SCOM 2007 (aka OpsMgr) nach Nagios berichten lassen

Ingo Lantschner (ingo@boxbe.com)

Version 1.01 erstellt am 10. 9. 2007

Dieses Dokument ist eine erste Zusammenstellung bis dahin loser Notizen. Korrekturhinweise, Ergänzungsvorschläge Lob&Kritik bitte direkt per Email an den Autor. Danke!

Die an sich naheliegende Methode Emails an den Nagios-Server zu senden und dort vom Postfix und Nagios auswerten zu lassen ist trickreicher als man zunächst denkt - denn wie bewege ich den OpsMgr dazu, die Emails an einen SMTP-Server zu übergeben? Inwieweit spielt da der Action-Account eine Rolle? Äusserst komplex für eine eigentlich einfache Aufgabenstellung. Und dann ist da noch die Konfiguration des Mailservers und des Nagios. All diese Fragen soll das folgende Dokument beantworten.

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 eingertaregn werden.
2. **Nur relevant , wenn als virtueller Server auf VMware aufgesetzt wird:** Nach einem Reboot war die Konfig wieder weg - das lag daran, dass der Windows Server 2003 einen versteckten Adapter konfiguriert hatte - den musste ich erst mal sichtbar machen und dann entfernen
   - DEVMGR_SHOW_NONPRESENT_DEVICES=1
   - DEVMGMT.MSC
   - Show Hidden Devices anhackeln
   - ausgegrauten Netzwerkadapter gelöscht

   Seitdem merkt sich der Server seine Konfiguration.
3. Anlegen eines Email Notification Action Account (enaa). Dieser braucht das Recht "Log on Interactivly" - nun gut, nicht ganz unlogisch. Ich hab ihn in die Domainadmins gegeben und siehe da, die Emails rauschen zum Mailserver.

Teststellung: Wie bekomme ich rasch Alarme zum Testen der Konfiguration

Wann der OpsMgr Alarm schlägt ist einigermaßen undurchsichtig. Der Verfügbarkeits-check für SNMP-Devices (der auf Basis von SNMP-get auf Port 161 basiert) alarmiert nur sehr sporadisch. Als einfach und schnell einzurichtender Monitor hat sich der TCP-Portcheck erwiesen. Auf einem Linux-Host läuft mit dem snmpd meist auch der smux auf tcp/199 mit. Dieser kann vom OpsMgr aus gecheckt und mit einem Intervall von 1 Minute versehen werden. So hat man eine sehr rasche Reaktion auf Veränderungen, was sich vor

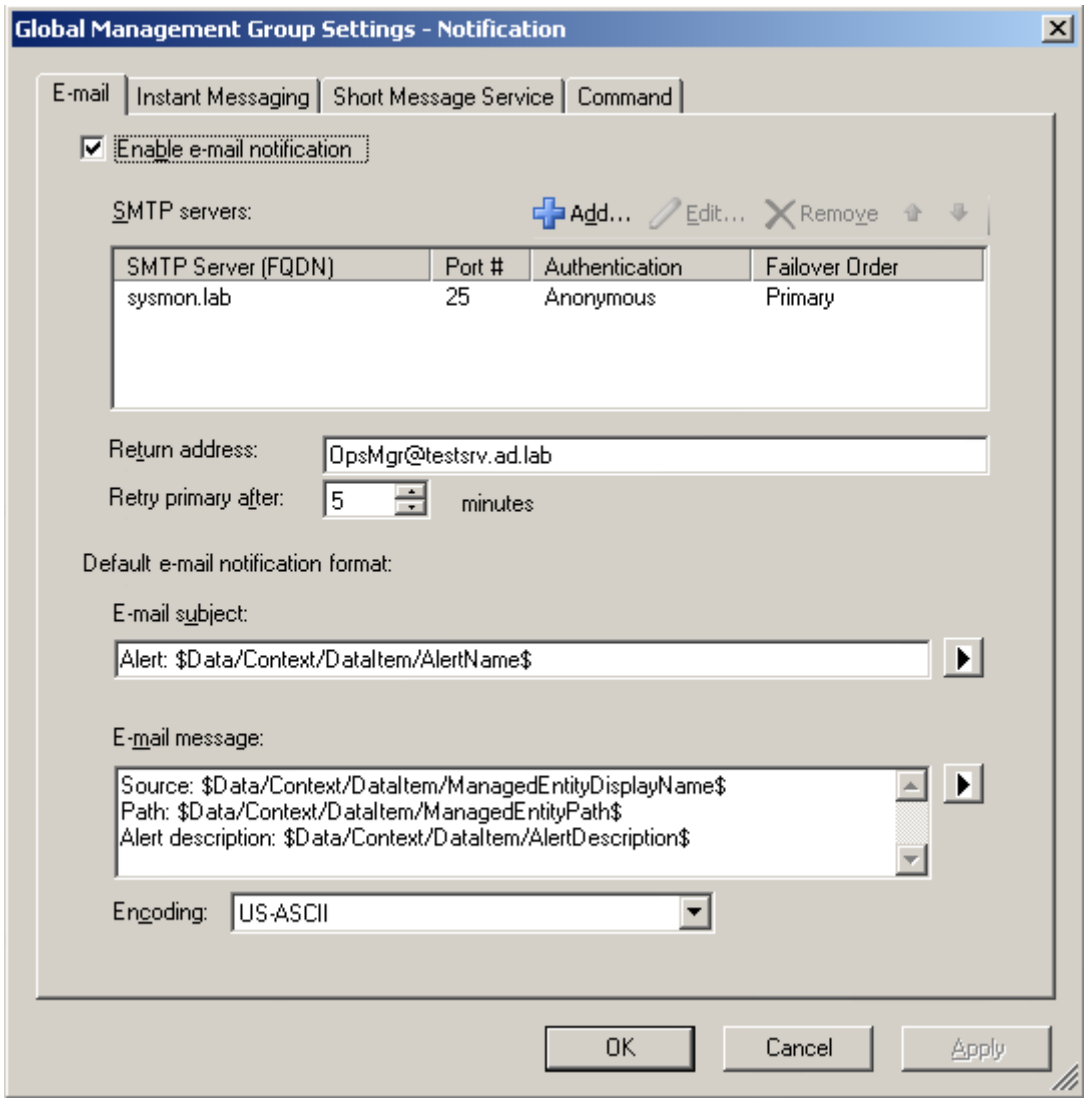
allem in der Testphase als Segen erweist.

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.



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 können nun auf dreierlei Art verarbeitet werden:

- Parsing des mbox-Files mit check_logfiles (active check)
- Auslesen der Mailbox mittels POP3-Servers (active check)
- Pipe von einem Mailalias in ein Programm (passive check)

Die Variante mit check_logfiles hat nicht auf Anhieb funktioniert - scheint wie wenn check_logfiles mit der mbox-Datei nicht so ganz klar kommt bzw. dazu eine intensivere Beschäftigung mit dem Plugin nötig wäre. Für die Variante des aktiven Checks über ein als POP3-Client arbeitendes Perl-Skript habe ich ein kleines Plugin entworfen. Dieses hat bei der Implementierung aber ein gewisses Problem: Was ist, wenn mehrere Emails zwischen zwei Aufrufen eingetroffen sind? Auch muss es wohl mit einem sehr engen Checkintervall eingerichtet werden, um die Latenzzeiten erträglich zu halten. Das verursacht dann wieder jede Menge Traffic am Netz. Am besten gefällt mir die oben als letztes beschriebene Variante mit dem passiven Check. Diese wird daher im Folgenden genauer beschrieben.

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:

- Servicename: Name des Service
- Status: CRITICAL, WARNING, OK
- Hostname: Name des Hosts, hier immer der Nagios-Name des OpsMgrs (siehe unten)
- 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.

```
#!/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. (Das ist nicht ganz einfach, siehe SMTP-Nagios-Doku[1] .

Folgend das Alias-File:

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

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

1http://www.nagiosexchange.org/Misc.54.0.html?&tx_netnagext_pi1[p_view]=1081

definiert welche Scopes alarmiert werden. Je Recipent wird dann eine eigene Emailadresse angegeben, deren Userteil ident 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 Obeject 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                 192.168.96.11
      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

Service Status Details For Host 'opsmgr.ad.lab'

Host	Service	Status	Last Check	Duration	Attempt	Status Information
opsmgr.ad.lab	opsmgr PASV	CRITICAL	08-07-2007 09:32:38	0d 17h 30m 29s	3/3	smux Group Roll-up Monitor?=

1 Matching Service Entries Displayed

ToDos

- Zuordnung der Alerts zu verschiedenen Services ohne deswegen neue Recipients/Aliases anlegen zu müssen.
- Bereinigung des outputs, „?=“ abschneiden