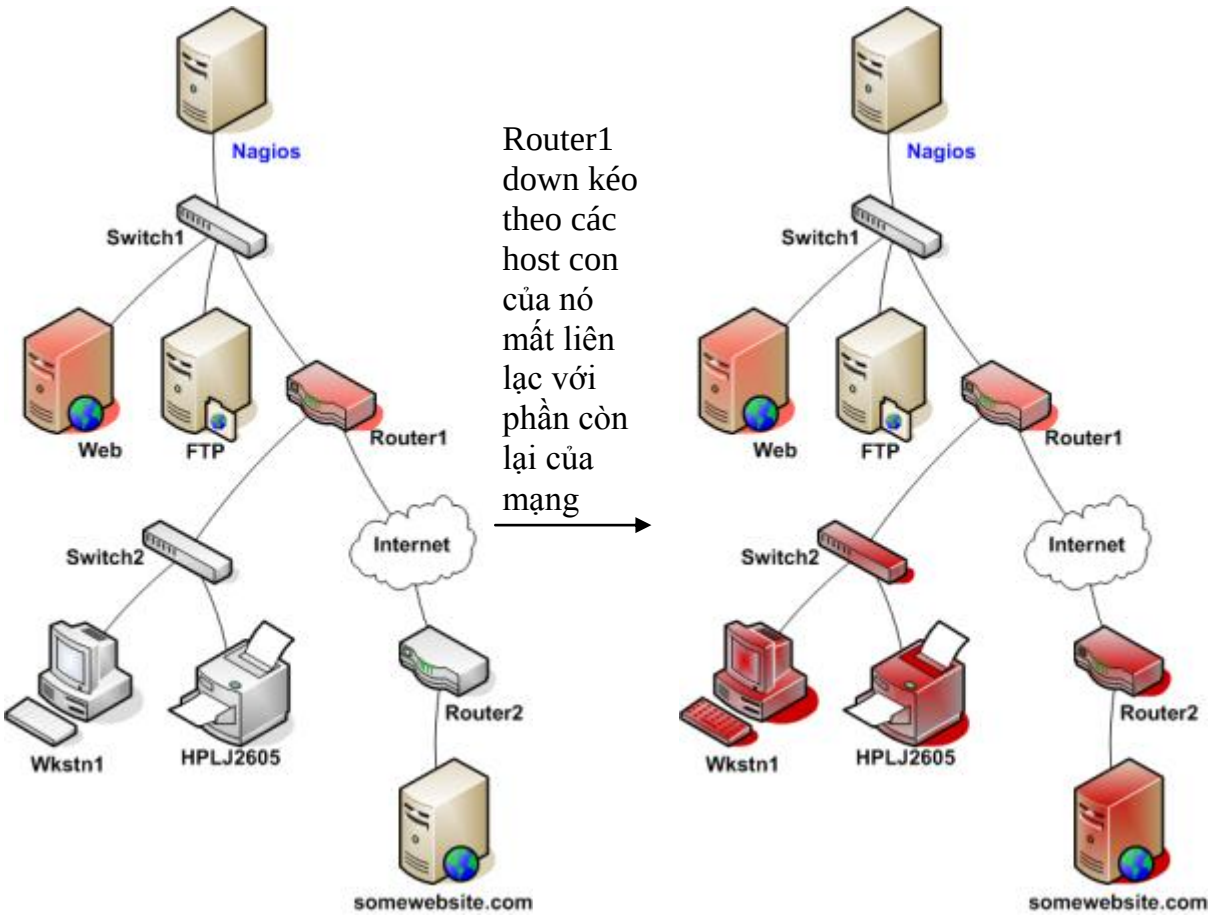
Hình 4.1 Mối quan hệ host cha/con.

Ví dụ mạng có kiến trúc như trên. Khi đó ta có Switch1 được coi là con của Nagios. Web, FTP, Router1 là con của Switch1, Switch2 được coi là con của Router1 … Tất cả mối quan hệ này đều phải do người dùng định nghĩa qua tùy chọn parents trong mỗi định nghĩa đối tượng. ví dụ:

```
define host{
        host_name
        …
        }

define host{
        host_name                       Switch1
        …
        parents                 Nagios
```

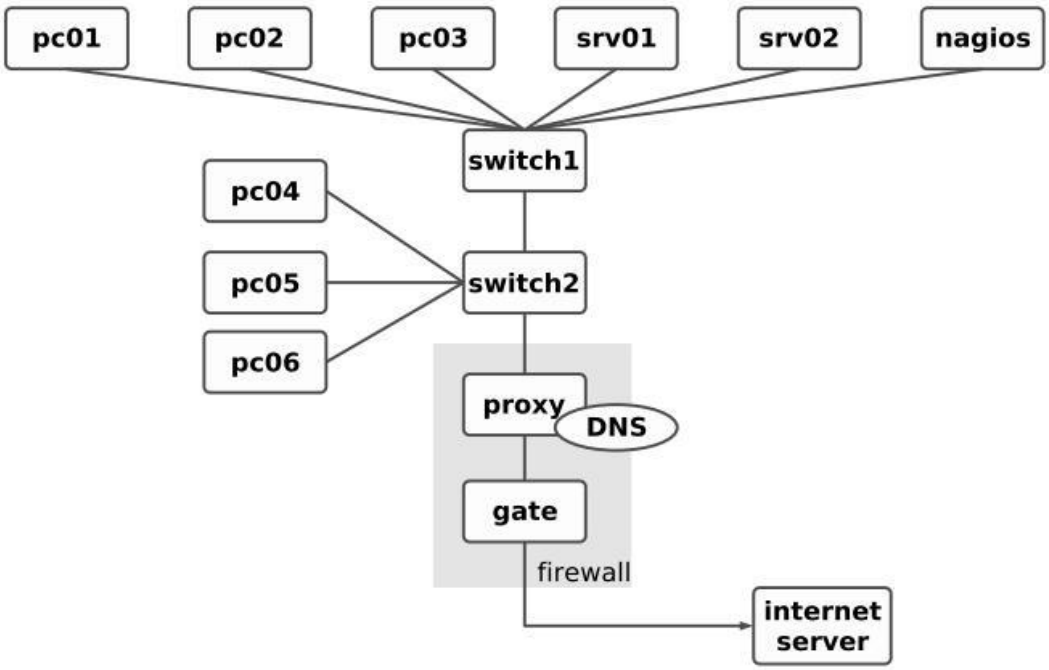
```
}

define host{
        host_name                       Web
        …
        parents                 Switch1
        }
```

Như ví dụ hình bên dưới, ta tắt host web và router1. Một hành động kiểm tra được thực hiện và trả về kết quả cho Nagios. Trường hợp này Nagios kết luận host web và router1 ở trạng thái DOWN bởi vì host cha Switch1 hoạt động bình thường. Trong khi đó các host nằm sau router1 được kết luận là UNREACHABLE<Không xác định>. Vì Nagios không thể liên lạc được với chúng vì router1 bị tắt kéo theo mất đường kết nối đến các host này.

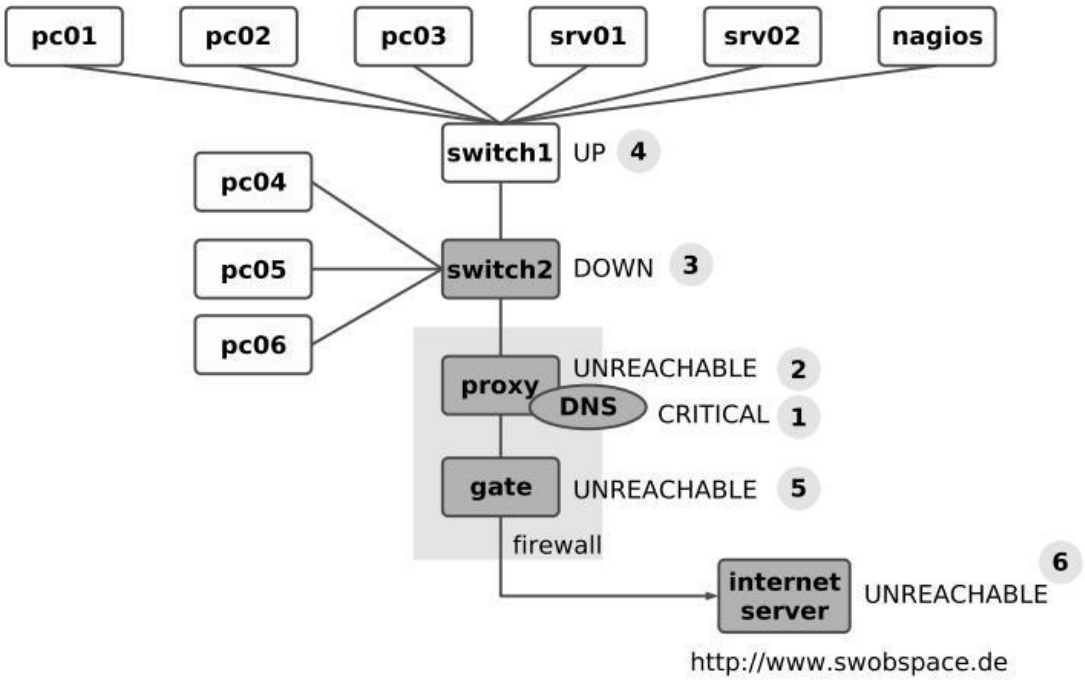


Hình 4.2 Phân biệt DOWN-UNREACHABLE.

Việc phân biệt trạng thái DOWN-UNREACHABLE của host giúp các nhà quản trị dễ dàng hơn trong việc xác định được nguyên nhân và vị trí của lỗi sảy ra trên mạng khi nhận được thông báo sự cố. Ta xét một ví dụ như sau: Khi giám sát dịch vụ DNS trên một mạng được định nghĩa như hình 4.2. Giả sử tình huống khi Nagios phát hiện DNS không trả lời truy vấn của nó. Nó thực hiện kiểm tra host cung cấp dịch vụ DNS(ở đây là proxy). Proxy không trả lời. Host cha của proxy là switch2 được kiểm tra. Switch2 không trả lời. Host cha của Switch2 là switch1 được kiểm tra. Switch1 trả lời. Từ đó Nagios kết luận Switch1 UP. Con của nó là switch2 DOWN. Con của switch bị DOWN là UNREARCHABLE. DNS không hoạt động : CRITICAL. Kết luận như hình 4.3



Hình 4.3 Ví dụ Xác định lỗi 1.



Hình 4.4 Ví dụ xác định lỗi 2.

Vậy trong trường hợp này khi khắc phục sự cố DNS, người quản trị đã xác định được ngay nguyên nhân đầu tiên dẫn đến sự cố là do switch2 bị DOWN.

4.1.6. Lập lịch downtime

Có những thiết bị chỉ hoạt động vào những khoảng thời gian nhất định trong ngày và ngoài khoảng thời gian đó nó được tắt đi. Hành động tắt bật được thực hiện có tính chu kỳ và thường xuyên. Ví dụ như thiết bị văn phòng, máy in … Hoặc có những server cần dừng hoạt động, nâng cấp, sửa chữa. Tóm lại là trong thực tế có nhiều trường hợp trạng thái của thiết bị mạng thay đổi do sự chủ động từ phía người quản trị hoặc người quản trị có thể kiểm soát được. Với những trường hợp này việc gửi cảnh báo cho người quản trị là không cần thiết. Vì thế Nagios cho phép người quản trị lập lịch thời gian ngừng kiểm tra cho từng host/dịch vụ. Khoảng thời gian này được gọi là downtime. Trong khoảng thời gian này không có bất cứ thông báo nào của host/dịch vụ được lập lịch được gửi đi. Việc lập lịch downtime cho host/dịch vụ khá đơn giản và được thực hiện ngay trên giao diện web của chương trình.

4.2. Bộ xử lý sự kiện

Khi trạng thái của host/dịch vụ thay đổi, nagios có thể chạy một chương trình bất kì được định sẵn với bộ xử lý sự kiện (event handler) để xử lý tình huống mà không cần sự can thiệp của người quản trị.

4.2.1. Thời gian chạy bộ xử lý sự kiện

Khi host/dịch vụ :

- ở trong trạng thái mềm
- bắt đầu vào trạng thái cứng
- Bắt đầu khôi phục lại bình thường từ trạng lỗi(mềm hoặc cứng)

4.2.2. Ví dụ event handler

Ví dụ định nghĩa một script khởi động lại máy in khi máy in bắt đầu rơi vào trạng thái lỗi và kiểm tra lại đến lần thứ 3 tình trạng lỗi vẫn sảy ra.

Sửa định nghĩa dịch vụ giám sát máy in, khai báo lệnh xử lý sự kiện restart-lpd

```
define service{
host_name printserver
service_description LPD
...
event_handler restart-lpd
...
}
```

Định nghĩa trong tệp lệnh lệnh restart-lpd:

```
define command{
command_name restart-lpd
command_line $USER1$/eventhandler/restart-lpd.sh \
$SERVICESTATE$ $SERVICESTATETYPE$ $SERVICEATTEMPT$
}
```

Lệnh này sẽ gọi một script có tên là *restart-lpd.sh* đặt trong thư mục /usr/local/nagios/libexec/eventhandler (thông thường các script được đặt trong thư mục

/usr/local/nagios/libexec/). Script này nhận 3 macro làm tham số đó là trạng thái hiện thời của dịch vụ $SERVICESTATE$ (OK,WARNING, CRITICAL, hoặc UNKNOWN), loại trạng thái $SERVICESTATETYPE$ (mềm hoặc cứng), và số lần kiểm tra lại hiện thời $SERVICEATTEMPT$). Đối với host thì các macro này là $HOSTSTATE$, $HOSTSTATETYPE$, và $HOSTATTEMPT$.

4.2.3. Script xử lý

```
#!/bin/bash
#/usr/local/nagios/libexec/eventhandlers/restart-lpd.sh
#$1= Status, $2 =status type, $3 =attempt
case $1 in
OK)
;;
WARNING)
;;
CRITICAL)
if [$2=="HARD" ]||[[$2=="SOFT" && $3 -eq 3]]; then
echo "Restarting lpd service"
/usr/bin/sudo /etc/init.d/lpd restart
fi
;;
UNKNOWN)
;;
esac
exit 0
```

Với script này nếu trạng thái dịch vụ là critical, loại trạng thái là HARD hoặc loại trạng thái là SOFT và đã kiểm tra lại đến lần thứ 3 thì dịch vụ lpd được gọi với tham số là restart. Script này được thực thi với quyền của người dùng Nagios (có thể không có quyền tạm dừng hoặc khởi động lại dịch vụ hệ thống). Vì vậy phải sử dụng lệnh sudo để dùng quyền root khởi động lại dịch vụ lpd.

Nếu như bạn muốn người dùng nagios có quyền với dịch vụ lpd thì thực hiện như sau:

linux:~ # visudo

Thêm dòng sau và tệp cấu hình

nagios nagsrv=(root)NOPASSWD: /etc/init.d/lpd

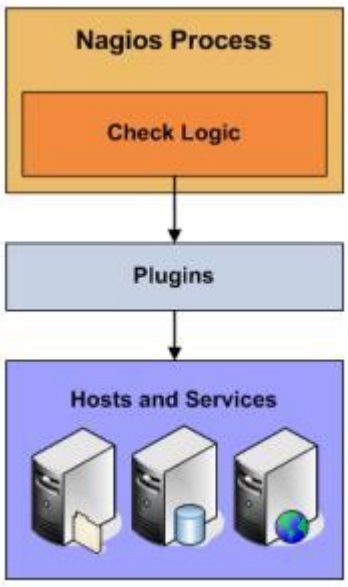
Dòng này cho phép người dùng nagios có quyền chạy lệnh /etc/init.d/lpd trên host nagsrv và không cần mật khẩu.

Nếu bạn khởi động lại dịch vụ khi nó đang ở trạng thái mềm thì người quản trị sẽ không nhận được bất kì thông báo nào. Tuy nhiên sự kiện vẫn được ghi lại vào tệp log.

4.3. Giám sát phân tán

4.3.1. Kiểm tra chủ động

Khi Nagios cần kiểm tra trạng thái của host/dịch vụ nó gọi một plugin thực hiện hành động kiểm tra đó. Nagios sẽ nhận kết quả từ plugin được gọi. Đây là cách thức kiểm tra cơ bản của Nagios. Trong trường hợp này Nagios quyết định khi nào hành động kiểm tra được thực hiện. Kiểm tra chủ động là phương thức kiểm tra cơ bản và được sử dụng nhiều. Tuy nhiên nó vẫn còn có một số nhược điểm ví dụ như nó có thời gian timeout của mỗi hành động kiểm tra là được định trước. Trong một số trường hợp thời gian đó có thể không đủ để có kết quả chính xác.



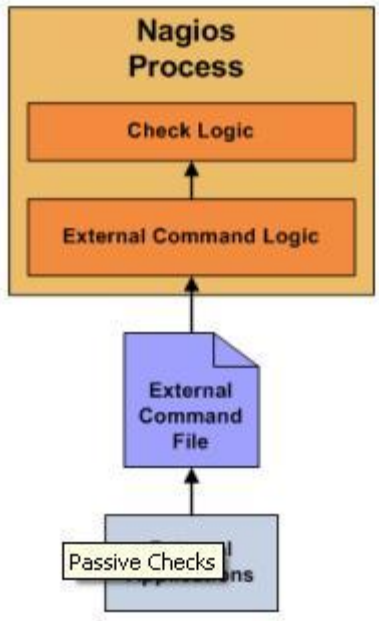
Hình 4.5 Kiểm tra chủ động

4.3.2. Kiểm tra bị động

Hành động kiểm tra host/dịch vụ được thực hiện bởi một ứng dụng/tiến trình bên ngoài. Kết quả kiểm tra sẽ được gửi về cho Nagios xử lý.

Kiểm tra bị động được dùng khi giám sát ở các vùng mạng khác nhau có tường lửa ngăn cách, dùng trong giám sát phân tán…

Cách thức kiểm tra bị động: Một ứng dụng bên ngoài thực hiện hành động kiểm tra host/dịch vụ. Kết quả đó được ghi ra tệp lệnh ngoại trú. Nagios định kỳ lấy kết quả kiểm tra từ tệp này và xử lý.



Hình 4.6 Kiểm tra bị động

4.3.3. Giám sát phân tán

Nagios có thể phân tán việc giám sát trên những mạng lớn bằng cách sử dụng một server trung tâm và nhiều server con phân tán trên các mạng con(mạng con ở đây có thể là một mạng WAN, các vùng mạng ngăn cách bởi tường lửa…). Nagios gọi mỗi mạng con này là một "cluster".

Server trung tâm: nhận và xử lý kết quả kiểm tra từ server con phân tán (passive check), các kết quả tự nó kiểm tra qua plugin (active check).

Server con phân tán: thực hiện các kiểm tra chủ động (active check) host/dịch vụ và gửi kết quả về cho server trung tâm. Thường thì với mỗi cluster nên đặt một server con Nagios.

Server con gửi kết quả kiểm tra về server trung tâm bằng một plugin có tên NSCA. Chi tiết cách thức cài đặt, giao tiếp của NSCA tham khảo phần phụ lục cuối tài liệu

TÀI LIỆU THAM KHẢO

[1] Max Schubert, Nagios 3 Enterprise Network Monitoring Including Plug-Ins and Hardware Devices, May 2008

[2] Wojciech Kocjan, Learning Nagios 3.0, Packt Publishing, October 2008

[3] Wolfgang Barth, Nagios System and Network Monitoring, No Starch Press, 2006

[4] http://nagios.org/

[5] http://community.nagios.org/

[6] http://www.monitoringexchange.org

[7] http://www.nagioswiki.org