Monitoring with Nagios

Introduction

This article will describe how to use Nagios Core to monitor critical services on sipXcom or openUC.

Prerequisites

You'll need a Nagios Core installation and administrative access to your sipXcom servers and network. For this article I have compiled Nagios Core from source using the default settings rather than using a OS package. The path to files may vary if you have installed via rpm or apt. I will use the Nagios Remote Plugin Executor (NRPE) as a means to aggregate the checks, but there are alternatives available if NRPE doesn't suit your needs. Most of the checks used are available from the standard Nagios Plugins. If you want to expand on these there are many more available from the Nagios Exchange. If you write your own checks I encourage you to share them with the community so others may benefit.

Overview

If compiled from source using the defaults, Nagios Core will install to /usr/local/nagios :

```
# tree --charset=ASCII -d nagios/
nagios/
|-- bin
|-- etc
|   `-- objects
|-- libexec
|-- sbin
|-- share
|   |-- contexthelp
|   |-- docs
|   |   `-- images
|   |-- images
|   |   `-- logos
|   |-- includes
|   |   `-- rss
|   |       `-- extlib
|   |-- js
|   |-- media
|   |-- ssi
|   `-- stylesheets
`-- var
    |-- archives
    |-- rw
    `-- spool
        `-- checkresults
```

The etc/objects directory is where your host configuration files are stored. I recommend organizing hosts into groups beneath the objects directory, then grouping similar services beneath that. For example, if you have a group of three openUC servers at example.org :

```
$ mkdir /usr/local/nagios/etc/objects/example.org
$ mkdir /usr/local/nagios/etc/objects/example.org/openuc
$ touch /usr/local/nagios/etc/objects/example.org/openuc/uc1.cfg
$ touch /usr/local/nagios/etc/objects/example.org/openuc/uc2.cfg
$ touch /usr/local/nagios/etc/objects/example.org/openuc/uc3.cfg
```

By structuring in this way the system administrator can quickly understand who it belongs to and what it does. This is also especially helpful if you intend on running Nagios in a multi tenant fashion.

Preparing a host for monitoring

Before stepping into the uc1.cfg configuration on the Nagios server we'll need to prepare the sipXcom or openUC host(s) for our checks. Nagios Remote Plugin Executor (NRPE) basically it works as an aggregate point for multiple check scripts :

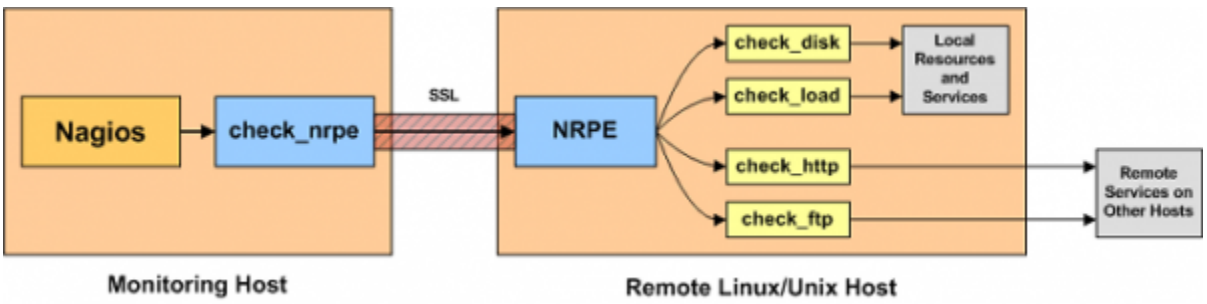

Image source : https://exchange.nagios.org/directory/image/93

You'll need to download and install both NRPE and the standard Nagios plugins on each host you intend on monitoring. After installing these you may wish to pause for a moment and review the check scripts now available under /usr/local/nagios/libexec. The NRPE configuration, /usr/local/nagios/etc/nrpe.cfg, was likely copied from the sample provided within the NRPE tarball. You should review this file for any environmental changes you may need to make such as partition locations :

```
command[check_hda1]=/usr/local/nagios/libexec/check_disk -w 20% -c 10% -p /dev/hda1
command[check_zombie_procs]=/usr/local/nagios/libexec/check_procs -w 5 -c 10 -s Z
command[check_total_procs]=/usr/local/nagios/libexec/check_procs -w 150 -c 200
```

The commands defined should match what is being called within the host configuration file. For example, checks for uc1.example.org are defined on the nagios server in /usr/local/nagios/etc/objects/example.org/openuc/uc1.cfg :

```
define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             Check Users
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_users
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             Check Swap
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_swap

        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             Check Load
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_load
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             Check Boot Partition
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_boot
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             Check Root Partition
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_root
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             Check Zombie Processes
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_zombie_procs
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             Check Total Processes
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_total_procs
        }
```

The check_command line is basically saying "connect with NRPE and run xxx". Be sure that xxx is a defined within /usr/local/nagios/etc/nrpe.cfg of the host you are checking against. For things you want to execute from the Nagios server, make certain that you've defined those commands in the Nagios server /usr/local/nagios/etc/objects/commands.cfg. For example I defined the SSL certificate check on my Nagios server command.cfg .. :

```
define command {

        command_name    check_ssl_certificate

        command_line    $USER1$/check_ssl_certificate -H $HOSTADDRESS$ -c 3 -w 7

        }
```

But in /usr/local/nagios/etc/objects/example.org/uc1.cfg, this is defined without the check_nrpe prefix so it will execute from the Nagios server rather than on the uc1.example.org host ..

```
define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             SSL Certificate Expiration
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_ssl_certificate
        }
```

sipXcom / openUC services

Below are additional examples for uc1.example.org that pertain to sipXcom/openUC services. These would be defined in /usr/local/nagios/etc/objects /example.org/openuc/uc1.cfg on our Nagios server.

```
define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             NTP
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_ntp_time!0.5!1
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             Check SIP Registration
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_sip_registration
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             SSH
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_ssh!-p 22
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             TFTP
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_tftp
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             FTP
```

```
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_ftp!-H uc1.example.org -p 21
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             Check openUC Web UI
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_ui
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             Check XMPP
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_jabber
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             TCP SIP SRV
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_tcp_sip_srv
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             UDP SIP SRV
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_udp_sip_srv
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             TCP SIPS SRV
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_tcp_sips_srv
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             SIP TLS SRV
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_sip_tls_srv
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             SIP RR SRV
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_sip_rr_srv
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             SIP MWI SRV
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_sip_mwi_srv
        }
```

```
define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             XMPP client SRV
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_xmpp_client_srv
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             XMPP server SRV
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_xmpp_server_srv
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             XMPP conference server SRV
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_xmpp_conf_srv
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             TCP Voicemail SRV
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_tcp_vm_srv
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             Check SIPXCONFIG
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_sipxconfig
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             Check SIPXCDR
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_sipxcdr
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             Check MySQL homer.db
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_homer
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             MongoDB Connection Check
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_mongo_connect
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             MongoDB Long running ops
        contact_groups                  admins
```

```
        notifications_enabled           1
        check_command                   check_nrpe!check_mongo_lag
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             MongoDB Operations Count
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_nrpe!check_mongo_ops
        }

define service{
        use                             generic-service
        host_name                       uc1.example.org
        service_description             SSL Certificate Expiration
        contact_groups                  admins
        notifications_enabled           1
        check_command                   check_ssl_certificate
        }
```

The command definitions for all commands prefixed with check_nrpe should be defined on uc1.example.org within /usr/local/nagios/etc/nrpe.cfg, for example :

```
# system checks
command[check_users]=/usr/local/nagios/libexec/check_users -w 5 -c 10
command[check_load]=/usr/local/nagios/libexec/check_load -w 15,10,5 -c 30,25,20
command[check_root]=/usr/local/nagios/libexec/check_disk -w 20% -c 10% -p /dev/mapper/vg_root-lv_root
command[check_boot]=/usr/local/nagios/libexec/check_disk -w 20% -c 10% -p /dev/vda1
command[check_zombie_procs]=/usr/local/nagios/libexec/check_procs -w 5 -c 10 -s Z
command[check_total_procs]=/usr/local/nagios/libexec/check_procs -w 200 -c 250
command[check_swap]=/usr/local/nagios/libexec/check_swap -w 80% -c 50%
command[check_memory]=/usr/local/nagios/libexec/check_memory.pl

# sipx service checks
command[check_sipxconfig]=/usr/local/nagios/libexec/check_postgres.pl -db SIPXCONFIG --action connection
command[check_sipxcdr]=/usr/local/nagios/libexec/check_postgres.pl -db SIPXCDR --action connection
command[check_ui]=/usr/local/nagios/libexec/check_http -w5 -c 10 --ssl -H uc1.example.org -u /sipxconfig/app
command[check_sip_registration]=/usr/local/nagios/libexec/check_registrations.sh
command[check_ntp_time]=/usr/local/nagios/libexec/check_ntp_time -H uc1.example.org -w 0.5 -c 1
command[check_mongo_connect]=/usr/bin/python /usr/local/nagios/libexec/check_mongo -H uc1.example.org -A connect
command[check_mongo_ops]=/usr/bin/python /usr/local/nagios/libexec/check_mongo -H uc1.example.org -A count
command[check_mongo_lag]=/usr/bin/python /usr/local/nagios/libexec/check_mongo -H uc1.example.org -A long

# dns checks
command[check_tcp_sip_srv]=/usr/local/nagios/libexec/check_dns -H _sip._tcp.example.org -s 127.0.0.1 -q SRV
command[check_udp_sip_srv]=/usr/local/nagios/libexec/check_dns -H _sip._udp.example.org -s 127.0.0.1 -q SRV
command[check_tcp_sips_srv]=/usr/local/nagios/libexec/check_dns -H _sips._tcp.example.org -s 127.0.0.1 -q SRV
command[check_sip_tls_srv]=/usr/local/nagios/libexec/check_dns -H _sip._tls.example.org -s 127.0.0.1 -q SRV
command[check_sip_mwi_srv]=/usr/local/nagios/libexec/check_dns -H _sip._tcp.mwi.example.org -s 127.0.0.1 -q SRV
command[check_sip_rr_srv]=/usr/local/nagios/libexec/check_dns -H _sip._tcp.rr.example.org -s 127.0.0.1 -q SRV
command[check_tcp_vm_srv]=/usr/local/nagios/libexec/check_dns -H _sip._tcp.vm.example.org -s 127.0.0.1 -q SRV
command[check_xmpp_server_srv]=/usr/local/nagios/libexec/check_dns -H _xmpp-server._tcp.example.org -s 127.0.0.1 -
q SRV
command[check_xmpp_client_srv]=/usr/local/nagios/libexec/check_dns -H _xmpp-client._tcp.example.org -s 127.0.0.1 -
q SRV
command[check_xmpp_conf_srv]=/usr/local/nagios/libexec/check_dns -H _xmpp-server._tcp.conference.example.org -s
127.0.0.1 -q SRV
```

As there are checks that are executed server side, those need to be defined in /usr/local/nagios/etc/objects/commands.cfg on the Nagios server.

```
define command{
command_name check_tftp
command_line $USER1$/check_tftp --get $HOSTADDRESS$ 000000000000.cfg 7167
}

define command{
command_name check_jabber
command_line $USER1$/check_jabber -H $HOSTADDRESS$ --expect='xmlns="jabber:client" from="example.org"'
}

define command {
command_name check_ssl_certificate
command_line $USER1$/check_ssl_certificate -H $HOSTADDRESS$ -c 3 -w 7
}
```

3rd Party Checks

check_jabber : https://exchange.nagios.org/directory/Plugins/Instant-Messaging/check_jabber_login/details

check_mongo : https://github.com/mzupan/nagios-plugin-mongodb

check_postgres : https://exchange.nagios.org/directory/Plugins/Databases/PostgresQL/check_postgres/details

For check_sip_registration I created a shell script that utilizes openuc-dbutil.

Additional Notes

You may find some checks complain of missing utils.pm. If you do, check if the script is making any references to the nagios plugins directory. You may need to alter the path to /usr/local/nagios/libexec/ .

Be sure to inspect any firewalls between your Nagios server and the sipXcom/openUC servers prior to running your checks. Some services such as ssh are restrictive by default in the sipXcom/openUC firewall.

It is possible to utilize sipsak to test against the SIP stack, however by default sipXcom / openUC will block the sipsak user agent and blacklist the source address it came from in iptables. Please be mindful of this!

Try not to cause unnecessary stress or bandwidth consumption on the server with your service checks. Once a day is probably good enough for a check interval for some services such as the SSL certificate check. See the "External Command Check Interval" section here : http://nagios.sourceforge.net/docs/3_0/configmain.html .