Microsoft System Center Operations Manager über SMTP an Nagios anbinden

Diese Dokumentation basiert auf den HowTo's „SMTP-Handling für Nagios HowTo" und „handle_SCOM_mail-README" von Ingo Lantschner.(e-Mail: ingo@boxbe.com) http://ingo.lantschner.name/Nagios.html

Zusammengefasst von Heiko Schmuck. Version 1.1

1 Konfiguration am OpsMgr

1.1 Konfiguration des OpsMgr für den Versand von Email-Notifications

Um den OpsMgr dazu zu bringen grundsätzlich einmal Emails zu versenden haben sich die folgenden Schritte bewährt:

1. Konfiguration des Netzwerkes am OpsMgr-Windowsserver statisch. Wenn der OpsMgr zugleich DC ist (was vor allem in Testumgebungen sein kann) dann muss localhost als DNS eingetragen werden.
2. Anlegen eines Email Notification Action Account (enaa). Dieser braucht das Recht "Log on Interactivly"

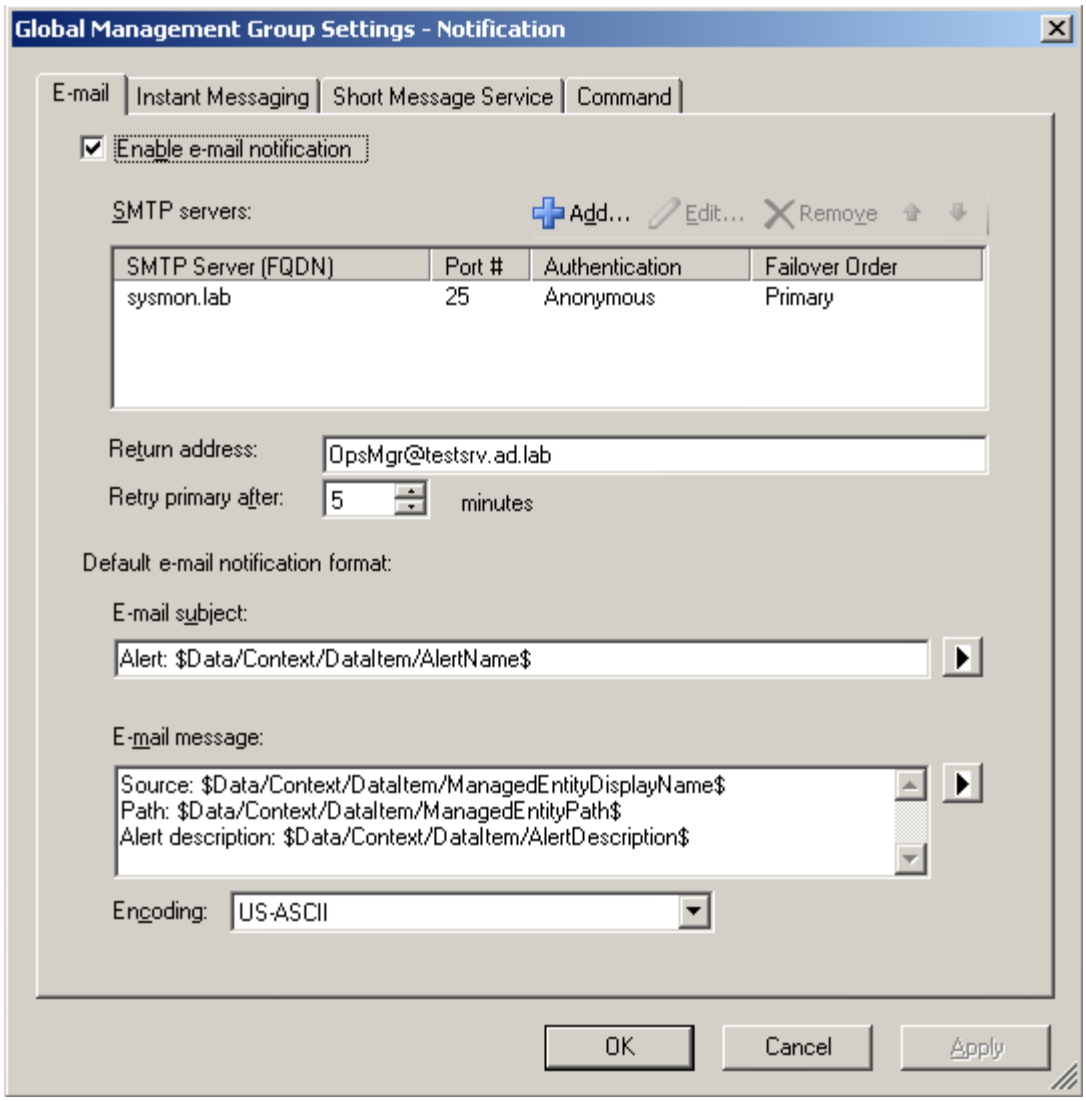
1.2 Konfiguration Alerts am OpsMgr

An sich wie in der Hilfe das OpsMgr beschrieben. Ein paar Abweichungen auf Grund der besonderen Verwendung sind:

Die zu übermittelnde Info auf das Nötigste reduzieren: AlertName, Entity-Displayname und -Path genügen.

Das Encoding wie bereits erwähnt auf US-ASCII umstellen - mit UTF-8 wird es nur komplizierter.

Problematisch kann sein, dass im OpsMgr nur ein E-Mail Pfad angelegt werden kann. Sollen auch weiterhin E-Mails an Admins versendet werden, muss im e-Mail System ein Gateway zum Nagios-Server angelegt werden.


Global Management Group Settings - Notification
E-mail
Instant Messaging
Short Message Service
Command
Enable e-mail notification
SMTP servers:
Add...
Edit...
Remove
SMTP Server (FQDN)
Port #
Authentication
Failover Order
sysmon.lab
25
Anonymous
Primary
Return address:
OpsMgr@testsrv.ad.lab
Retry primary after:
5
minutes
Default e-mail notification format:
E-mail subject:
Alert: $Data/Context/DataItem/AlertName$
E-mail message:
Source: $Data/Context/DataItem/ManagedEntityDisplayName$
Path: $Data/Context/DataItem/ManagedEntityPath$
Alert description: $Data/Context/DataItem/AlertDescription$
Encoding:
US-ASCII
OK
Cancel
Apply

2 Die Nagios-Seite

Auf Seite des Nagios-Servers haben wir zunächst einmal einen Mailserver, der auf Port 25 lauscht. Die vom OpsMgr empfangenen Emails werden über eine Pipe von einem Mailalias in ein Programm (passive check) übergeben

2.1 Architektur des Checks

Die am Port 25 des Nagios-/SMTP-Servers eintreffende Email wird über einen Mailalias an ein kleines Skript übergeben, welches die Infos ausfiltert und in das Nagios Commandfiles schreibt. Für eine erfolgreiche Verarbeitung eines passiven Check-Ergebnisses benötigt Nagios:

1. Servicename: Name des Service
2. Status: CRITICAL, WARNING, OK
3. Hostname: Name des Hosts, hier immer der Nagios-Name des OpsMgrs (siehe unten)
4. Output: Ausgabe des Checks (erläuternder Text zum jeweiligen Zustand)

Diese werden nun aus dem vom OpsMgr versendeten Email-Allert extrahiert wie folgt:

SERVICE	Empfängername; Userteil der Empfängeradresse	to: **opsmgr@**sysmon.lab, also opsmgr
STATE	dzt. immer CRITICAL	könnte zukünftig aus dem „Resolution State:“ bzw. der „Severity“ geschlossen werden.
HOST	OpsMgr-Server; also der Serverteil der Absenderadresse	from: OpsMgr**@scom.ad.lab**, also scom.ad.lab
OUTPUT	Text nach „Alert: “ aus Subject bzw. Mailbody	Alert: **Rootserver not running**

2.2 Das Skript

Das im folgenden aufgelistete Script ist der eigentliche Eventhandler, der die Information aus dem vom OpsMgr versendeten Email extrahiert und als Passivecheck in die Nagios Commandpipe schreibt. Die Pfade müssen möglicherweise an das entsprechende System angepasst werden.

```
#!/usr/bin/perl -w

use strict;
use warnings;

my $commandfile = "/usr/local/nagios/var/rw/nagios.cmd";
my $debug = 1;

my @input=<STDIN>;
my ($service,$state,$output,$host);

foreach (@input) {
        if ($debug==1) {
                open DEBUGLOG, ">> /tmp/smtpdebug.log" or die "Can not open
/tmp/smtpdebug.log\n";
                        print DEBUGLOG "$_\n";
                        close DEBUGLOG;
        }

        if (/^[tT]o:\ (.+)@/) {
         $service = $1;  # Service ist set to the userpart of the destination-address
    }

    if (/^[fF]rom:\ .+@(.+)[>]*/) {
         $host = $1;  # Host is set to the host-part of the from-address
    }

    if (/^[sS]ubject:\ .*Alert:\ (.*)/) {
         $output = $1;  # output is set to all text after "Alert: " in the subject-line
    }

    $state = 2;  # each email received sets the service in Nagios to CRITICAL
                                # (this could be tweaked later)

}
my $return=system("/usr/local/nagios/libexec/eventhandlers/submit_check_result", $host,
$service, $state, "$output");

if ($debug==1) {
        open DEBUGOUT, ">> /tmp/smtpdebug" or die "Can't open /tmp/smtpdebug\n";
       print DEBUGOUT "PROCESS_SERVICE_CHECK_RESULT;$host;$service;$state;\"$output\"\n";
       close DEBUGOUT;
}
exit $return;
```

2.3 Konfiguration Mailserver

Um die OpsMgr-Alerts per Mailpipe als Passive-Check im Nagios verwertet, muss lediglich ein zusätzliches Aliasfile angelegt und gehashed werden.

Folgend das Alias-File:

```
## /etc/postfix/nagios
        opsmgr: "|/usr/bin/perl -w /usr/local/nagios/libexec/eventhandlers/handle_SCOM_mail.pl"
```

Konfiguration Postfix

Wenn Postfix schon so konfiguriert ist, dass er Email versendet, dann müssen in der `main.cf` noch eine Zeile hinzugefügt und drei ergänzt werden:

```
1 ## /etc/postfix/main.cf
2
3 smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
4 biff = no
5
6 append_dot_mydomain = yes
7
8 mydomain = meinefirma.net
9 alias_maps = hash:/etc/aliases, hash:/etc/postfix/nagios
10 alias_database = hash:/etc/aliases, hash:/etc/postfix/nagios
11 myorigin = $mydomain
12 mydestination = $myhostname, localhost.localdomain, localhost,
nagiosalert.meinefirma.net
13 relayhost = [10.10.4.xx]
14 mynetworks = 10.10.4.0/24, 127.0.0.0/8
15 relay_domains =
16 mailbox_size_limit = 40000000
17 inet_interfaces = 10.10.4.11
18 message_size_limit = 10240
```

Ohne die Zeile 17 und die Angabe des LANs in mynetworks (Zeile 14) hört Postfix nur auf localhost. Diese Beispieladresse ist die des Nagiosservers und natürlich an die eigene Umgebung anzupassen.
Weiters sollen die eingehende, an Nagios adressierten Email nicht in einem Postkörbchen abgelegt werden, sondern direkt einem Eventhandler übergeben werden, der dann die weitere Bearbeitung und Übergabe an Nagios steuert. Dies ist möglich, indem in der Datei `main.cf` (im Beispiel oben Zeile 9) eine zusätzlicher Alias-Datei namens `/etc/postfix/nagios` definiert wird. In dieser steht dann der Aufruf eines Scripts „handle_SCOM_mail.pl“.

```
opsmgr: "|/usr/bin/perl -w
/usr/local/nagios/libexec/eventhandlers/handle_SCOM_mail.pl"
```

Warum nicht in /etc/aliases?

Externe Kommmandos in dieser Datei, werden mit den Rechten *nobody:nogroup* ausgeführt. Diese haben natürlich nie und nimmer die Erlaubnis in das Nagios Commandfile zu schreiben. Der *local*-Daemon führt die Zustellung an externe Komandos, in unserem Fall das handle_SCOM_mail.pl Script, mit den Rechten des Owners der Alias-Datenbank aus, wenn dieser nicht root ist. Daher definieren wir ein **eigenes** Alias-Table-File nur für nagios und berechtigen diese entsprechend:

```
$ ls -l /etc/postfix/nagios
-rw-rw-r-- 1 opsmgr root nagios
```

Mit postalias wird diese Textdatei dann zum ersten mal gehasht.

```
/etc/postfix$ sudo postalias -o nagios
```

Wegen der ungewöhnlichen Berechtigung muss der Parameter `-o` mitangegeben werden, und nacher das db-File mit `chown` dem User *nagios* zugewiesen werden. Letztendlich muss es so aussehen:

```
/etc/postfix$ ll nagios*
-rw-rw-r-- 1 nagios root nagios
-rw-r--r-- 1 nagios root nagios.db
```

Übrigends: Die Zeile 10 sorgt dafür, dass mit Aufruf des Komandos newaliases auch diese nagios-Tabelle aktualisiert wird.

2.4 Nagios-Konfiguration

Architektur

Im Nagios werden die Alerts Services zugewiesen, die wiederum zu einem Host gehören. Der einfachste Ansatz und somit ein guter Ausgangspunkt ist, alle Alerts die vom OpsMgr kommen einem Pseudo-Host (z.B. dem OpsMgr selber) zuzuordnen. Theoretisch könnte man alle Alerts in einem Service zusammenfassen. Sinnvoller wird es aber sein, eine gewisse Gliederung vorzunehmen. Diese erfolgt in dem hier vorgeschlagenen Modell nach Recipients, die einer bestimmten Subscription zugeordnet werden, die wiederum definiert welche Scopes alarmiert werden. Je Recipent wird dann eine eigene Emailadresse angegeben, deren Userteil identisch mit dem Servicename im Nagios sein müsste.

Schöner und vor allem leichter zu warten wäre, wenn der Servicename anderwertig ermittelt werden würde. Dazu muss man noch sehen, was der OpsMgr so alles ausgibt.

Nagios Object Definitionen

Anzulegen ist ein Host mit dem gleichen Namen wie der Hostteil des versenden Users im OpsMgr. Als Service ist der Userteil der Empfängeradresse anzulegen.

```
## Configuration for SCOM 2007

define host {
      use                     generic-host
      host_name               opsmgr.ad.lab
      address                 10.10.4.21
      alias                   opsmgr.ad.lab
      max_check_attempts      3
      check_period            24x7
      contacts                nagiosadmin
      notification_interval   60
      notification_period     24x7

      }
```

```
define service {
      use                     generic-service
      check_freshness         0
      check_period            none
      passive_checks_enabled  1
      active_checks_enabled   0
      check_command           check_dummy
      host_name               opsmgr.ad.lab
      service_description     opsmgr
      }
```

2.5 Hurra

Service Status Details For All Hosts

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	08-07-2007 09:31:37	30d 4h 8m 24s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	08-07-2007 09:32:51	30d 4h 7m 46s	1/4	USERS OK - 4 users currently logged in
	HTTP	OK	08-07-2007 09:34:05	30d 4h 7m 9s	1/4	HTTP OK HTTP/1.1 200 OK - 740 bytes in 0.030 seconds
	PING	OK	08-07-2007 09:30:19	30d 4h 11m 31s	1/4	PING OK - Packet loss = 0%, RTA = 0.08 ms
	Portscan	OK	07-13-2007 09:51:29	24d 23h 43m 19s	1/4	Status OK
	Root Partition	OK	08-07-2007 09:31:33	30d 4h 10m 54s	1/4	DISK OK - free space: / 4820 MB (88% inode=95%):
	SSH	OK	08-07-2007 09:33:31	30d 0h 42m 16s	1/4	SSH OK - OpenSSH_4.2p1 Debian-7ubuntu3.1 (protocol 2.0)
	Snort_Alert	OK	08-07-2007 09:33:28	23d 21h 2m 26s	1/1	p□F no errors or warnings
	Swap Usage	OK	08-07-2007 09:34:42	30d 4h 9m 39s	1/4	SWAP OK - 100% free (290 MB out of 290 MB)
	Total Processes	OK	08-07-2007 09:30:56	30d 4h 9m 1s	1/4	PROCS OK: 32 processes with STATE = RSZDT
opsmgr.ad.lab	opsmgr	CRITICAL	08-07-2007 09:32:38	0d 17h 29m 48s	3/3	smux Group Roll-up Monitor?=

11 Matching Service Entries Displayed