

# Nagios and Request Tracker Integration

## Creating Tickets

---

## Introduction

### Goals

- Learn how to configure Nagios to raise tickets in RT

### Notes

- Commands preceded with "\$" imply that you should execute the command as a general user - not as root.
- Commands preceded with "#" imply that you should be working as root.
- Commands with more specific command lines (e.g. "RTR-GW>" or "mysql>") imply that you are executing commands on remote equipment, or within another program.

## Exercises

To configure RT and Nagios so that alerts from Nagios automatically create tickets requires a few steps:

- Create a proper contact entry for Nagios in `/etc/nagios3/conf.d/contacts_nagios2.cfg`
- Update either `services_nagios2.cfg` or an individual host entry to use the new contact group.

These next two items should already be done in RT if you have finished the RT exercises.

- Install the `rt-mailgate` software and configure it properly in your `/etc/aliases` file for your MTA in use.
- Configure the appropriate queues in RT to receive emails passed to it from Nagios via the `rt-mailgate` software.

## Exercises

Log in to your virtual machine as the `sysadm` user.

## Configure a Contact in Nagios

Become root on your PC:

```
$ sudo -s
```

Edit the file /etc/nagios3/conf.d/contacts\_nagios2.cfg

```
# editor /etc/nagios3/conf.d/contacts_nagios2.cfg
```

In this file we will first add a new contact name under the default root contact entry.

WARNING: DO NOT remove the root contact entry.

REALLY - READ THIS! DO NOT remove the root contact entry.

The new contact should look like this (UNDERNEATH the root contact entry):

```
define contact{
    contact_name          net
    alias                 RT Alert Queue
    service_notification_period 24x7
    host_notification_period 24x7
    service_notification_options c
    host_notification_options d
    service_notification_commands notify-service-by-email
    host_notification_commands notify-host-by-email
    email                 net@pcX.ws.nsrc.org
}
```

Now at the end of the file add the following entry: (Note: do not delete or update other entries.)

```
define contactgroup{
    contactgroup_name      tickets
    alias                  email to ticket system for RT
    members                 net,root
}
```

Save and exit from the file.

## Notes

- the service\_notification\_option of "c" means only notify once a service is considered "critical" by Nagios (i.e. down). The host\_notification\_option of "d" means down. By specifying only "c" and "d" this means that notifications will not be sent for other states.
- Note the email address in use "net@pcX.ws.nsrc.org" - this is important as this was previously defined in the Request Tracker (RT) exercises.
- You could leave off "root" as a member, but we've left this on to have another user that receives email to help us troubleshoot if there are issues.

## Choose a Service to Monitor that Creates Tickets in RT

To send email to generate tickets in RT if SSH goes down on a box you would edit the SSH service check:

```
# editor /etc/nagios3/conf.d/services_nagios2.cfg
```

Find the service with the `hostgroup_name` of "ssh-servers" and add the "contact\_groups" entry at the end of the definition. When you are done your definition should look like this:

```
define service {
    hostgroup_name      ssh-servers
    service_description SSH
    check_command       check_ssh
    use                 generic-service
    notification_interval 0 ; set > 0 if you want to be
    renotified
    contact_groups      tickets,admins
}
```

Save and exit from the file.

## Notes

- Note the additional item that we now have, "contact\_groups." You can do this for other entries as well if you wish.
- We, also, included the default contact group of admins. You could leave this off if you wish.

## Include noc in as members of ssh-server

```
# editor /etc/nagios3/conf.d/hostgroups_nagios2.cfg
```

Find the group with the `hostgroup_name` of "ssh-servers" and add "noc" to the "members" entry at the end of the definition. When you are done your definition should look like this:

```
# A list of your ssh-accessible servers
define hostgroup {
    hostgroup_name  ssh-servers
    alias           SSH servers
    members         localhost,noc
}
```

Restart Nagios to verify your changes are correct:

```
# service nagios3 restart
```

If SSH goes down on any of the devices you are monitoring Nagios should generate a new ticket in Request Tracker. We will stop the SSH service on the classroom NOC server. If you are not monitoring this machine, then you will need to add an entry for NOC in your Nagios configuration and add it to the ssh hostgroup defined in the file `hostgroups_nagios2.cfg`.

## See Nagios Tickets in RT

- It will take a bit (up to 5 minutes) for Nagios to report that SSH is "critical". Then you must wait

for a total of 4 checks before the SSH service is deemed to be down "hard". At that point Nagios will send an email to [net@pcX.ws.nsrc.org](mailto:net@pcX.ws.nsrc.org) and a ticket will be created in RT.

- Remember to see this go to <http://pcX.ws.nsrc.org/rt/> and log in as Username "sysadm" with the password you chose when you created the RT sysadm account. The new ticket should appear in the "10 newest unowned tickets" box in the main login page in RT.