Lab 9.2: Nagios Notifications
IN719 Systems Administration

Introduction

Right now our Nagios servers are capable of sending us notifications via email when something happens. This is good, but if something happens while we're not checking email, then we won't find out until later. We would like to be able to push notifications to mobile devices so that we get alerts wherever we happen to be. Probably the best approach is to set up a server that is capable of directly sending SMS messages, but in situations like ours that is not feasible. Another approach is to use an SMS gateway service that accepts messages over the network via SMPP and forwards them via SMS. The downside of this approach is that if your network goes down you also lose your ability to send a notification that the network is down, but sometimes that's a compromise we have to make.

If we're sending notifications over the network anyway, we can also use modern team messaging solutions such as *Slack*, or, in our case, *Mattermost*. Both of them are largely compatible, but it shouldn't matter anyway, as you will learn how to integrate new commands (e.g. for notifications) into Nagios and how you can use them for notification purposes. In practice this will involve downloading and installing a script that is invoked by Nagios.

In the case of Mattermost (and Slack) we rely on webhooks to produce notifications, which involves the server-side configuration, before adding the command to Nagios and integrating it with your system monitoring.

1 Configure Mattermost notifications

1.1 Server-side configuration

In your Mattermost account you should see a channel that is specific to your team (along the lines of 'sysadmin-teamX' – where 'X' is your team number). Let me know if you can't see this channel. It means that I haven't assigned you to it yet.

Click on your username on the top left of the Mattermost menu. In the appearing menu click on 'Integrations'.

You should then see a windows that allows you to select between incoming and outgoing webhooks, as well as slash commands. Since we need to deal with incoming notifications, click on 'Incoming Webhook'.

You will see a list of available webhooks. Please don't modify any existing webhooks, since they may belong to other users. Create a new webhook by clicking on 'Add Incoming Webhook'.

In the following form, enter a display name following the pattern 'sysadmin-teamX' (where 'X' is your team id). This way we can see which webhook belongs to which installation. Enter a description of your choice (e.g. 'Notification webhook for SysAdmin Team X'). Finally, select the destination channel for any notification, which should be your team channel. Once done, save the new webhook.

Mattermost then generates a webhook URL, which you need to copy for later use. (If you forgot to copy it, you can look it up under the 'Integrations' menu at any time.)

1.2 Client-side configuration

On your `mgmt` machine, download the nagios-mattermost plugin from

`https://github.com/NDrive/nagios-mattermost/raw/master/mattermost.py`

using `wget`. Skim over the script to see what it does, and to ensure that it has been downloaded properly.

(You can find more information about the extension we are using under `https://github.com/NDrive/nagios-mattermost`.)

- Once done, copy the downloaded file to `/usr/sbin`
- Make it executable for everyone (including owning user and group).
- Test it by running it from the command line. Run `mattermost.py --help` to see the syntax. The necessary parameters are
  - `--url` (This is your webhook URL)
  - `--notificationtype` (Hint: Check the Nagios documentation for notification types.)
  - `--hostalias` (You can make up a value of your choice.)
  - `--hostaddress` (You can make up a value of your choice.)

  Ensure that you get a notification in Mattermost before you continue.

2 Integrating Nagios with Mattermost Notification Commands

Now it is time to make the script accessible from Nagios.

For this purpose, add the raw commands shown below to the file `/etc/nagios3/commands.cfg`. This effectively allows you to use the commands `notify-service-by-mattermost` and `notify-host-by-mattermost` from within Nagios.

When adding the commands, replace all occurrences of `[MATTERMOST-WEBHOOK-URL]` with your webhook URL you saved earlier.

Replace `[OPTIONAL-MATTERMOST-CHANNEL]` with your team's Mattermost channel name.

```
define command {
    command_name notify-service-by-mattermost
    command_line /usr/sbin/mattermost.py --url [MATTERMOST-WEBHOOK-URL] \
                                                    --channel [OPTIONAL-MATTERMOST-CHANNEL] \
                                                    --notificationtype "$NOTIFICATIONTYPE$" \
                                                    --hostalias "$HOSTNAME$" \
                                                    --hostaddress "$HOSTADDRESS$" \
                                                    --servicedesc "$SERVICEDESC$" \
                                                    --servicestate "$SERVICESTATE$" \
                                                    --serviceoutput "$SERVICEOUTPUT$"
}

define command {
    command_name notify-host-by-mattermost
    command_line /usr/sbin/mattermost.py --url [MATTERMOST-WEBHOOK-URL] \
                                                    --channel [OPTIONAL-MATTERMOST-CHANNEL] \
                                                    --notificationtype "$NOTIFICATIONTYPE$" \
                                                    --hostalias "$HOSTNAME$" \
                                                    --hostaddress "$HOSTADDRESS$" \
                                                    --hoststate "$HOSTSTATE$" \
                                                    --hostoutput "$HOSTOUTPUT$"
}
```

Restart nagios to check whether the configuration is picked up without error. Hint: For debugging, use the syslog on the nagios machine.

3 Integrating Mattermost Notifications into your Puppet-based Nagios configuration

Now you can use the new commands in your Nagios configuration. Simply add a new contact for mattermost that triggers the new commands.

```
nagios_contact { 'mattermost':
              target => '/etc/nagios3/conf.d/ppt_contacts.cfg',
              alias => 'Mattermost Webhook',
              service_notification_period => '24x7',
              host_notification_period => '24x7',
              service_notification_options => 'w,u,c,r',
              host_notification_options => 'd,r',
              service_notification_commands => 'notify-service-by-mattermost',
              host_notification_commands => 'notify-host-by-mattermost',
              email => 'root@localhost',
}
```

As a last step, add the new contact as a member to your *contact group* (e.g. `'user1, user2, mattermost'`), so it gets triggered whenever a service or host outage is detected.

Apply the new configuration, and ensure that Nagios restarts properly. Remember: you may need to manually update the permissions on the Nagios configuration files – the involved commands should be in your documentation.

Produce a service outage and see whether the notifications work.

Hint: Use `tail -f` to follow the syslog output, so you can see what nagios and other services are doing.

Once everything works, reflect on your system and update your puppet configuration.